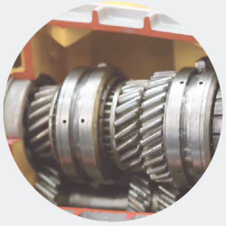
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA E GESTÃO INDUSTRIAL






Simultaneous Pallet Loading Problem: Picker-to-Parts System

ANDREIA SOARES CORREIA
outubro de 2024

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Simultaneous Pallet Loading Problem: Picker-to-Parts System

Instituto Superior de Engenharia do Porto
Master in Management and Industrial Engineering

Andreia Soares Correia
ID number 1170880

Advisor
Prof. António José Galrão Ramos

Academic Year 2023/2024

Thesis defended on 9 October 2024
in front of a Board of Examiners composed by:

Prof. Manuel Joaquim Pereira Lopes (chairman)
Prof. Elsa Marília Costa Silva
Prof. António José Galrão Ramos

Simultaneous Pallet Loading Problem: Picker-to-Parts System
Instituto Superior de Engenharia do Porto

This thesis has been typeset by LaTeX and the isepthesis class.

Author's email: andreia_correia2000@live.com.pt

Acknowledgments

As I conclude this stage of my academic education, it is with deep gratitude that I acknowledge the various people who have played a crucial role throughout my journey.

Firstly, I would like to thank my advisor, Professor António Galrão Ramos, for his guidance throughout this project. His advice was fundamental to the development and success of this thesis.

I also want to thank the professors from both my Bachelor's and Master's degrees who have made my passion for Management and Industrial Engineering grow over the years and contributed to my academic and personal development.

I thank my close friends who have shared this academic journey with me for all their support and for all the unforgettable memories we have created.

I would like to give a very special thank you to my family, especially my parents, Margareth and João, for being my main foundation throughout the years and for always believing in my potential. It is impossible to thank them enough for everything they have done and continue to do for me. Every achievement I have made undoubtedly reflects the support and values you have taught me.

I also want to express my deep gratitude to Pedro for his unconditional encouragement, for always being there for me, and for contributing to my emotional well-being.

Finally, I wish to express my deepest gratitude to all those I have mentioned and to all those who have contributed in any way to the realisation of this project. This journey would have been much more difficult without your support.

Abstract

This study focuses on a novel robot order picking optimisation problem. Therefore, a Decision Support System was developed to optimise the picking and delivery process in a picker-to-parts system assisted by Autonomous Mobile Robots (AMRs). This problem takes place in the context of logistics for a fictitious retail company. The main goal is to automate the order picking process to reduce the total operation time and optimise the efficiency of Autonomous Mobile Robot routes.

This research has been motivated by the need to improve the efficiency of operations in automated warehouses. Order picking for several customers can be complex and sometimes time-consuming. Therefore, it became crucial to study the motion of the AMRs to follow a predefined delivery sequence to optimally allocate the pallets and deliver them in the right order to meet customer demands.

The problem has been modelled using a Mixed Integer Programming approach, and the software *IBM ILOG CPLEX Optimization Studio* was used to implement it. The model uses a lexicographic objective function that optimises, in order of priority, the makespan, the number of pallets used, and the unpairing of Autonomous Mobile Robots and pallets in consecutive time slots. Additionally, a total of 81 different instances were tested, varying the sizes of the sets and the values of the parameters, to evaluate the performance of the model under different conditions.

The results indicate that the proposed model effectively reduces operation time by optimising the movement and allocation of AMRs, especially in scenarios with moderate complexity. In cases where the number of customers and pallets increased significantly, the model still found optimal solutions within reasonable computation times, although in some instances the time limit imposed was reached before full optimisation could occur.

It is concluded that the proposed model efficiently optimises robot routes and pallet allocation, showing significant improvements in order picking efficiency in automated warehouses. However, the model faced some computational challenges as the problem complexity increased, which suggests that future work could explore heuristic or metaheuristic approaches to complement the optimisation in larger-scale instances.

Keywords

Order Picking, Picker-to-Parts System, Autonomous Mobile Robot, Robot Picking System, Dynamic Picking, Warehouse Automation, Logistics

Resumo

Este estudo centra-se num novo problema de otimização da recolha e entrega de itens por robôs. Assim, foi desenvolvido um sistema de apoio à decisão para otimizar o processo de recolha e entrega num sistema de *picking* assistido por *Autonomous Mobile Robots* (AMRs). Este problema ocorre no contexto da logística de uma empresa de retalho fictícia. O principal objetivo é automatizar o processo de recolha para reduzir o tempo total da operação e otimizar a eficiência das rotas dos *Autonomous Mobile Robots*.

Esta investigação foi motivada pela necessidade de melhorar a eficiência das operações em armazéns automatizados. O processo de entrega de encomendas para vários clientes pode ser complexo e por vezes demorado. Por conseguinte, tornou-se crucial estudar o movimento dos AMRs, de modo a otimizar o processo de entrega de paletes e satisfazer as procuras dos vários clientes.

O problema foi modelado através de uma abordagem de *Mixed Integer Programming*, e o *software IBM ILOG CPLEX Optimization Studio* foi utilizado para o implementar. O modelo matemático compreende uma função objetivo lexicográfica que otimiza, por ordem de prioridade, o *makespan*, o número de paletes utilizadas e o desemparelhamento dos *Autonomous Mobile Robots* e paletes em intervalos de tempo consecutivos. Adicionalmente, foram testadas 81 instâncias diferentes, variando o tamanho dos conjuntos e os valores dos parâmetros, para avaliar o desempenho do modelo em diferentes condições.

Os resultados indicam que o modelo proposto reduz eficazmente o tempo de operação através da otimização da movimentação e alocação de AMRs, especialmente em cenários de complexidade moderada. Nos casos em que o número de clientes e de paletes aumentou significativamente, o modelo encontrou soluções ótimas dentro de tempos de execução razoáveis, embora em alguns casos o limite de tempo imposto foi atingido antes que a otimização completa pudesse ocorrer.

Conclui-se que o modelo proposto otimiza eficientemente as rotas dos robôs e a atribuição de paletes, mostrando melhorias significativas na eficiência da recolha de encomendas em armazéns automatizados. No entanto, o modelo enfrentou alguns desafios computacionais à medida que a complexidade aumentou, o que sugere que o trabalho futuro poderia explorar abordagens heurísticas ou metaheurísticas para complementar a otimização em instâncias de maior escala.

Contents

List of Abbreviations

2D-BPP	Two-dimensional Bin Packing Problems
3D-BPP	Three-dimensional Bin Packing Problem
3D-PP	Three-dimensional Packing Problem
AGV	Automatic Guided Vehicles
ALNS	Adaptive Large Neighborhood Search
AMR	Autonomous Mobile Robot
AOPP	AMR-assisted Order Picking Problem
AS/RS	Automated Storage and Retrieval Systems
BPP	Bin Packing Problem
COP	Combinatorial Optimisation Problem
LKH	Lin-Kernighan Heuristic
MCPP	Mixed-case Palletising Problem
MIP	Mixed-Integer Programming
NEH	Nawaz, Enscore, and Ham Heuristic
OP	Order Picking
OPL	Optimization Programming Language
PTRs	Pick and Transport Robots
QOP	Quasi-online Packing Problems
S/R	Storage and Retrieval

SKU	Stock Keeping Units
TS	Tabu Search
TSP	Traveling Salesman Problem
VBA	Visual Basic for Applications
VLM	Vertical Lift Modules
VRP	Vehicle Routing Problem

List of Tables

List of Figures

Chapter 1

Introduction

This chapter describes the scope of the project and its relevance to the picking paradigm. It also provides a brief overview of the various aspects of the project. It then explains the overall aim of the project, its research question, and the objectives that it aims to achieve are then explained.

1.1 Research problem, framework, and relevance

The proposed problem focuses on the internal logistics of a fictitious retail company seeking to automate its order picking process. In this system, there is an intermediate picking area with a designated space for each customer, where empty pallets are fulfilled based on received orders. Unlike traditional picker-to-parts systems, where human pickers select items from shelves using a pick list, this problem introduces automation with Autonomous Mobile Robot (AMR) that handles all picking and delivery tasks. In this process, AMRs retrieve pallets with specific items from the warehouse and deliver them to a dynamic picking area, where items are distributed to customer-specific pallets. This process is repeated until all orders are complete. Once an order is ready, operators move the corresponding pallet to the marshalling area and then to the truck loading area. Given that the orders vary across multiple customers, the system must account for the pre-defined delivery sequence of each customer and the efficient assembly of pallets.

It is important to highlight that this order picking system contrasts with traditional parts-to-picker systems. In this case, the AMRs operate in a continuous cycle, moving between various locations in the warehouse and the customer-specific lanes, directly collecting and delivering items to the respective customer pallets. This demonstrates that the picker (AMR) actively moves through the warehouse to gather items, rather than the parts being brought to a stationary picker. In a true parts-to-picker system, the process would involve AMRs or automated systems transporting items to a fixed station, where sorting and order fulfilment occur in one location, without requiring repeated returns to the warehouse for additional items. This distinction confirms that the system in question is a picker-to-parts model, where the picker seeks the parts rather than vice versa.

The objective of this research is to integrate AMRs into the order picking process, seeking optimal routes for efficient deliveries, by minimising the total time required to complete the delivery process.

Thus, to thoroughly understand the problem at hand, intense research was initially carried out, focusing on the various order picking systems. In addition to optimising AMR routes, it is also essential to consider how collected items are arranged on pallets to maximize space utilisation and load stability. To address this, Three-dimensional Packing Problem (3D-PP) were examined, particularly focusing on Mixed-Case Palletising Problems, which deal with organizing different items on pallets. While the 3D-PP problem was not the primary focus of this research, its concepts significantly influenced the optimised arrangement of pallets in the picking process. These principles complemented the AMR route optimisation, ensuring efficient transport and allocation. This integrated approach provides a holistic solution, where transport efficiency and pallet allocation are considered together to enhance overall operational performance in an automated warehouse.

After analysing all relevant studies made on the field of order picking, it has been determined that this problem falls under the category of a picker-to-parts order picking system, in which the loading sequence is already known. Thus, it was developed a mathematical model that determines optimal AMR routes while respecting the pallet delivery sequence, thus ensuring an efficient order preparation process. Moreover, these considerations align with the research question and objectives, which will be further discussed in the following section.

1.2 Research question and objectives

Starting from the aforementioned problem, this research seeks to answer the following question: "How can the development of a Decision Support System optimise the efficiency of AMR-assisted order picking in a picker-to-parts system, considering pre-defined customer sequences for mixed-pallet construction?".

Based on the presented research question, this can be answered by the main objective of this project, which is to develop a mathematical model for the optimisation of a picking and delivery process in an AMR-assisted picker-to-parts system, specifically addressing the pre-defined customer delivery sequences. Furthermore, the following specific objectives have been developed to enhance the general objective of this research:

- Review the technical-scientific literature concerning order picking systems and the role of Autonomous Mobile Robots (AMRs) in order picking;
- Characterize and describe the problem under study;
- Design and develop the mathematical model based on the described problem, ensuring it accommodates the constraints and goals of the picking system;
- Optimisation of the parameters of the mathematical model to enhance its performance and applicability;
- Test and validate the formulated mathematical model;
- Evaluate the results and assess the efficiency of the model in improving the picking process.

1.3 Methodological Considerations

According to Williams (2007), there are three common approaches to conducting research: quantitative, qualitative, and mixed research. Quantitative research begins with a problem statement and involves developing a hypothesis, performing a literature review, and carrying out a quantitative analysis. It also involves the use of various strategies, such as surveys and experimental studies. On the other hand, Traqueia et al. (2021) say that qualitative research offers a variety of tools that allow researchers to explore the uniqueness of their subjects, such as tools with open-ended questions, interviews, observations, documents, audio-visuals, and text and picture analysis. A quali-quantitative research technique known as "mixed research" involves the use of various forms of data collection that can be analysed statistically and textually. This project falls under the quantitative approach since it has objective solutions that might be established via scientific methods, and all objects of study are observable, quantifiable, and measurable (Coutinho, 2014).

This project aims to optimise the order picking process in a fictitious retail setting, focusing on the efficient use of Autonomous Mobile Robots in a picker-to-parts system. The approach to developing a Decision Support System model involves the formulation of a Mixed-Integer Programming (MIP) model with a discrete time approach that optimises AMR routes, considering pre-defined customer sequences for the construction of mixed pallets. The mathematical model is implemented in *IBM ILOG CPLEX Optimization Studio* software, which allows for iterative modifications and refinements until the model closely mirrors the operational realities of the problem. Once the final model is established, it is validated, evaluated, and analysed.

1.4 Structure

This thesis is divided into five chapters: "Introduction", "Literature Review", "Problem Description and Mathematical Model", "Results and Discussion", and "Conclusions".

In Chapter 2 ("Literature Review"), the relevant literature is explored in order to frame this research. The chapter adopts a funnel strategy, starting with an overview of Combinatorial Optimisation Problems, followed by Three-dimensional Packing Problems, and, finally, Order Picking Processes. Special emphasis is placed on parts-to-picker and picker-to-parts systems, as well as examining the use of automated solutions in warehouse environments.

In Chapter 3 ("Problem Description and Mathematical Model"), the mathematical model that optimises the robot-assisted picking process is presented. The chapter begins with a general description of the order picking process in a retail company, detailing the different stages involved. This is followed by the development of the mathematical model, which includes sets, parameters, decision variables, a lexicographic objective function, and various constraints.

In Chapter 4 ("Results and Discussion"), the test instances and configurations used in the computational experiments are described, along with the decision support tool created for generating and managing these instances. Besides, a validation of the proposed mathematical model is also presented. The relevant results and an analysis of the robot's routes in four different scenarios are then analysed.

Finally, in Chapter 5 ("Conclusions"), the main conclusions of the project are highlighted, addressing the limitations encountered during the implementation of the mathematical model and suggesting directions for future work.

Complementing these chapters are the complementary sections such as the Abstract, Contents, Lists of Abbreviations, Tables, and Figures, Appendices, and Bibliography.

Chapter 2

Literature Review

This chapter explores the relevant literature to frame this research, taking a funnel approach until reaching the main focus of the problem. It begins with an overview of Combinatorial Optimisation Problems, moves on to Three-Dimensional Packing Problems and culminates in Order Picking Problems. The emphasis is on the latter part, where different order picking systems are discussed, with special emphasis on picker-to-parts and parts-to-picker systems. Furthermore, since the practical implementation of this project includes the involvement of Autonomous Mobile Robots in a warehouse environment, the applications of these robots, as well as other types of robots, in order picking systems will also be explored.

2.1 Combinatorial Optimisation Problems

This section briefly introduces the concept of Combinatorial Optimisation Problem (COP), which serves as the foundation for all of the theoretical work needed to develop this project. Concepts such as $\mathcal{NP}$-hard problems and approximate algorithms are outlined as subtopics.

The selection of the "optimal" arrangement or set of parameters to accomplish specific goals is a key component of many optimisation problems. These appear to fall into two categories: those in which the solutions are represented using continuous variables (also known as real-valued variables) and those in which the solutions are encoded using discrete variables. In the last case, Combinatorial Optimisation Problems can be found (Blum and Roli, 2003; Papadimitriou and Steiglitz, 1998).

One of the most active areas in operations research, computer science, and applied mathematics is combinatorial optimisation, also known as discrete optimisation. Besides that, Combinatorial Optimisation Problems can also be found in a wide range of fields, including number and graph theory, integer and linear programming, and artificial intelligence (Pardalos et al., 2013).

As the name implies, Combinatorial Optimisation Problems are optimisation problems in which the main goal is finding the optimal object from a limited or potentially countably infinite set of objects. Hence, a subset, a set, a permutation, a graph structure, or an integer number are common examples of this object (Blum and Roli, 2003; Papadimitriou and Steiglitz, 1998; Puchinger, 2005).

Speaking technically, COPs can be specified as follows:

Definition 2.1.1 *A Combinatorial Optimisation Problem can be either a minimisation or a maximisation problem and is defined by a set of problem instances (Pardalos et al., 2013; Puchinger, 2005).*

Definition 2.1.2 *An instance of a Combinatorial Optimisation Problem can be defined as P=(S,f), where S is the set of all feasible solutions and f is an objective function that has to be minimised or maximised. Each element of the set S can be considered a possible solution (Blum and Roli, 2003). This problem has:*

- *a set of variables:* $X = \{x_1, \dots, x_n\}$;
- *variable domains:* $D_1, \dots, D_n$;
- *variables with constraints.*

With,

$$\{S = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\} | v_i \in D_i, \\ s \text{ satisfies all the constraints}\} \tag{2.1}$$

$$f : D_1 \times \cdots \times D_n \rightarrow R^+ \tag{2.2}$$

When a COP is being solved, the main goal is to find a solution $s^* \in S$ *with a minimum objective function value,* $f(s^*) \leq f(s) \quad \forall s \in S$*, or with a maximum objective function,* $f(s^*) \geq f(s) \quad \forall s \in S$*. Additionally, it can be said that the set* $S^* \subseteq S$ *is referred to as the set of globally optimal solutions and* s^* *is referred to as the globally optimal solution of (S,f), while* $S^* = \{s \subseteq S | f(s) = f^*\}$ *represents the set of optimal solutions and* $f^* = f(s^*)$ *identifies the optimal cost (Blum and Roli, 2003; Puchinger, 2005).*

2.1.1 $\mathcal{NP}$-hard Problems

Some Combinatorial Optimisation Problems, in which finding a solution is a complex process, are known as $\mathcal{NP}$-hard (Puchinger, 2005; Ulungu and Teghem, 1994).

On the first hand, a decision problem P_d – problem that requires an input and, consequently, returns 0 or 1 (yes or no) as the output – is solvable in polynomial time when an algorithm can generate the proper solution "for any input of length n in a polynomial bounded number of steps" (Puchinger, 2005, p. 4). For instance, COPs that are solvable in polynomial time are special cases of linear programming (Pardalos et al., 2013). On the other hand, it can be claimed that $\mathcal{NP}$ refers to problems whose solutions may be obtained by algorithms. Besides that, if some problem can be polynomially transformed from $\mathcal{NP}$ to P_d, then a decision problem, belonging to $\mathcal{NP}$ ($P_d \in \mathcal{NP}$), is said to be $\mathcal{NP}$-complete, and it is said to be $\mathcal{NP}$-hard if a decision problem is at least as complex as any other problem in $\mathcal{NP}$. Overall, if the underlying decision problem is $\mathcal{NP}$-complete, then the optimisation problem is already $\mathcal{NP}$-hard (Puchinger, 2005).

2.1.2 Approximate Algorithms

The Cross-docking Problem, Scheduling Problem, Travelling Salesman Problem, Cutting and Packing Problem, and Quadratic Assignment Problem are examples of Combinatorial Optimisation Problems. To tackle COPs, complete and approximate algorithms have been explored, but over the years, it is clear that a strong emphasis has been placed on the development of approximation methods (Blum and Roli, 2003). Many approximation methods for solving $\mathcal{NP}$-hard problems, for instance, are frequently based on linear programming relaxations (Pardalos et al., 2013). From approximate algorithms, it is possible to obtain quality solutions in a significantly reduced period rather than optimal solutions, and these can be distinguished between constructive methods and local search methods. These two algorithms can be characterised as follows (Blum and Roli, 2003):

- Constructive Algorithms: Create solutions from the "ground up" based on the addition of components;
- Local Search Algorithms: Begin with an initial solution and iteratively seek to replace the existing solution with a better one located within a properly identified neighbourhood of the present solution.

Compared to local search algorithms, constructive algorithms frequently provide worse-quality solutions. Despite this, they are normally the fastest among the approximate methods (Blum and Roli, 2003).

According to Blum and Roli (2003), apart from constructive and local search algorithms, metaheuristic algorithms are a recent method that seeks to efficiently explore the search space to find optimal or near-optimal solutions. Algorithms such as Ant Colony Optimisation, Evolutionary Computation, which includes Genetic Algorithms, Iterated Local Search, Simulated Annealing, and Tabu Search are examples of metaheuristic algorithms.

2.2 Three-dimensional Packing Problems

This topic focuses on the 3D-PP. Furthermore, online, semi-online, and offline Packing Problems are identified. Within these, a focus is placed on semi-online and online problems.

The 3D-PP is an extension of the One- and Two-Dimensional Packing Problems, and it is also a $\mathcal{NP}$-hard problem. As a result, the ideal solutions are typically unknown or cannot be solved computationally (Faina, 2000; Li and Cheng, 1992). An efficient 3D-PP algorithm can reduce transportation costs and increase its importance due to environmental and economic benefits (Ali et al., 2022).

As such, the main goal of this type of problem is to reduce the total packing's height by packing orthogonally and orienting in all three dimensions a list of rectangular boxes, each with a determined length, width, and height, into a rectangular box with a fixed-size bottom and an unbounded height (Li and Cheng, 1990, 1992; Miyazawa and Wakabayashi, 1997).

The 3D-PP is concerned with the availability of various items and containers. According to Ali et al. (2022), two distinct paths can be recognised between Three-Dimensional Packing Problem investigations: offline and online problems (real-time problems). The first one refers to problems where all information about all input items is previously known. On the other hand, the second one is related to the fact that there is no advanced knowledge of the items' complete information; that is, the items that arrive one at a time must be packed immediately without knowing their features (Ali et al., 2022; Ha et al., 2017). In offline Packing Problems, the knowledge about the various input items is usually available beforehand. Online problems, on the other hand, do not have this advantage (Ali et al., 2022).

Despite the numerous practical applications of online problems in real-world scenarios, it is known that most research has concentrated on offline problems, and when it comes to online solutions, information is lacking (Ali et al., 2022).

In accordance with Ali et al. (2022), there is an intermediate class between online and offline problems that are called semi-online problems. This allows for some pre-processing operations. For instance, in some studies, the restrictions on the online environment are lifted so that items can be temporarily stored before being shipped.

2.2.1 Online Packing Problems

Online Packing Problems can be encountered in various real-world problems, including robotic or automatic container loading in warehouse storage (Ali et al., 2022; Ha et al., 2017).

By categorising the online algorithms into four different models based on the deletion strategy and repacking method (after packing, items can be slightly adjusted), researchers presented a classification for online 3D-Packing Problems (Ali et al., 2022). This classification comprehends the classic problem of online packing, the relaxed online packing model, dynamic packing, and fully dynamic bin packing (Berndt et al., 2020).

- Online packing problem – Items come in gradually and must be packed when they arrive, with no possibility of repacking already packaged items (Berndt et al., 2020).
- Relaxed online packing model – After packing, the items can be rearranged. Allows for the continuous movement of the previously packed items from one container to another as new input elements are introduced (Gambosi et al., 2000).
- Dynamic packing problem – Aside from insertion, items in this model can also depart over time (deletion strategy). Repacking is not allowed (Ali et al., 2022; Berndt et al., 2020).
- Fully dynamic packing problem – Repacking of previously packed items is allowed in dynamic bin packing. In this model, the items are arranged and transported in an online manner (Berndt et al., 2020).

2.2.2 Semi-online Packing Problems

A semi-online algorithm can be defined between offline and online Packing Problems. It can be used to perform various operations in each step, such as repacking finite numbers of items and pre-processing the items by placing orders based on size. This type of problem is slightly different from pure online ones. In addition to these, it allows the execution of other types of operations in each step, such as buffering them before packing them (Epstein and Kleiman, 2009). As stated by Ali et al. (2022), in real-world applications, "a subset of all upcoming items may be visible to the loading system" or can be stored temporarily in a buffer (p. 15).

It is essential to realise that bins become active or open once they receive their first item, and once they have been declared inactive or closed, the algorithm stops considering them for packing. A bin becomes empty if no items are assigned to it.

Moreover, while some online algorithms only allow a certain number of bins that can accept items at any given time throughout processing, others may keep all of the bins open for all items that have been placed in them (Epstein and Kleiman, 2009).

Wang and Hauser (2021) take into consideration another type of semi-online Packing Problems, or Quasi-online Packing Problems (QOP) as they usually refer to them, which also falls in the middle of offline and online Packing Problems. Their research presented two non-deterministic strategies for dealing with the various challenges that arise when implementing robot packing systems in automated warehouses. One of them is the already known semi-online problem, and the second one is a non-deterministically ordered packing variant. These two formulations are ideal for warehouses where the ultimate item set is known, but the sequence in which the items arrive is controlled by an uncontrollable element of the packaging system.

The non-deterministic version of the ordered packing system is known as the NDOP, which means that the container's feasibility is verified under various non-deterministic orders. However, the arrival sequence is not revealed until the packaging is carried out. This one requests the creation of a certificate proving the existence of a packing plan for each item order. In the quasi-online version, each object is packed before the next one is revealed. Besides, a partial observable packing is computed by making a map that chooses the appropriate location for each item. These two versions provide greater uncertainty compared to the offline packing problem, but when the item set is known, they want a single container with assured packing, which is better suited for certain fulfilment applications than the online Packing Problems (Wang and Hauser, 2021).

According to all the existing literature, despite not being an algorithm that has been widely studied, the semi-online algorithm is mainly found in the Bin Packing Problem (BPP). Gambosi et al. (2000) presented three sub-classes of semi-online Bin Packing Problems explored in the literature in the last decade. However, first, the roles of the "scheduler" and the "packer" are going to be established. The "scheduler" is tasked with producing the input list, while the "packer" is responsible for packing the items. In the other cases, the scheduler's role is trivial, as it merely gives the elements one by one and checks the end of the list. Hence, the three different types of semi-online BPP – the batched Bin Packing Problem, the dynamic Bin Packing Problem, and the c-repacking semi-online Bin Packing Problem – are going to be briefly explained next.

- Batched BPP – Occurs when the input list is divided into multiple batches. The "scheduler" will either give a new element or mark the end of the present batch in every step. The "packer" will then pack each batch as an offline list, which means that the earlier ones cannot be moved (Gambosi et al., 2000).

- Dynamic BPP – The "scheduler" can insert a new operation, or it can delete an earlier given operation. The number of bins the algorithm uses has been determined by the maximum number of non-empty bins that can be used at any stage of the packing process. A special type of this subclass is called the fully dynamic Bin Packing Problem. This occurs when the "packer" is allowed to repack the items in every step (Balogh et al., 2010; Gambosi et al., 2000).

- c-Repacking semi-online BPP – Allows the repacking of all the c elements in every step. The classical problem of online BPP returns when the case $c = 0$ is used (Gambosi et al., 2000).

2.2.3 Mixed-case Palletising Problems

The distribution and warehousing operations are vital to the success of a retailer's supply chain, and their efficiency depends on the proper utilisation of the available space and the optimal combination of boxes (de Carvalho and Elhedhli, 2022). In accordance with Tsai et al. (1993), palletising systems have traditionally loaded only boxes of the same size onto a single pallet. A diverse product mix of different box sizes must be placed on the same pallet and sent to retail destinations for operations such as food distribution.

The Mixed-case Palletising Problem, also known as the Distributor's Pallet Loading Problem, is an extension of the Three-dimensional Bin Packing Problem and a multi-objective multi-constraint optimisation problem (Balakirsky et al., 2010; de Carvalho and Elhedhli, 2022). This type of problem is found in research that is mostly concerned with robotics. It is a common problem that affects the transportation of orders in a given area, such as a warehouse or logistics operations. The most popular transportation method is pallets and containers, which are effectively three-dimensional packaging with side restrictions. Besides, the Mixed-case Palletising Problem (MCPP) and the Three-dimensional Bin Packing Problem (3D-BPP) share the goal of efficiently packing items into containers to minimise unused space. Both problems deal with packing boxes as items; however, in 3D-BPP the boxes are packed into a rectangular container, whereas in MCPP the boxes are placed onto a pallet. Therefore, this problem consists of finding the ideal arrangement of heterogeneous boxes in pallets or bins, in which the boxes should be positioned side by side and parallel to the container's edges. Thus, the characteristics of the container, the heterogeneity of boxes, and the packing objective are some of the factors that are considered when it comes to solving this problem (de Carvalho and Elhedhli, 2022). To further add, Balakirsky et al. (2010) stated that the type of packages that are being stacked is a factor that affects the packing strategy, and, on the other hand, pallet utilisation is directly related to shipping constraints.

One of the most challenging mixed-integer optimisation problems to solve is the 3D-BPP, and consequently the MCPP. The third dimension includes significant problems, such as vertical support, bin stability, and load bearing, even though it is a direct extension of the Two-dimensional Bin Packing Problems (2D-BPP) (Elhedhli et al., 2019). Bortfeldt and Wäscher (2013) defined five practical constraints. These include container-related constraints (weight limit and weight distribution constraints), item-related constraints (loading priorities, orientation, and stacking constraints), cargo-related constraints (complete shipment, allocation, and positioning constraints), and load-related constraints (stability and complexity constraints). However, as stated by de Carvalho and Elhedhli (2022), most of the work on the 3D-BPP lacks addressing these practical constraints, and when they are added, they make this type of problem even harder to solve. Since exact methods tend to encounter limitations when handling large cases, heuristics are often utilised to resolve practical problems. Zhao et al. (2016) defined two variants of heuristics: improvement and placement heuristics. The former focuses on searching for better ways to construct a layout, concerning the placement of boxes in layers, while the latter is used for seeking out better solutions.

2.3 Order Picking

In accordance with Bartholdi and Hankman (2019), the Order Picking (OP) Process is one of the most crucial warehouse operations, as it is labour-intensive and, mostly, defines the level of customer service. Therefore, this process consists of collecting items from the different warehouse locations in order to fulfil the customer's demands. Its main phases are: travelling to the storage location, which takes a lot of effort and adds no value; local search to find the exact location of the wanted items, which is essential to avoid mistakes in picking the wrong items, but also has no direct value; and, finally, reaching and picking the item, which is the only phase that adds value and is one of the warehouse activities that is resistant to automation.

On the one hand, in low-volume distribution processes, the customer's orders are small and urgent, and, consequently, require long distances and variable routes for each picking operation. In this type of distribution process, the challenge is to minimise the journeys made by the pickers. On the other hand, in high-volume distribution processes, the orders are larger and more standardised, which makes the picking process easier as the pick-ups can be performed in common routes. The main focus of high-volume distribution processes is on avoiding bottlenecks (Bartholdi and Hankman, 2019).

In accordance with Yoon and Sharp (1996), there are several order picking methods that can take place in a warehouse, such as single-order picking, batching and sort-while-pick, batching and sort after-pick, single-order picking with zoning, and batching with zoning. Thus, according to Gu et al. (2007), batching, sequencing, routing, and sorting are some basic steps that can be included in almost every order picking method.

Batching Problems

Order picking planning has a crucial component, which revolves around the batching problem. In this way, this problem consists of dividing the orders into batches in order to make the processes of picking, packing, and shipping more efficient within the established time windows. This type of problem is subject to performance criteria and constraints, such as picker effort, time slots associated with a pick wave, picker capacity, and time constraints associated with order due dates (Gu et al., 2007).

Gu et al. (2007) considers that the batching problems can have two segmentation levels: partitioning into time slots and partitioning of orders among the pickers. In this manner, the partitioning in time slots resembles a BPP and aims to balance the pick time between the different time slots. Although, this type of problem also has some setbacks, as it is only after the determination of the batch, assignment of that batch to the different pickers, and determination of the picker's routes through the warehouse, that it is possible to know how long it will take to pick that batch. Conversely, the partitioning of the orders between the pickers can be considered a variation of the Vehicle Routing Problem (VRP) (Gu et al., 2007). This type of problem represents a broad class of Combinatorial Optimisation Problems where customer stops are assigned to predetermined routes (Gu et al., 2007; Montoya-Torres et al., 2015). A fleet of vehicles is responsible for serving the customers, starting from a depot, servicing customers within a network, and returning to the depot upon completing their routes (Montoya-Torres et al., 2015). When an order is linked to a picker's route, the location of each Stock Keeping Units (SKU) is included in its route. Consequently, if the order is large and has many SKUs, the journey may become too long. Therefore, the primary goal of the VRP is to minimise the total distance travelled or the total time taken to finish the routes (Gu et al., 2007).

Sequencing and Routing Problems

The sequencing and routing problem finds the best picking sequence and routes. This type of problem can be considered as a Traveling Salesman Problem (TSP) in the context of a warehouse, as the locations of items for the picking or storing process are given. Additionally, usually in this type of problem, the main objective is minimising the total cost of handling the material. It is known that the sequencing and routing problem can be applied to various warehouse systems, such as conventional multi-parallel-aisle systems, man-on-board AS/RS systems, unit-load AS/RS systems, and carousel systems (Gu et al., 2007).

Sorting Problems

When several orders are picked together and the sorting process can be done during or after picking, this is known as a Sorting Problem. Thus, when sorting is done during picking, the process proves to be simple and usually only involves adjusting the time it takes to extract the items. However, when sorting is done after picking, a downstream sorting system is required, which includes an accumulation conveyor, a re-circulation conveyor, and exit lanes, all working together during a picking wave. In this process, the items are placed on the re-circulation conveyor after the sorting of the previous wave has been completed. Orders are assigned to specific sorting lanes based on lane order rules. The items circulate until they can enter their assigned lane, bypassing it if the previous order has not finished sorting. Once sorted, the items are picked, checked, packed, and dispatched. Key decisions include releasing waves and assigning orders to lanes to optimise the sorting process (Gu et al., 2007).

2.3.1 Order Picking Methods

de Koster et al. (2007) mentions that a warehouse can have multiple order picking systems. In turn, these systems can have the intervention of humans or automatic machines. In order picking, when there is a process that is carried out by a human, there are three systems: Picker-to-parts; Put System (also known as Order Distribution System); and Parts-to-picker. On the other hand, when automatic machines are used, there are two methods: Automated Picking and Picking Robots. Figure 2.1 shows a schematic of all the existing picking methods, as well as their variants.

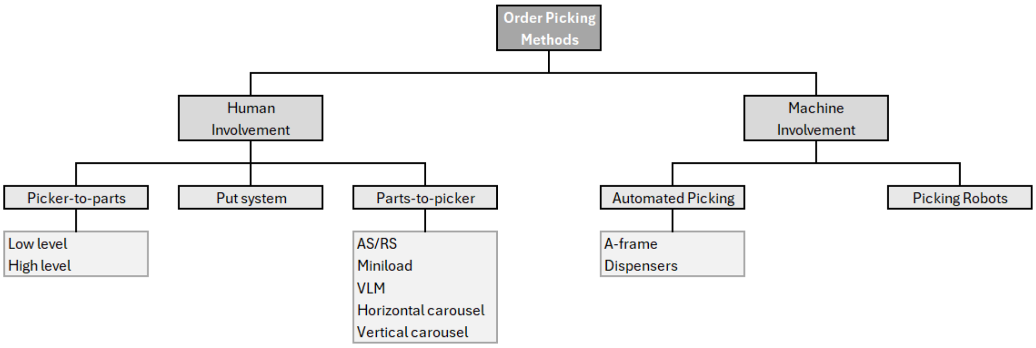


Figure 2.1. Order picking methods (based in de Koster et al. (2007)).

Although there are several order picking systems, this subsection focuses on picker-to-parts and parts-to-picker systems.

As de Koster et al. (2007) mentions, the most common systems that are implemented in warehouses are the picker-to-parts order picking systems. In this respect, in a picker-to-parts system, the order picker moves along the aisles to manually pick items from the various warehouse locations. This method includes a series of manual tasks that the operator needs to carry out sequentially, such as preparing the pick list, moving to the storage location, locating the item, and finally picking the correct quantity of items (de Koster et al., 2007; Vijayakumar and Sgarbossa, 2021).

Two types of picker-to-parts systems can be distinguished: low-level picking and high-level picking. In the first one, the order pickers move around the various storage racks or bins until they find the location of the required items and pick them up. In the second system, also known as the man-aboard order-picking system, the pickers move around in a lifting order-pick truck or crane to collect the requested items. This structure stops exactly where the requested item is and waits until the picker has finished collecting it. In turn, this system is implemented in warehouses that have high storage racks (de Koster et al., 2007).

In the second method mentioned, the parts-to-picker system, the items are transported from the various storage locations to the order picker using Automated Storage and Retrieval Systems (AS/RS), miniload, Vertical Lift Modules (VLM), and carousel racks (de Koster et al., 2007; Vijayakumar and Sgarbossa, 2021).

The most common type of parts-to-picker system is an AS/RS, also called a unit-load or end-of-aisle order-picking system. Therefore, the process typically employs aisle-bound cranes, which pick up unit loads, either in the form of pallets or bins–the latter is often referred to as "mini-load" systems–and deliver them to an assigned picking station, at which an order picker picks a desired quantity of the product, and then the remaining load is stored again in its storage location. Furthermore, the automated crane, sometimes called a Storage and Retrieval (S/R) machine, operates in single, dual, and multiple command cycles. In a single-command cycle, the crane moves either a load from the depot to storage or retrieves a load from storage to the depot. This includes placing a load into storage in one command cycle and then retrieving another in dual command mode. In multiple-command cycles, a S/R machine having more than one shuttle will be able to handle multiple loads by picking and dropping off several items in one operation (de Koster et al., 2007).

The main objective of a parts-to-picker picking order system is to automatically transport the SKUs to the pickers so that they can only focus on the packing activities (Boysen et al., 2017).

2.3.2 Order Picking Process assisted by AMRs

Over the last decades, materials handling technology has advanced rapidly, with one of the main advances being the transition from Automatic Guided Vehicles (AGV) to Autonomous Mobile Robots (Fragapane et al., 2021).

For many years, AGVs were the most frequently used automated systems in sectors such as manufacturing, warehousing and container terminals (Le-Anh and De Koster, 2006). Although, nowadays, AMRs are increasingly present in industrial, healthcare, hospitality, security, and even domestic environments, performing a wide variety of functions. In addition to tasks, such as loading machinery and transport, they also act as assistance systems, collaboratively interacting with humans (Fragapane et al., 2021).

In warehouses, AMRs mostly perform activities related to the picking order process. These robots transport small containers inside the picking areas to assist operators in picking order activities. Therefore, their operation consists of moving to the location where the operator should pick an item, and after it is picked, they autonomously move to the next location where another item should be picked. Whenever the items of a picking order have all been collected, the robot moves on alone to the packing area, where the containers that hold the different items collected are emptied, and then, if applicable, a new set of orders is given to the AMR (Fragapane et al., 2021).

These autonomous robots can perform material handling activities, such as retrieving, moving, transporting, sorting, collaborative order picking activities with operators, and full-service operations, in which they can have robot arms and can make the picking and fetching process by themselves (Hernández et al., 2018).

This type of robot has many advantages for industrial environments, as they can use intelligent and cognitive methodologies and technologies to increase flexibility and productivity, and they interact with humans or other AMRs as a group. Besides, they have quick and cost-effective adaptation in industries, and they have the ability to recover from failures, guaranteeing operational resilience (Hernández et al., 2018).

2.3.3 Studies on robot-assisted Order Picking Systems

This section presents a review of several studies carried out in recent years on the involvement of robots in order picking systems, with an emphasis on picker-to-parts systems. The review covers recent research exploring the role of robots in the automation and optimisation of these processes, highlighting the main advances, challenges, and benefits associated with the implementation of robotic technologies in warehouse and logistics environments.

Hernandez et al. (2017) conducted a study that focused on the development of intelligent service robots for office-like environments, where tasks such as picking up and delivering items like mail, goods, or trash need to be performed. These tasks are complex due to the need for robots to navigate around static and dynamic obstacles while optimising for multiple goals, including energy efficiency. The research proposed an autonomous multi-goal path planning system based on a modified Lin-Kernighan Heuristic (LKH) algorithm to handle these tasks in mobile robotics. The system integrated non-Euclidean distances and Hamiltonian paths for efficient navigation, using a Pioneer 3DX mobile robot. It introduced a client-robot system where clients request pickup or delivery services, and the robot continuously plans a Hamiltonian path to fulfil the requests and return to its base. To validate the system, the authors compared the performance using two key metrics: distance travelled and time elapsed. The results of the LKH algorithm (with non-Euclidean distances) are compared to approaches such as nearest goal selection, random selection, and the LKH algorithm with Euclidean distances, demonstrating the effectiveness of the proposed solution.

Žulj et al. (2021) studied an AMR-assisted Order Picking Problem (AOPP) in a warehouse environment where human pickers collaborate with Autonomous Mobile Robots to improve order picking efficiency. The study taook place at a real-world logistics centre of a German automotive manufacturer, where customer orders are batched and AMRs transport the collected items to a central depot. The study aimed to minimise the total tardiness of customer orders, which occurs when orders are not picked up by their due dates. The research introduced new mathematical models for optimising the batching and sequencing of orders. It also proposed a two-stage heuristic approach to solve the AOPP, consisting of an Adaptive Large Neighborhood Search (ALNS) for batching orders and a modified Nawaz, Enscore, and Ham Heuristic (NEH) for sequencing batches. In this way, the authors conducted extensive experiments to evaluate the performance of the proposed methods, comparing different scenarios such as varying AMR fleet sizes and travel speeds. Furthermore, the study provided key insights, showing that increasing the AMR fleet size or speed can significantly reduce tardiness. Besides that, it demonstrated that a slight increase in AMR speed leads to larger improvements in efficiency than simply expanding the fleet. Finally, the research suggested that AMR-assisted picking systems are a promising direction for practical warehouse optimisation, especially when combined with advanced storage assignment strategies and alternative warehouse layouts.

The research conducted by Pugliese et al. (2022) focused on a hybrid picker-to-parts order picking system, with human operators cooperating with Autonomous Mobile Robots in a two-block warehouse layout based on real data from a grocery warehouse. This study aimed to minimise the human operator's time spent on activities that add no value. In this sense, novel mathematical models, based on a Mixed Integer Linear Programming Formulation, were developed in order to determine an optimal coordination of pickers and AMRs. The developed mathematical models were optimised using black-box solvers. Furthermore, this study considered two alternative approaches: one in which opposite sub-aisles share handover locations; and the other in which each approach has its own separate location for handovers. Therefore, the experimental analysis showed that, on the one hand, the use of separated handover locations is more efficient for smaller order volumes; on the other hand, shared handover locations are effective in cases when the number of items increases, since reduced waiting times for carts outweigh the negative effects of longer initial loading times. Thus, the shared handover layout proved to be the most efficient and effective for warehouses of high item volume.

Srinivas and Yu (2022) conducted a study that addressed the increasing need for speed and flexibility in order fulfilment by exploring a collaborative human-robot order-picking system where humans retrieve items and Autonomous Mobile Robots transport them to the depot. The research identified three key sub-problems that impact delivery performance, such as order batching, batch assignment and sequencing, and picker-robot routing. This research is considered to be the first to jointly address these sub-problems in a multi-picker-multi-AMR system. Hence, the authors developed an optimisation model that aimed to minimise the total order tardiness and propose a simulated annealing algorithm with an ALNS and an optimisation-based restart strategy to solve large instances. The results highlighted the impact of factors such as team composition, AMR speed, AMR capacity, and warehouse layout on picking efficiency.

The study presented by Löffler et al. (2023) explored the coordination of multiple Autonomous Mobile Robots and human pickers in robot-assisted order picking systems to minimise the makespan. The research introduced a heuristic method that managed the requirements of large e-commerce fulfilment centres, solving instances with over a thousand picking positions. In addition to this, the study analysed factors such as stochastic picking times, speed differences between AMRs and pickers, and zoning strategies. The findings revealed that separating the workforce into smaller subgroups can mitigate the ripple effect caused by delays in synchronised schedules. Moreover, AMRs and pickers must have similar travel speeds, as slower AMRs negatively impact the picker's performance. Finally, the study stated that AMR-assisted systems significantly reduce the makespan of the order picking problem in question.

Research conducted by Vijayakumar and Sobhani (2023) focused on optimising picker-to-parts order picking systems, a labour-intensive process responsible for 55% of warehouse costs in e-commerce. To address this, the research explored the integration of Pick and Transport Robots (PTRs), aligning with the human-centric focus of Industry 5.0, which aims to enhance working conditions rather than replace workers. The study developed a mathematical model to improve productivity, quality, and worker well-being in OP systems using PTRs. The model was tested with data from a real-life company, and results showed that assigning heavy tasks to robots increased system productivity, reduced errors, and improved the health of workers. Key challenges addressed include the need for separate zones for robots and workers, determining the optimal number of robots and workers, and addressing robot limitations in handling irregularly shaped items. The optimisation model minimised operational costs, worker energy expenditure, and error rates, providing valuable insights for improving OP system performance.

Zhao et al. (2024) developed a study that addressed the optimisation of robot-assisted goods-to-person order picking systems in smart warehouses, where human pickers collaborate with robots that transport storage pods to a picking station. The challenge lay in the interdependent tasks of pod selection, robot scheduling, and manual picking, which together determined the system's efficiency. To optimise this human-robot collaboration, a Mixed Integer Programming Problem was developed for small-scale problems, while a metaheuristic approach–an ALNS combined with a Tabu Search (TS) algorithm–was proposed for large-scale instances. The system studied operated with predetermined warehouse layouts, including pod and robot locations, and assumed uniform robot speed, predefined picking times, and negligible robot path conflicts. The proposed optimisation model for minimising makespan filled a gap in the literature by jointly optimising the three core tasks. Experimental results highlighted the superiority of the proposed algorithm over existing methods, demonstrating its potential to enhance operational efficiency in robot-assisted warehouses.

Chapter 3

Problem Description and Mathematical Model

This chapter plays a key role in the development of this project. It begins with a general overview of the order delivery process at a retail company and then provides a more detailed description of the various stages involved in the process. It is important to note that, although Section 3.1 describes the entire process, the project only focuses on the dynamic picking process performed by Autonomous Mobile Robots. Building on this, a mathematical model was developed to optimise the order picking process by minimising the total operational time needed. Therefore, in Section 3.2, the model's logic is firstly introduced through a graphical representation of the problem's decomposition, followed by the presentation of the mathematical model, which defines the sets, parameters, decision variables, lexicographic objective function, and constraints.

3.1 Order Delivery Process

This section provides an overview of the process involved in the understudy problem. It is followed by a detailed description of each stage, supplemented by illustrative figures for clarity and ease of understanding.

Before examining each stage of the process in detail, it is important to have a clear comprehension of the main elements involved in this scenario. The following key concepts are fundamental to a more effective analysis:

- **Robots in the warehouse environment**

  In the context of the process under study, several robots are deployed and operate in the warehouse environment. The robots selected are the Autonomous Mobile Robots (AMRs), a type of robot that can orient and move without human control. They are essential elements that play a central role in various operations, such as the loading of pallets, their transport and the unloading of items.

- **Diversity of customers**

  This process also involves interaction with a wide range of customers. These represent the stakeholders who demand different items from the retailer.

- **Variety of Pallet Types**

  The process uses different types of pallets to accommodate the variety of items and customer requirements. These pallet types are identified by letters, such as 'A', 'B', and 'C', according to the items they contain.

Once these fundamental concepts are well understood, a broad description of the process is provided. At first glance, the process can be summarised as follows:

1. AMRs pick pallets from different locations in the warehouse according to a picking sequence;
   - Each pallet type contains items that are available in stock (pallets with identical items);
   - The stock of each pallet type is dynamic (stock is continually replenished based on demand).
2. These pallets are transported to a dynamic picking area and their items are distributed onto empty pallets to fulfil the orders of each customer;
   - These pallets are located in various lanes that make up this area;
   - Each lane has the customer's identification;
   - Each lane is associated with a single customer, but a customer can be associated with more than one lane.
3. The AMRs repeat the process of fulfilling the pallets until the customer's orders are completed;
   - A customer can only be served by one AMR at a time;
   - A pallet can only be transported by one AMR at a time;
   - Each AMR can only move a maximum of one pallet at a time.
4. Once a customer's order has been completed, the respective full pallet(s) are transported to a marshalling area;

5. Later, the pallets in the marshalling area are loaded onto their allocated trucks;

6. The trucks transport the various pallets to the destinations requested by the customers.

Figure 3.1 displays the order delivery process. Each relevant area is associated with a specific designation, and the arrows represent the expected flow of movement by all parties involved in the process.

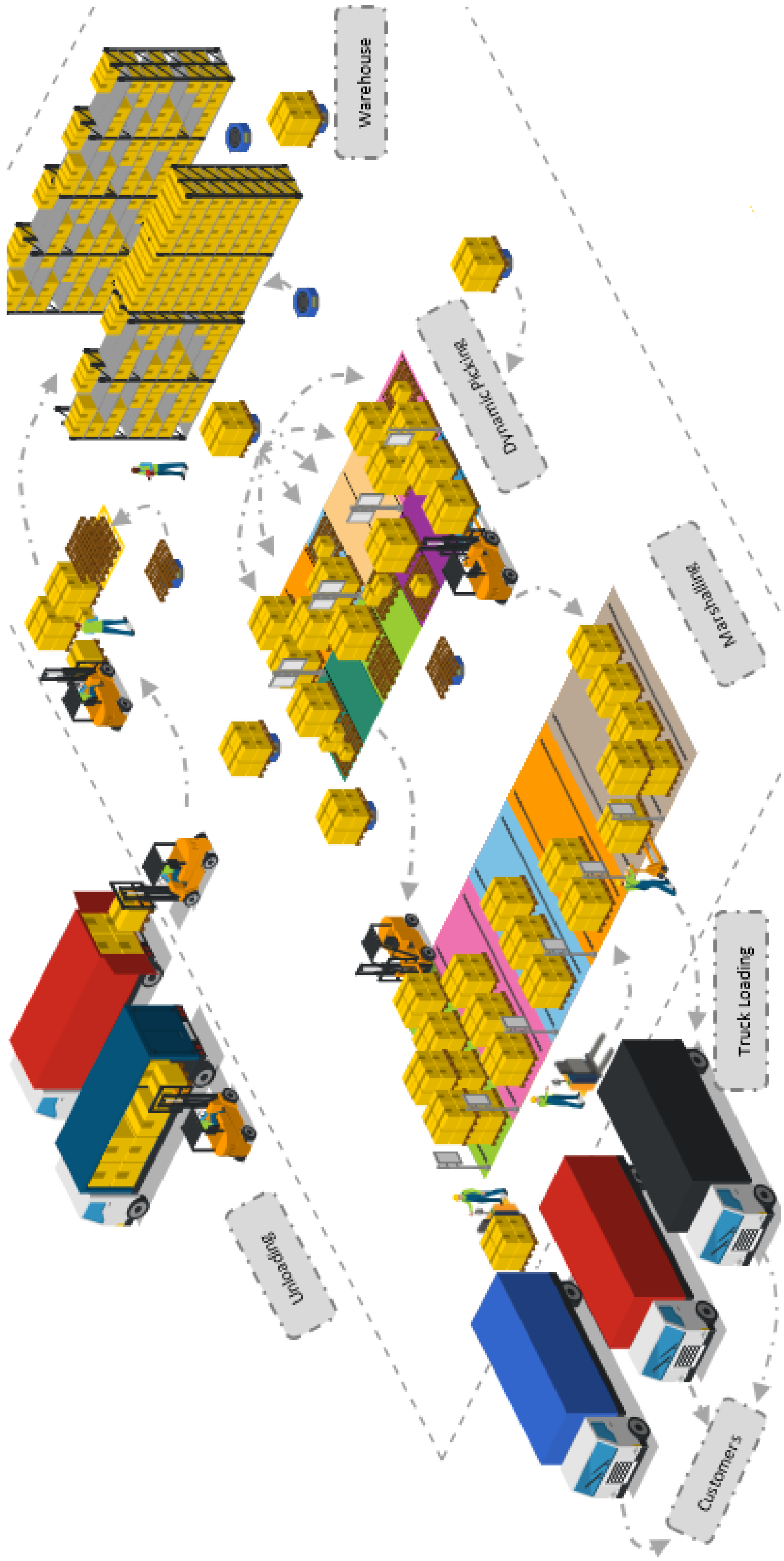


Figure 3.1. Order Delivery Process.

3.1.1 Detailed description of the process

This subsection provides a detailed overview of the processes involved, exploring each phase sequentially – from the initial unloading of items to the final loading of the delivery trucks – and preparing the stage for the subsequent analysis of the mathematical model developed to optimise the dynamic picking process.

By highlighting the key activities carried out by the AMRs, the movement of pallets, and the coordination between different areas of the warehouse, it offers a holistic view that enables an in-depth analysis of the order delivery process.

Unloading Process

The process starts with the step shown in Figure 3.2. Therefore, several trucks arrive at the unloading area to discharge the items they have brought from suppliers. Then, these items are unloaded and arranged on pallets according to their type. Subsequently, the new pallets are transported to the warehouse.

Figure 3.2. Process of unloading items coming from the suppliers.

Warehouse Operation

Figure 3.3 illustrates the warehouse where the different items required to fulfil customer orders are stored and from where the AMR's distribution process begins. In addition, these robots can transport pallets to the dynamic picking area and return them to their respective locations in the warehouse if they still have items and are not immediately required by another customer. Empty pallets are stored in a specific warehouse area when all items have been fully distributed.

Figure 3.3. Warehouse dynamics.

Dynamic Picking Process

As previously stated, the dynamic picking area, depicted in Figure 3.4, consists of multiple lanes, with each lane being assigned to a customer. Depending on the size of the order, a customer can be assigned to multiple lanes. Once an order is fulfilled, the lane assigned to a particular customer is reassigned to another customer in a customer rotation system. In each customer's lane, empty pallets are fulfilled in a predetermined building sequence with items transported by the AMRs.

Figure 3.4. Dynamic Picking Area.

The process represented in Figure 3.5 is central to the problem addressed in this work. Here, several AMRs pick homogeneous pallets (containing identical items) from various warehouse locations and take them to the preparation zone, a dynamic picking area, where multiple orders are prepared simultaneously.

Figure 3.5. Dynamic Picking Process.

It is essential to take into account how AMRs distribute items, as their movement follows the sequential order of assembly for each customer's pallet:

- Each customer has a predefined sequence that determines the order of delivery of the items required to meet their demands.

- For example, a given customer has ordered items in the following sequence: 'B-A-E-D-C'. As a result, the delivery order conducted by AMRs should follow this sequence, starting with the items on pallet type 'B', followed by items of pallet type 'A', then 'E', 'D', and finally items of pallet type 'C'. The items on pallet type 'E', for example, can only be delivered after the items on types 'B' and 'A' have been completely delivered. Each item can only be placed if all the previous items present on the precedence order have been placed. In this way, the delivery sequence must be strictly adhered to in order to guarantee compliance with the customer's order.

- Once the items on the pallet that an AMR is carrying have all been distributed, the robot returns to the warehouse to collect a new pallet of the same type.
  - For instance, if an AMR is delivering items from pallet 'A' and the items on that pallet run out, it needs to pick up another pallet of the same type to continue distributing the necessary items.

Marshalling and Truck Loading Processes

After a customer order has been met, the corresponding pallets are transported from the dynamic picking area to a marshalling area, illustrated in Figure 3.6. This area also comprises multiple lanes, with each lane being linked to a specific truck.

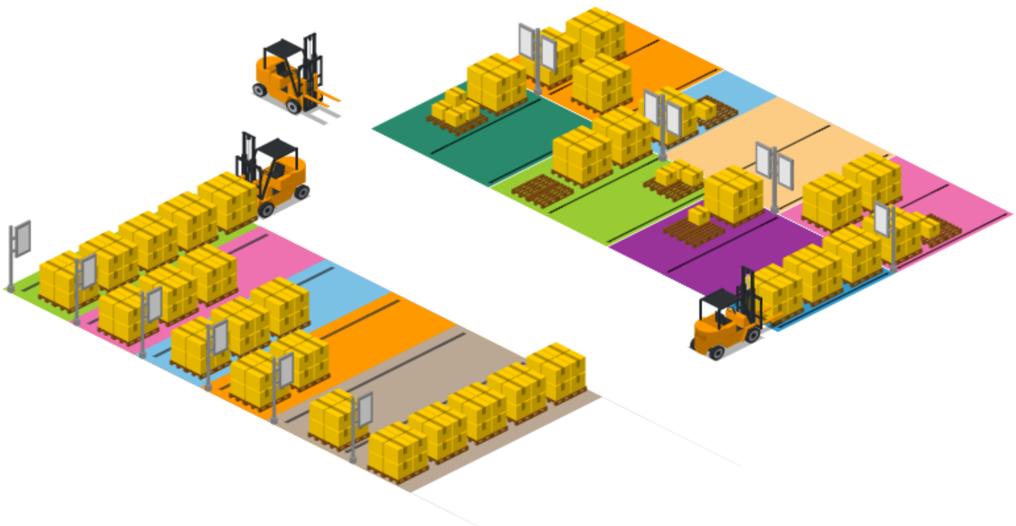


Figure 3.6. Transport of customers' orders from the dynamic picking area to the marshalling area.

Once the list of orders assigned to a particular truck is all in the marshalling area and the same truck is in the loading dock, the pallets are loaded onto it, as shown in Figure 3.7.

Finally, each truck goes to the different customers to deliver their orders.

Figure 3.7. Truck Loading.

3.2 Mathematical model for the case study

In this section, a mathematical model developed to optimise an order picking problem, formulated as a MIP model with a discrete time approach, is presented. The mathematical model is designed to optimise the process and meet the specifications outlined previously. Additionally, a visual representation detailing the decomposition of the problem is provided and thoroughly explained.

3.2.1 Graphical explanation of the problem decomposition

In order to facilitate the understanding of the logic behind the construction and operation of the mathematical model, an example is illustrated in Figure 3.8. Hence, in this figure, some essential components can be identified, including:

- Four customers;
- Three time slots;
- Three types of pallets (labelled as 'A', 'B', and 'C');
- Three AMRs;
- Three units available in the warehouse stock for each pallet type.

Time Slots

To build the mathematical model, it was essential to consider the concept of time slots. A time slot denotes a discrete and predefined time window during which operational decisions are made. In this respect, it is considered to be a unit of time in which it is determined whether the delivery of a pallet to a customer occurs, as well as the allocation of each pallet and AMR at each instant.

Predefined Delivery Sequence

As mentioned before, the movement of AMRs is dictated by the predefined delivery sequence of the items requested by the customers. On this basis, in each time slot, each AMR, according to its availability, delivers items from the different pallet types following the sequential order. By analysing Figure 3.8, it can be concluded that Customer 1 requested a sequence of items of type 'B', followed by type 'A', and Customers 2 and 3 submitted more complex requests. In this regard, Customer 2 demanded a sequence starting with items of type 'A', then type 'B', and finally type 'C', while Customer 3 requested items of types 'C', 'A', and 'B', according to this sequence. Lastly, Customer 4 solicited only items of type 'C'.

Pallet Identification

Another important aspect to consider is that each pallet is identified by a letter corresponding to its type and a unique number. For example, in Figure 3.8, the first pallet of type 'A' to be distributed is labelled as 'A_2'. However, if the items on that pallet are depleted, the next one to be collected can be 'A_1' or 'A_3', as long as it still has sufficient items to meet customer demand. The selection of the pallet does not follow a strict numerical sequence. Instead, it is chosen based on availability in the warehouse, regardless of its identification number. In this manner, in each time slot, only one pallet with a specific identification (e.g. pallet 'A_1') can be displayed, and the simultaneous presence of multiple pallets with the same identification in the same time slot is not permitted. Nevertheless, various pallets of the same type can move simultaneously, as long as they have distinct identification numbers.

Pallet Stock

The number of pallets of a given type in stock is determined by the demand for them, which means that a virtually unlimited number of pallets of the same type is available. However, it is important to note that the quantity of items contained in each pallet type is limited by its capacity.

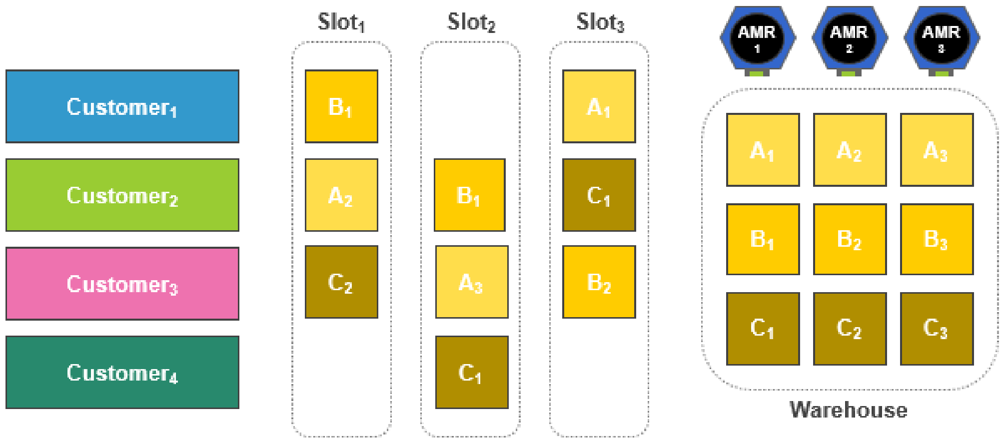


Figure 3.8. Visual representation of the model's construction logic.

3.2.2 Mathematical model

Sets

In the context of the problem under study, five sets have been established. As a result, set A includes the multiple AMRs that perform essential functions in the picking cycle, interacting in the pallet flows between the warehouse and the dynamic picking area. On the other hand, set C includes all the customers who have placed orders and thus occupy positions at the various locations in the dynamic picking area. It is important to realise that this mathematical model is based on the premise that each customer is assigned to a single lane in the dynamic picking area. In this manner, each customer has placed only one order, with no redistribution of new customers between lanes. Additionally, set P contains all the pallet types that may be involved in the process, and set N comprises the number of pallets available in stock for each pallet type needed to satisfy the demand of all customers. Finally, set S includes the multiple time slots that are needed to complete the entire delivery process.

Table 3.1 shows the various definitions of the sets.

Table 3.1. Sets of the mathematical model

A	Set of AMRs: $a = \{1, \ldots, A\}$.
C	Set of customers: $c = \{1, \ldots, C\}$.
P	Set of pallet types ordered: $p = \{1, \ldots, P\}$.
N	Set of pallets in stock for each pallet type: $n = \{1, \ldots, N\}$.
S	Set of time slots: $s = \{1, \ldots, S\}$.

Parameters

Based on the definition of the previously established sets, four parameters can be identified. At first glance, the parameter $demand_{cp}$ is included to represent a customer's demand for items of a given pallet type. In another aspect, the parameter $capacity_p$ is specified as the maximum quantity of items a pallet can hold. The parameter $order_{cp}$ refers to the sequence in which the AMRs must deliver the items of a pallet type to a customer. This predefined delivery order must be strictly followed by the robots and, consequently, guides their movements during the transport and pallet allocation cycle. Lastly, the parameter M is a constant, representative of a very large positive number, which must be at least greater than the maximum number of required items.

Table 3.2 briefly summarises the various fundamental parameters that make up this model.

Table 3.2. Parameters of the mathematical model

$demand_{cp}$	Demand from customer $c \in C$ for items of pallet $p \in P$.
$capacity_p$	Maximum number of items that a pallet $p\ inP$ can contain.
$order_{cp}$	Order of the items from pallet $p \in P$ in relation to the other items requested on the pallet of customer $c \in C$.
M	Constant that represents a large positive number.

Decision Variables

Within the framework of the optimisation problem under investigation, seven decision variables were formulated, including integer, binary, and auxiliary variables.

Among the integer variables, C_{max} represents the problem's makespan, i.e., the total time needed to complete all deliveries. Furthermore, y_{pnsac} represents the number of items of a pallet type delivered to a customer.

With regard to the binary variables, the decision variable x_{pnsac} is related to the decision of delivering or not the items of a pallet type to a customer at a given instant of time. In addition, u_{pn} points out which pallets were consumed.

There are three binary auxiliary variables: w_{pcs}, k_s, and z_{pns}. Variable w_{pcs} specifies whether a pallet is delivered to a customer up to a specific time slot; meanwhile, k_s verifies whether any delivery takes place in a time slot. Last but not least, z_{pns} indicates whether there has been a detachment between an AMR and a pallet that it has already transported; that is, when a pallet is moved again, it checks whether there has been a change of AMR.

On the basis of the definitions presented, Table 3.3 displays the various decision variables in a concise and organised manner, outlining their functions in the context of the mathematical model.

Table 3.3. Decision Variables of the mathematical model

C_{max}	Makespan.
y_{pnsac}	Quantity of pallet $p \in P$ (number $n \in N$) delivered in time slot $s \in S$ by AMR $a \in A$ to customer $c \in C$.
x_{pnsac}	Binary variable, equal to 1 if pallet $p \in P$ (number $n \in N$) is delivered in time slot $s \in S$ by AMR $a \in A$ to customer $c \in C$.
u_{pn}	Binary variable, equal to 1 if pallet $p \in P$ (number $n \in N$) is used.
w_{pcs}	Auxiliary binary variable, equal to 1 if pallet $p \in P$ is delivered to customer $c \in C$ up to time slot $s \in S$.
k_s	Auxiliary binary variable, equal to 1 if any delivery takes place in time slot $s \in S$.
z_{pns}	Auxiliary binary variable, equal to 1 if pallet $p \in P$ (number $n \in N$), in time slot $s \in S$, is transported by a different AMR.

Objective Function

The developed mathematical model has a three-objective function. In this way, the objective function is lexicographic, prioritising the minimisation of total distribution operation time. It also aims to minimise the number of pallets used and unnecessary movements of AMRs while maintaining operational efficiency.

The lexicographic approach was chosen to prioritise and solve multiple objectives, one at a time, according to their order of hierarchy. Therefore, each objective is only considered after the previous one has been completely optimised.

The lexicographic formulation (3.1) of this mathematical model has three main objectives, within which, firstly, $f_1(x)$ is minimised, then $f_2(x)$, and, finally, $f_3(x)$.

$$\text{Minimize} \quad (f_1(x); f_2(x); f_3(x)) \tag{3.1}$$

Hence, the multi-objectives, $f_1(x)$, $f_2(x)$, and $f_3(x)$, are detailed as follows:

$$f_1(x) = C_{max} \tag{3.2}$$

$$f_2(x) = \sum_{p=1}^{P} \sum_{n=1}^{N} u_{pn} \tag{3.3}$$

$$f_3(x) = \sum_{p=1}^{P} \sum_{n=1}^{N} \sum_{s=2}^{S} z_{pns} \tag{3.4}$$

The first objective (3.2) minimises the makespan of the problem. In this way, the total time needed until the last delivery is measured by the total number of time slots that are needed. The second one (3.3) minimises the number of pallets used. Lastly, the third objective (3.4) minimises the mismatching of AMRs with the pallets they transport in consecutive time slots.

Constraints

Based on the previously defined sets, parameters, lexicographic objective function, and decision variables, the following constraints were formulated:

$$\sum_{p=1}^{P}\sum_{n=1}^{N}\sum_{c=1}^{C} x_{pnsac} \leq 1, \qquad \forall \quad s \in S, \quad a \in A \tag{3.5}$$

$$\sum_{p=1}^{P}\sum_{n=1}^{N}\sum_{a=1}^{A} x_{pnsac} \leq 1, \qquad \forall \quad s \in S, \quad c \in C \tag{3.6}$$

$$\sum_{n=1}^{N}\sum_{a=1}^{A}\sum_{c=1}^{C} x_{pnsac} \leq 1, \qquad \forall \quad p \in P, \quad s \in S \tag{3.7}$$

$$y_{pnsac} \leq x_{pnsac} \cdot M, \qquad \forall \quad p \in P, \quad n \in N, \quad s \in S, \quad a \in A, \quad c \in C \tag{3.8}$$

$$x_{pnsac} \leq y_{pnsac}, \qquad \forall \quad p \in P, \quad n \in N, \quad s \in S, \quad a \in A, \quad c \in C \tag{3.9}$$

$$\sum_{n=1}^{N}\sum_{s=1}^{S}\sum_{a=1}^{A} y_{pnsac} = demand_{cp}, \qquad \forall \quad p \in P, \quad c \in C \tag{3.10}$$

$$\sum_{s=1}^{S}\sum_{a=1}^{A}\sum_{c=1}^{C} y_{pnsac} \leq capacity_{p}, \qquad \forall \quad p \in P, \quad n \in N \tag{3.11}$$

$$w_{pc1} = \sum_{n=1}^{N}\sum_{a=1}^{A} x_{pn1ac}, \qquad \forall \quad p \in P, \quad c \in C \tag{3.12}$$

$$w_{pcs} \geq w_{pc(s-1)}, \qquad \forall \quad p \in P, \quad c \in C, \quad s \in S\backslash\{1\} \tag{3.13}$$

$$w_{pcs} \geq \sum_{n=1}^{N}\sum_{a=1}^{A} x_{pnsac}, \qquad \forall \quad p \in P, \quad c \in C, \quad s \in S\backslash\{1\} \tag{3.14}$$

$$w_{pcs} \leq w_{pc(s-1)} + \sum_{n=1}^{N}\sum_{a=1}^{A} x_{pnsac}, \qquad \forall \quad p \in P, \quad c \in C, \quad s \in S\backslash\{1\} \tag{3.15}$$

$$\sum_{n=1}^{N}\sum_{a=1}^{A} x_{pnsac} \leq \sum_{\substack{p_1 \in P \\ order_{cp_1} = order_{cp} - 1}} \sum_{s_1=1}^{s} w_{p_1 c s_1},$$
$$\forall \quad p \in P : order_{cp} > 1, \quad s \in S, \quad c \in C, \quad p \neq p_1 \tag{3.16}$$

$$k_s \geq \frac{\sum_{p=1}^{P}\sum_{n=1}^{N}\sum_{a=1}^{A}\sum_{c=1}^{C} x_{pnsac}}{M}, \qquad \forall \quad s \in S \tag{3.17}$$

$$C_{max} \geq s \cdot k_s, \qquad \forall \quad s \in S \tag{3.18}$$

$$z_{pns} \geq (x_{pn(s-1)ac} + x_{pnsa_1c_1}) - 1, \qquad \forall \quad p \in P, \quad n \in N,$$
$$s \in S \backslash \{1\}, \quad a, a_1 \in A, \quad c, c_1 \in C, \quad a \neq a_1, \quad c \neq c_1 \tag{3.19}$$

$$u_{pn} \geq x_{pnsac}, \qquad \forall \quad p \in P, \quad n \in N, \quad s \in S, \quad a \in A, \quad c \in C \tag{3.20}$$

$$C_{max}, y_{pnsac} \in \mathbb{N}_0, \qquad \forall \quad p \in P, \quad n \in N, \quad s \in S, \quad a \in A, \quad c \in C \tag{3.21}$$

$$x_{pnsac}, u_{pn}, w_{pcs}, k_s, z_{pns} \in \{0,1\}, \qquad \forall \quad p \in P,$$
$$n \in N, \quad s \in S, \quad a \in A, \quad c \in C \tag{3.22}$$

Constraints (3.5), (3.6), and (3.7) ensure that the fundamentals of the problem are respected: in a time slot, an AMR can only serve one customer, a customer is only served by a single AMR, and a pallet is transported by a single AMR (uniqueness of pallets). Subsequently, Constraints (3.8), (3.9), (3.10), and (3.11) ensure that the delivery process is optimised, customer demands are met efficiently, and the pallet's capacities are respected. Constraint (3.8) sets an upper limit for the number of items on a pallet that can be delivered to a customer, linking y_{pnsac} with x_{pnsac}. Additionally, Constraint (3.9) ensures that when an AMR does not deliver any item, it is not mistakenly considered that a delivery has taken place. Constraint (3.10) guarantees that each customer's demand is fully met, and Constraint (3.11) is centred on ensuring that the total quantity of items delivered from a pallet does not exceed its capacity; otherwise, the robot needs to pick another pallet of the same type with enough items to fulfil the customer's demand. Moreover, Constraints (3.12), (3.13), (3.14), and (3.15) work as auxiliaries to Constraint (3.16), ensuring that the predefined sequence of deliveries is respected. These constraints make it possible to track whether items of a pallet type have already been delivered to a customer in previous time slots, ensuring that the customer's delivery sequence is being followed. Constraint (3.17) checks whether any deliveries have been made in a time slot so that Constraint (3.18) can ensure that the makespan of the problem is as large as the last time slot in which a delivery takes place. In addition, Constraint (3.19) intends to identify whether a pallet has been transported by different AMRs in consecutive time slots, thereby verifying the decoupling between AMRs and pallets. Constraint (3.20) aims to determine which pallets are used in the distribution process. Finally, Constraint (3.21) ensures that the associated decision variables, specifically the integer variables, only assume non-negative integer values, i.e., equal to or greater than zero. In contrast, Constraint (3.22) establishes the binary nature of the decision variables.

Estimation of the N and S sets

The size of the N and S sets must be determined to ensure that both sets are large enough to guarantee efficiency in the delivery process. The methodology used to determine the appropriate size of each of these sets is detailed below.

N-set estimation

The size of the N-set is determined through the following steps:

1. Calculation of the total sum of the demand for each pallet type, considering all customers;
2. Division of the total demand for each pallet type by its capacity;
3. Determination of the maximum value obtained from the division.

As a result, this approach can be represented by the following expression:

$$N = \max \left\{ \frac{\sum_{c \in C} demand_{cp}}{capacity_p}, \forall \quad p \in P \right\} \tag{3.23}$$

As such, the highest value obtained defines the size of the N-set, ensuring that it is large enough to allow all pallet types to have enough pallets to satisfy customer demands.

S-set estimation

The size of the S-set is estimated using a heuristic that considers two scenarios: a best and a worst one.

On the one hand, in the best-case scenario, S_{min} represents the minimum number of time slots required, assuming that each pallet type is delivered optimally. In this case, the minimum number of time slots is equal to the number of different pallet types required (P):

$$S_{\text{min}} = P \tag{3.24}$$

On the other hand, in the worst-case scenario, S_{max} represents the maximum number of time slots, assuming that there is only one AMR available to carry out the entire distribution operation, which means that each customer order for each pallet type will need a separate time slot. The S_{max} is calculated by:

$$S_{\text{max}} = \sum_{c=1}^{C} \sum_{p=1}^{P} 1 \quad \Rightarrow \quad demand_{cp} > 0 \tag{3.25}$$

In this expression, the sum takes the value 1 for each time the demand of customer c for pallet type p is greater than 0.

Lastly, to calculate the S value, a weighting parameter, θ, is used to balance the worst-case and best-case scenarios:

$$S = \{(1 - \theta) \cdot S_{\text{min}} + \theta \cdot S_{\text{max}} \mid \theta \in [0, 1]\} \tag{3.26}$$

The value of θ can vary between 0 and 1, and it enables the number of time slots to be adjusted according to the complexity of the problem instance. In this way, it can be assumed that:

- $\theta = 0$ – Corresponds to the best scenario, resulting in $S = S_{\text{min}}$;
- $\theta = 1$ – Corresponds to the worst scenario, resulting in $S = S_{\text{max}}$.

Chapter 4

Results and Discussion

This chapter provides an overview of the test instances and configurations employed to obtain the results. Besides, it presents the decision support tool developed, which was used to generate the various instances and manage the necessary data. After that, a validation of the proposed mathematical model is also presented in order to show that the mathematical model is following the imposed conditions. Finally, the relevant results obtained are presented, and an analysis of the results of four different scenarios, namely the AMRs routes, is also made.

4.1 Test Instances and Configurations

The computational experiments were performed on a computer equipped with an Intel(R) Core(TM) i7-7700HQ processor at 2.80GHz, 16GB of RAM, and the Windows 10 Home operating system (64-bit, version 22H2). The mathematical model was solved using *IBM ILOG CPLEX Optimization Studio* 22.1.1. For each test instance, a time limit of 2700 seconds (45 minutes) was imposed as an optimisation parameter within *CPLEX*.

Since this project is based on a fictitious retail company problem, it was necessary to generate various datasets to simulate different operational scenarios. The test instances were constructed by varying the sizes of three main sets: number of customers (C), number of pallet types (P), and number of AMRs (A). The chosen values for these sets were selected to represent simple, moderate, and complex instances. The set combinations are as follows:

- Three different values of the number of customers – $C = \{4, 6, 8\}$;
- Three different values of the number of pallet types – $P = \{4, 6, 10\}$;
- Three different values of the number of AMRs – $A = \{2, 4, 6\}$.

This resulted in a total of twenty-seven combinations of the above sets. Additionally, each combination was tested with three repetitions, leading to a total of eighty-one instances. However, for the result presentation, only the average performance across the three repetitions for each combination will be discussed. The full set of combinations is presented in Table 4.1. It is important to emphasise that the three repetitions for each combination ensured that the model behaved consistently under different data.

In turn, for each instance, the values of the $demand_{cp}$, $capacity_p$, and $order_{cp}$ parameters were randomly generated by an instance generator. Consequently, the size of the N set (number of pallets of each type in stock) was determined dynamically based on the generated parameter values.

Table 4.1. Set of combinations

A	***P***	***C***
2	4	4
2	6	4
2	10	4
4	4	4
4	6	4
4	10	4
6	4	4
6	6	4
6	10	4
2	4	6
2	6	6
2	10	6
4	4	6
4	6	6
4	10	6
6	4	6
6	6	6
6	10	6
2	4	8
2	6	8
2	10	8
4	4	8
4	6	8
4	10	8
6	4	8
6	6	8
6	10	8

4.2 Decision Support Tool and Procedures

This section will describe the tool developed in *Microsoft Excel*, which acted as a decision support tool, and its integration process with *IBM ILOG CPLEX Optimization Studio*, used to solve the proposed mathematical model. This tool was used to generate the input data for the model, process the results obtained and facilitate their visualisation and interpretation.

4.2.1 General overview of the Decision Support Tool

Microsoft Excel played a key role, as, on the one hand, it is used for generating, organising, and managing the data needed to run the mathematical model. On the other hand, it presents the results in a clear and visually accessible way, providing an easier and more intuitive way of analysing and interpreting the results. As a result, an *Excel* file was created, which is structured according to Figure 4.1.

Figure 4.1. *Excel* file organisation.

The *Excel* file contains six worksheets: the "Data", "Demand", "Capacity", and "Order" sheets are where the necessary data is entered and created, while the "Results" and "Aux" sheets are used to visualise and analyse the results obtained after running the model.

It is important to mention that the entire automation process was implemented using Visual Basic for Applications (VBA) macros in order to make the whole process more efficient by reducing the time needed to create the data and organise the results obtained.

The "Data" sheet (Figure 4.2) contains the editable fields for entering data sets, such as the number of pallets (P), customers (C), AMRs (A), and the M value – a sufficiently large constant. Additionally, this sheet displays a menu with seven buttons: "Instances Generator", "N determination", "Consult Demands", "Consult Pallet Capacities", "Consult Delivery Order", "Obtain Results", and "Consult Results".

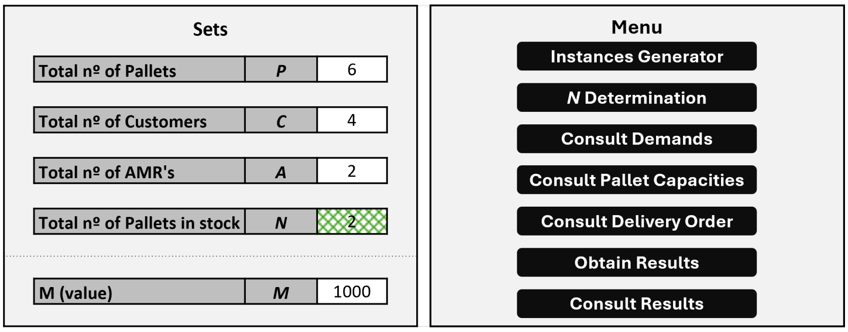


Figure 4.2. Layout of "Data" sheet.

With this in mind, the next steps should be followed:

1. To begin with, the values for "P", "C", "A", and "M" must be entered manually in the editable area. It is important to note that the field corresponding to the set N (highlighted in green in Figure 4.2) is not editable.

2. By pressing the "Instances Generator" button, random instances are automatically generated for the parameters $demand_{cp}$, $capacity_p$, and $order_{cp}$, based on the values entered for the P and C sets. Consequently, the "Demand", "Capacity", and "Order" sheets are automatically filled in.

3. Based on the values of the $demand_{cp}$ and $capacity_p$ parameters generated in the previous step, the "N determination" button calculates automatically the N set, according to the logic of Expression (3.23).

4. By clicking on the "Consult Demands", "Consult Pallet Capacities", and "Consult Delivery Order" buttons, the user is directed to the "Demand", "Capacity", or "Order" sheets, respectively. The layout of these sheets is illustrated in Figures 4.3, 4.4, and 4.5.

5. The "Obtain Results" button triggers the execution of the model in *IBM ILOG CPLEX Optimization Studio*, establishing an automatic link between the *Excel* file and *CPLEX*. Therefore, *CPLEX* reads the data from *Excel* and processes the optimisation model. After execution, the results are imported back into the *Excel* file.

6. Lastly, the "Consult Results" button is the last to be pressed, redirecting the user to the "Results" sheet, where it is possible to visualise the obtained results.

$Demand_{cp}$ | Demand from customer c for items of p-type pallet

	Pallet A	Pallet B	Pallet C	Pallet D	Pallet E	Pallet F
Customer 1	1	8	0	6	0	2
Customer 2	3	5	3	10	0	10
Customer 3	4	5	9	9	8	6
Customer 4	5	7	1	6	9	1

Figure 4.3. Layout of "Demand" worksheet.

$Capacity_{p}$ | Maximum item capacity of a p-type pallet

Pallet	Pallet A	Pallet B	Pallet C	Pallet D	Pallet E	Pallet F
Capacity	22	27	20	30	24	21

Figure 4.4. Layout of "Capacity" worksheet.

$Order_{cp}$ | Order of the items from pallet p in relation to the other items requested on the pallet of customer c

	Pallet A	Pallet B	Pallet C	Pallet D	Pallet E	Pallet F
Customer 1	4	1	0	2	0	3
Customer 2	5	2	3	4	0	1
Customer 3	2	6	5	4	3	1
Customer 4	2	4	3	1	5	6

Figure 4.5. Layout of "Order" table.

Instances Generation

The automatic generation of instances was carried out by developing a VBA code. Therefore, limits were imposed for the generation of values contained in the "Demand" table (Figure 4.3) and in the "Capacity" table (Figure 4.4). On the one hand, the "Demand" table includes values between 0 and 10. When demand takes the value 0, it means that a customer has not requested items of a given pallet type. On the other hand, the "Capacity" table contains random values between 20 and 30.

Meanwhile, the code developed to populate the "Order" table (Figure 4.5) also generates a random order in which the items of each pallet type will be delivered to each customer. This sequential order can range from 0 to the total number of pallet types available (P). Specifically, a random permutation of the numbers 0 to P is generated without repetitions, ensuring that each pallet type is assigned a unique delivery position for each customer. The value associated with the order indicates the delivery position; for example, a value of 1 means that pallet type should be the first to be delivered to the customer, while a value of 2 indicates that it will be the second in the sequence, and so on. If a customer does not request a pallet type (value 0 in the "Demand" table), the corresponding value in the "Order" table will also be 0, and this pallet will not be included in the customer's delivery sequence. Based on the table shown in Figure 4.5, it can be concluded that the delivery order for Customer 2 is 'F–B–C–D–A', as:

- Items from Pallet 'F' are the first in the delivery sequence, as they are associated with value 1;
- Items from Pallet 'B' are the second in the delivery sequence, as they are associated with value 2;
- Items from Pallet 'C' are the third in the delivery sequence, as they are assigned value 3;
- Items from Pallet 'D' are the fourth in the delivery sequence, as they are associated with value 4;
- Items from Pallet 'A' are the last in the delivery sequence, as they are assigned to value 5;
- Items from Pallet 'E' are associated with value 0, so they are not placed in the delivery order.

To make the display of the delivery sequence more intuitive, the "View Delivery Order" button was also created on the "Order" sheet, as shown in Figure 4.6. This button associates the delivery order values with the letters of the corresponding pallet types and organises these letters in the correct sequence.

View Delivery Order

	Delivery Order
Customer 1	B - D - F - A
Customer 2	F - B - C - D - A
Customer 3	F - A - E - D - C - B
Customer 4	D - A - C - B - E - F

Figure 4.6. Display of the delivery order.

Results Visualisation

Once the results have been obtained in *CPLEX*, they are exported to *Excel*.

With regard to the "Aux" sheet, this is an auxiliary sheet on which the results of the decision variables, x_{pnsac}, y_{pnsac}, and u_{pn}, obtained from the *CPLEX* are displayed in a raw way, without any processing.

As for the "Results" sheet, an interface as shown in Figure 4.7 can be visualised. It has four buttons: "Results", "Used Pallets", "Generate Display", and "Clear".

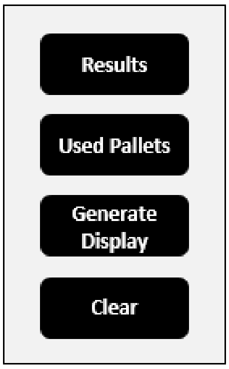


Figure 4.7. Interface for visualising results.

Thus, the first button fills in the table shown in Figure 4.8 with the values from the decision variables x_{pnsac} and y_{pnsac}, contained in the "Aux" sheet, showing which pallets have been delivered, in which time slot, which AMR delivered them, which customer they were delivered to, as well as the number of items delivered. According to the results of the u_{pn} variable, also displayed on the "Aux" sheet, the second button visually displays which pallets have been used, and according to their type, associates a different colour for each one, as can be seen in Figure 4.9. In turn, the third button provides a visual representation of the results presented on the table shown in Figure 4.8. In this way, according to Figure 4.10, it can be seen each robot's route, gradually showing the various time slots in which they made a delivery and showing which pallet was delivered (by painting the cell with the corresponding colour) and the customer to whom the delivery was made. The last button clears all the results from this sheet.

Pallet	Pallet ID	Time Slot	AMR	Customer	Delivered Qnt.
2	1	1	1	1	8
4	1	2	1	4	6
5	1	3	1	3	8
3	1	4	1	4	1
6	2	5	1	2	10
3	1	6	1	3	1
4	2	7	1	1	6
3	1	8	1	2	3
1	1	9	1	1	1
3	1	10	1	3	8
1	1	11	1	2	3
6	2	1	2	3	6
1	1	2	2	3	4
1	1	3	2	4	5
4	2	4	2	3	9
2	1	5	2	4	7
2	1	6	2	2	5
2	1	7	2	3	5
6	2	8	2	1	2
5	1	9	2	4	9
4	2	10	2	2	10
6	2	11	2	4	1

Figure 4.8. Results table.

Used Pallets	A1	B1	C1	D1	D2	E1	F2

Figure 4.9. Display of used pallets.

	Slot 1		Slot 2		Slot 3		Slot 4		Slot 5		Slot 6	
	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer
AMR 1	B1	1	D1	4	E1	3	C1	4	F2	2	C1	3
AMR 2	F2	3	A1	3	A1	4	D2	3	B1	4	B1	2

	Slot 7		Slot 8		Slot 9		Slot 10		Slot 11	
	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer
AMR 1	D2	1	C1	2	A1	1	C1	3	A1	2
AMR 2	B1	3	F2	1	E1	4	D2	2	F2	4

Figure 4.10. Display of AMR's routes.

4.2.2 Model implementation in *CPLEX*

IBM ILOG CPLEX Optimization Studio was the main tool used to solve the mathematical model presented in the Subsection 3.2.2. Therefore, it was implemented using the Optimization Programming Language (OPL) language.

The implementation was conducted using two main files: the *.mod* file and the *.dat* file. That said, on the one hand, the *.mod* file contains the model formulation, and, on the other hand, the *.dat* file collects the necessary data for the problem sets and parameters and exports the obtained results. As already mentioned, in order to simplify the processes of data input and result output, an integration between *CPLEX* and *Microsoft Excel* was established.

An important aspect of the *.mod* file is the implementation of the Objective Function using a lexicographic approach. For this purpose, it was necessary to use the *staticLex* function, and, in this regard, the objective function was defined in *CPLEX* as demonstrated in Figure 4.11.

```
//Objective function

dexpr int f1 = Cmax;
dexpr int f2 = sum(p in Pallets, n in PNumber) u[p][n];
dexpr int f3 = sum(p in Pallets, n in PNumber, s in 1..S-1) z[p][n][s];

minimize staticLex(f1, f2, f3);
```

Figure 4.11. *StaticLex* function applied in *CPLEX*.

For instances with a higher level of complexity, it was necessary to adjust *CPLEX's* optimisation parameters in the *.ops* file, namely setting a global time limit for the optimisation, in order to find a viable (but not necessarily optimal) solution by reducing the execution times of the model.

4.3 Validation

With the aim of guaranteeing the accuracy of the results and the robustness of the proposed mathematical model, a validation of the main conditions and constraints established during the formulation of the model was carried out. These validation elements consist of:

- Quantity of items delivered matches the quantity ordered;
- Correct allocation of AMRs;
- Compliance with the delivery order;
- Respect of pallet capacities;
- Respect of the number of pallets available for each pallet type;
- Respect of the continuous transportation of the same pallet by the same robot;
- Compliance of the number of time slots used with the determined makespan.

This stage is intended to confirm that all the mentioned conditions are being respected as defined in the mathematical model by analysing the results obtained for a validation scenario.

Therefore, Figure 4.12 shows the sizes of the different sets, P, C, A, and N, and the value considered for M.

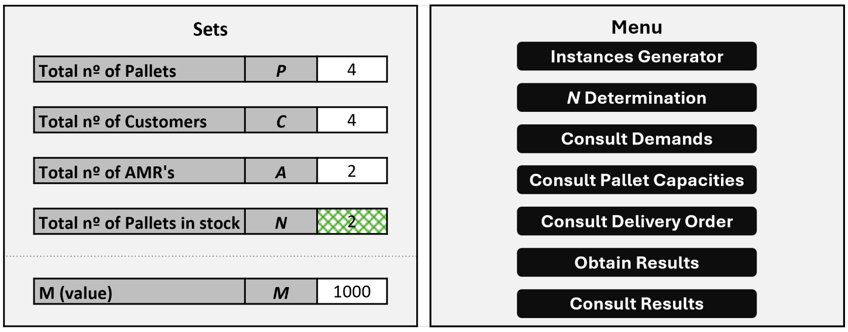


Figure 4.12. Sets of the validation scenario.

In turn, the generated data for the $demand_{cp}$ and $capacity_p$ parameters can be seen, respectively, in Table 4.2 and Table 4.3, and the delivery order corresponding to the $order_{cp}$ parameter is represented with an intuitive design in Table 4.4.

Table 4.2. Customer demands (validation scenario)

Demand$_{cp}$	**Pallet A**	**Pallet B**	**Pallet C**	**Pallet D**
Customer 1	4	8	0	3
Customer 2	8	3	3	3
Customer 3	10	0	2	3
Customer 4	1	8	1	3

Table 4.3. Pallet capacities (validation scenario)

Pallet	**Pallet A**	**Pallet B**	**Pallet C**	**Pallet D**
Capacity$_p$	22	21	22	23

Table 4.4. Delivery orders (validation scenario)

	Delivery Order
Customer 1	B - D - A
Customer 2	C - B - A - D
Customer 3	A - C - D
Customer 4	D - C - B - A

Based on the presented data, Table 4.5 shows the results obtained for the decision variable y_{pnsac}, indicating the pallet types used (p), the identification of the pallet used (n), the time slot in which the delivery took place (s), the AMR that made the delivery (a), the customer to whom the delivery was made (c), and finally the quantity delivered to each customer ("Delivered Qnt.").

Table 4.5. Variable y_{pnsac} results (validation scenario)

Pallet	Pallet ID	Time Slot	AMR	Customer	Delivered Qnt.
1	1	3	2	1	4
1	1	5	2	2	8
1	1	1	1	3	10
1	2	7	2	4	1
2	1	1	2	1	8
2	1	3	1	2	3
2	1	6	2	4	8
3	1	2	1	2	3
3	1	4	1	3	2
3	1	5	1	4	1
4	2	2	2	1	3
4	2	7	1	2	3
4	2	6	1	3	3
4	2	4	2	4	3

By comparing the results of the "Delivered Qnt." column with the values of the $demand_{cp}$ parameter represented in Table 4.2, it can be confirmed that all the required items were duly delivered, with no faults or overruns.

In Figure 4.13 the routes of each AMR can be observed, which were built based on the results of Table 4.5.

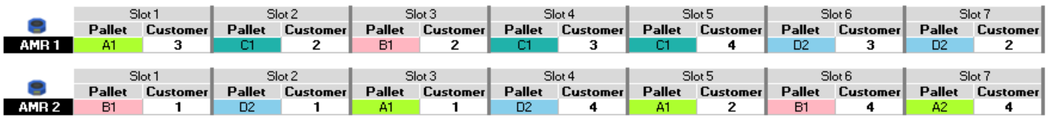


Figure 4.13. AMR routes (validation scenario).

Based on the visual representation of each AMR's journey, the following conclusions can be drawn:

- In each time slot, each pallet was transported by a single robot;
- Each AMR transported a maximum of one pallet at a time;
- In each time slot, each customer was served by only one robot.

Comparing each robot's pallet delivery order with the delivery order for each customer shown in Table 4.4, it was found that the delivery sequence was correctly followed for all customers. For instance, focusing on Customer 2's delivery order, it is known that the items from the different pallet types must be delivered in the following sequence: 'C–B–A–D'. With this in mind, by analysing the routes of the AMRs represented in Figure 4.13, it can be said that:

- In time slot 2, AMR 1 delivered the items from Pallet 'C';
- In time slot 3, AMR 1 delivered the items from Pallet 'B';
- In time slot 5, AMR 2 delivered the items from Pallet 'A';
- In time slot 7, AMR 1 delivered the items from Pallet 'D';

As such, it can be confirmed that the pallet delivery order was followed, concluding that the sequential logic established by the mathematical model is being respected.

By analysing the capacity of each pallet type and the demand of the various customers for each pallet type, it can be noticed that, for instance, to fulfil the demand of all customers, two pallets of type 'A' are needed. Therefore, the route of AMR 1 demonstrates that it first delivered items from pallet 'A1' to Customer 3, and then AMR 2 also delivered items of this pallet to Customer 1 and Customer 2. The demand from these three customers totalled 22 items, and since the capacity of this pallet is also 22 items, there were no more items left over. As Customer 4 requested 1 item of this pallet type, the robot had to pick up a new pallet of the same type, 'A2', to satisfy this customer's demand. This behaviour shows that the model is respecting the capacity constraints by picking a new pallet of the same type when the number of items left on a pallet is not enough to satisfy another customer's demand.

As can be verified in Figure 4.12, the value determined for the N-set was 2, indicating that there are a maximum of two pallets available in stock for each pallet type. In this sense, Figure 4.14 proves that no more than two pallets were used for each pallet type.

Used Pallets	A1	A2	B1	C1	D2

Figure 4.14. Sets of the validation scenario.

By observing the behaviour of, for example, AMR 1, which delivered items from pallet 'C1' or 'D2' to different customers in two consecutive time slots, it can be inferred that the decoupling condition is being met, as the robots continued to transport the same pallets over successive time slots.

Lastly, the number of time slots used to serve all the customers was 7. According to the makespan determined by the model (C_{max}), which was also 7, it can be said that the correspondence between the number of time slots used and the calculated makespan validates that the solution obtained is optimal, since the model met the constraint of minimising the total execution time without wasting time slots.

4.4 Results Discussion

This section presents and analyses the various results obtained from the application of the developed mathematical model.

The focus of this analysis is on the execution time, the values obtained for the three lexicographic objectives, and the performance of the solver. It is important to mention that because the objective function is lexicographic, a gap analysis cannot be carried out at this stage. Therefore, the results of the twenty-seven combinations were analysed based on the following criteria:

- **Execution Time** – Total time it took *CPLEX* to solve each instance;
- **Three objectives** – Values obtained for each objective of the lexicographic Objective Function;
- **Number of nodes and iterations** – Number of nodes explored and iterations to understand the computational complexity of solving each instance;
- $S_{\mathbf{min}}$**,** $S_{\mathbf{max}}$**, and** S **values** – Number of time slots to understand the influence it has on the model's performance.

4.4.1 Results Analysis

This subsection presents the results obtained for the aforementioned criteria.

As previously stated, three instances were created for each combination of sets (C, P, A). As such, in order to analyse the various criteria, an average was made of the results obtained for each combination of sets.

Execution Time

The first criteria evaluated was the execution time (T_{exec}). Table 4.6 shows the average execution times (in seconds) for each combination of parameters. While the first column indicates the number of AMRs (A) used, the following columns show the corresponding execution times for different combinations of number of customers (C) and pallet types (P). As such, each combination of C and P is evaluated in relation to A, allowing a comparative analysis of the impact of these variables on the execution time.

Based on the analysis of the average execution times shown in Table 4.6, in general, it can be concluded that for most instances, the execution time is lower than 12 minutes. However, for more complex instances where the number of customers and pallets is the maximum, ($P = 10$; $C = 8$), the execution time reached the limit imposed in the execution time optimisation parameter, i.e., 45 minutes.

Table 4.6. Average Execution Time (seconds)

A	C=4 P=4	C=4 P=6	C=4 P=10	C=6 P=4	C=6 P=6	C=6 P=10	C=8 P=4	C=8 P=6	C=8 P=10
2	8	15	69	15	31	511	80	554	2720
4	7	19	45	15	366	702	85	2710	2725
6	7	16	36	20	82	537	165	190	2727

Although increasing the number of AMRs offers small improvements in some instances, it can be seen that as the number of AMRs increases (from 2 to 4, or from 4 to 6), the execution time does not necessarily reduce. For example, for instance ($A = 2$; $P = 4$; $C = 4$), the execution time is 8 seconds, while with 4 and 6 AMRs the time is 7 seconds, showing only a slight difference.

On the other hand, increasing the number of pallet types has a significant impact on execution time. For example, comparing instance ($A = 6$; $P = 4$; $C = 6$) with instance ($A = 6$; $P = 10$; $C = 6$), it can be said that the execution time increased from 20 seconds to 537 seconds, respectively. This trend is consistently observed in other combinations of C and A, showing that increasing P makes the picking process more complex, which leads to a considerable increase in execution time.

The number of customers also significantly affects the execution time. For instance ($A = 2$; $P = 10$; $C = 4$) the execution time is 69 seconds, but with the increase of the number of customers to $C = 6$, the time also increased to 511 seconds. When C is increased to 8, the execution time reached a limit of 2720 seconds, indicating that the number of customers can be an important limiting factor in more complex instances. This pattern suggests that increasing the number of customers, in particular, increases the difficulty of optimising the system as it results in a greater number of interactions and route combinations between the AMRs and the pallets.

Table 4.7 shows the average execution time associated with the average number of pallets of each type needed (N) in each instance. Generally, the values obtained for the N-set were 2 and 3. With this in mind, it can be concluded that instances with $N = 3$ consistently showed longer execution times, especially when combined with high values of C and P. This suggests that as the demand for each pallet type increases and, consequently, the number of pallets of each type also increases, the optimisation process becomes more complex and the time needed to find the optimal solution is higher.

Table 4.7. Comparison between T_{exec} and N

$\boldsymbol{A}$	$\boldsymbol{P}$	$\boldsymbol{C}$	$\boldsymbol{N}$	T_{exec}
6	4	4	2	7
4	4	4	2	7
2	4	4	2	8
2	4	6	2	15
4	4	6	2	15
2	6	4	2	15
6	6	4	2	16
4	6	4	2	19
6	4	6	2	20
2	6	6	2	31
6	10	4	2	36
4	10	4	2	45
2	10	4	2	69
2	4	8	3	80
6	6	6	2	82
4	4	8	2	85
6	4	8	2	165
6	6	8	2	190
4	6	6	2	366
2	10	6	2	511
6	10	6	2	537
2	6	8	3	554
4	10	6	2	702
4	6	8	3	2710
2	10	8	3	2720
4	10	8	3	2725
6	10	8	3	2727

Three objectives

Another criteria to be evaluated was the values of the three objectives ($f_1(x)$; $f_2(x)$; $f_3(x)$) of the Lexicographic Objective Function (3.1). The results are given in Table 4.8.

- **Objective 1** – $f_1(x)$

  Firstly, looking at the results of the first objective, $f_1(x)$ (see Expression (3.2)), which represents the time needed to complete the delivery process (Makespan), it can be stated that, in several instances, increasing the **number of AMRs** **(**A**)** does not result in a drastic reduction of the makespan. For example, in instances with only four pallet types and four customers ($P = 4$ and $C = 4$), the makespan varies from 7 time slots (with $A = 2$) to 6 time slots (with $A = 4$

and $A = 6$). This variation of just one unit suggests that, for simpler scenarios, the model is already efficient at allocating AMRs, and the number of AMRs is not a determining factor for makespan. However, in more complex instances, such as scenarios with a higher number of customers and pallet types, like $P = 10$ and $C = 8$, the increase in the number of AMRs revealed to help reduce the makespan. Therefore, with only two robots, the makespan is 35 time slots, while with six robots (the triple), the makespan is reduced to 13 time slots.

With regard to the **number of pallet types (P)**, it can be said that it has a significant impact on makespan. Looking at instance ($A = 2$; $P = 4$; $C = 4$), the makespan turns out to be relatively low, with a value of 7 time slots. However, in instances where the number of pallets increases to 10, for example, instance ($A = 2$; $P = 10$; $C = 4$), the makespan also increased considerably, being 18 time slots. With this in mind, it can be concluded that the number of existing pallet types directly affects the efficiency of the process, as more pallets need to be distributed among the customers, turning the delivery process more complex and, consequently, increasing the number of time slots necessary to complete all the deliveries.

In turn, by analysing the relationship between the **number of customers (C)** and the makespan, it can be noted that as the number of customers increases, the value of $f_1(x)$ also increases. When comparing instance ($A = 2$; $P = 10$; $C = 4$) with instance ($A = 2$; $P = 10$; $C = 8$), it can be seen that in the former the makespan is 18, while in the latter the makespan is 35. This suggests that the demand generated by a greater number of customers makes the problem more complex and requires more time slots to meet all orders.

With this in mind, the first objective, $f_1(x)$, increases as the number of pallet types and customers increases, and adding more AMRs mostly has a significant impact on its reduction.

- **Objective 2** – $f_2(x)$

  Looking at the results of $f_2(x)$ (see Expression (3.3)), i.e., the number of pallets used, it can be concluded that an increase in the **number of different pallet types (P)** combined with an increase in the **number of customers (C)** implies a logical increase in the value of this objective.

  When the number of customers increases from four to eight, it is noticeable that there is a difference in the number of used pallets, indicating that the extra demand from the new customers impacts the number of pallets used. For example, in instance ($A = 2$; $P = 4$; $C = 4$) the number of pallets used is 5, while in instance ($A = 2$; $P = 4$; $C = 8$) a total of 15 pallets are used.

On the other hand, comparing the first instance, in which only four different types of pallets are involved, with instance ($A = 2$; $P = 10$; $C = 4$), it can be noticed that the number of pallets used increases from 5 to 12, revealing that increasing the number of different pallet types also increases the chance of more pallets being used in each scenario.

It is important to mention that the number of used pallets depends on the $demand_{cp}$ and $capacity_p$ parameters. As previously stated, these two parameters have a direct influence on the **number of each pallet type available in the stock (N)**, as if there is a large demand for a given pallet type, more pallets of the same type are needed and, consequently, more pallets, in general, are used. At the same time, demand increases as the number of existing customers increases. Looking at instances ($A = 2$; $P = 4$; $C = 8$) and ($A = 4$; $P = 4$; $C = 8$), it can be seen that the demand is higher in the first instance, so that there are three pallets in stock for each pallet type, while in the second instance there are only two pallets available for each pallet type. By comparing the results of $f_2(x)$ of these instances (same P and C, different N), it may be stated that the higher the value of N, the higher the number of pallets used.

Finally, it can also be said that the **number of AMRs (A)** available to make the deliveries does not have any impact on the number of used pallets.

- **Objective 3** – $f_3(x)$

  The last objective, $f_3(x)$ (see Expression (3.4)), is 0E+00 for almost every instance, which indicates that, in consecutive time slots, AMRs and pallets are mostly not unpaired. This behaviour reveals that in the majority of the scenarios, the mathematical model is efficient in maintaining a consistent allocation of AMRs to pallets, managing to optimise deliveries without causing interruptions in the continuity of these robots' operations.

  However, in some instances, such as ($A = 6$; $P = 4$; $C = 8$), ($A = 4$; $P = 6$; $C = 8$), ($A = 6$; $P = 6$; $C = 8$), and ($A = 6$; $P = 10$; $C = 8$), the value of $f_3(x)$ turns out to be different from 0E+00. These instances can be considered more complex than the others, as they have a large number of customers. In these scenarios, the model reaches the execution time limit imposed in the optimisation parameters, indicating that *CPLEX* does not have enough time to fully optimise the third objective, resulting in mismatches between AMRs and pallets. With this in mind, it can be concluded that when the execution time reaches the 45-minute limit, the model prioritises the optimisation of the first two objectives ($f_1(x)$ and $f_2(x)$), and the third objective ($f_3(x)$) is not fully optimised.

Table 4.8. Lexicographic Objective Function Results

A	***P***	***C***	***N***	$f_1(x)$	$f_2(x)$	$f_3(x)$
2	4	4	2	7	5	0E+00
4	4	4	2	6	5	0E+00
6	4	4	2	6	5	0E+00
2	6	4	2	11	8	0E+00
4	6	4	2	8	9	0E+00
6	6	4	2	7	7	0E+00
2	10	4	2	18	12	0E+00
4	10	4	2	11	12	0E+00
6	10	4	2	10	13	0E+00
2	4	6	2	11	7	0E+00
4	4	6	2	6	8	0E+00
6	4	6	2	7	7	0E+00
2	6	6	2	15	9	0E+00
4	6	6	2	9	11	0E+00
6	6	6	2	8	10	0E+00
2	10	6	2	27	17	0E+00
4	10	6	2	14	16	0E+00
6	10	6	2	11	15	0E+00
2	4	8	3	15	9	0E+00
4	4	8	2	9	7	0E+00
6	4	8	2	8	7	1140
2	6	8	3	23	13	0E+00
4	6	8	3	12	18	234
6	6	8	2	9	12	630
2	10	8	3	35	21	0E+00
4	10	8	3	19	26	0E+00
6	10	8	3	13	22	450

Number of nodes and iterations

The number of nodes explored and the number of iterations were also analysed to understand how increasing the complexity of the instances influences the performance of the optimisation model. Table 4.9 shows the results for both criteria evaluated, as well as the *CPLEX* execution time (T_{exec}) for each instance (already presented).

In simpler instances, with a reduced number of customers and pallet types, the model manages to find the optimal solution quickly, without exploring many nodes and with a reduced number of iterations. This indicates that the model is efficient for less complex scenarios, where there are fewer decisions to be made. For example, in the instance ($A = 2$; $P = 4$; $C = 4$), the number of nodes explored is 0 and the number of iterations is relatively low (2273 iterations), resulting in an execution time of 8 seconds, indicating that the model managed to find the optimal solution at the branch and bound tree quickly, without the need to explore more nodes.

As the number of customers and pallet types increases, the number of nodes and iterations also increases, indicating that the solver has to explore more combinations and decisions. However, the execution time is still relatively controlled, showing that the model can cope with this additional complexity.

For more complex instances, the number of nodes and iterations increases significantly, bringing the runtime to the imposed limit of 45 minutes. This shows that the model becomes less and less efficient as complexity increases, especially when the number of customers and pallet types is high. In instance ($A = 6$; $P = 10$; $C = 8$), the number of nodes explored is 5469 and the number of iterations was 11336311, with the execution time reaching the imposed limit of 2727 seconds. The fact that some instances reached the time limit indicates that the model explored many potential solutions without converging on an optimal solution in the time available. This is reflected by the high number of nodes examined and iterations, which continue to increase until the time limit is reached.

Table 4.9. Number of Nodes and Iterations

A	*P*	*C*	# Nodes	# Iterations	T_{exec}
2	4	4	0	2273	8
6	4	4	0	2382	7
4	4	4	4	3358	7
4	4	6	9	7681	15
4	6	4	0	7810	19
2	6	4	0	8126	15
6	6	4	9	8396	16
2	4	6	0	11066	15
6	10	4	0	12727	36
2	6	6	0	12817	31
2	10	4	88	49256	69
6	4	6	307	51787	20
4	10	4	657	76368	45
6	6	6	445	149465	82
6	6	8	303	231853	190
4	4	8	814	319000	85
2	10	6	3	347819	511
2	4	8	1841	541326	80
6	4	8	3929	657906	165
4	6	6	2619	928059	366
6	10	6	3508	941813	537
4	10	8	240	1124480	2725
2	6	8	1545	1208091	554
4	10	6	5845	1398768	702
2	10	8	765	1802278	2720
6	10	8	5469	11336311	2727
4	6	8	10510	11614466	2710

Number of time slots

To assess the direct influence of the number of time slots (S) on the complexity of the problem, an analysis was carried out. As mentioned in Subsection 3.2.2, its value can be determined using the heuristic defined by Expressions (3.24), (3.25), and (3.26). As such, this heuristic balances, by adjusting the weighting factor (θ), the minimum number of time slots, S_{min}, and the maximum number of time slots, S_{max}. The θ values assigned and the S, S_{min}, and S_{max} results obtained can be seen in Table 4.10.

- **Weighting factor**

  Based on the complexity of each instance, the value of the weighting factor was adjusted, ranging from 0.1 to 0.5. The assignment of θ followed the criterion of starting with lower values (close to 0), which give greater weight to the best scenario ($S_{\min}$), and gradually increasing this value until the optimum solution was obtained. Therefore, it was found that θ values of 0.1 and 0.2 were used in instances where the number of AMRs is four and six, while θ values of 0.3, 0.4, and 0.5 were used in scenarios where only two robots are involved, indicating that more time slots are needed to carry out the distribution process.

Table 4.10. Number of Time Slots

A	P	C	θ	$S_{\min}$	$S_{\max}$	S
6	4	4	0,1	4	16	6
4	4	6	0,1	4	22	7
4	4	4	0,2	4	15	7
2	4	4	0,3	4	14	7
6	6	4	0,1	6	23	8
4	6	4	0,1	6	24	8
6	4	6	0,2	4	22	8
6	6	6	0,1	6	34	9
6	4	8	0,2	4	29	10
4	4	8	0,2	4	30	10
6	6	8	0,1	6	43	10
4	6	6	0,1	6	34	10
2	6	4	0,3	6	22	11
2	4	6	0,4	4	21	12
6	10	4	0,1	10	36	13
4	10	4	0,1	10	37	13
4	10	6	0,1	10	54	15
4	6	8	0,2	6	48	15
6	10	6	0,1	10	54	15
2	4	8	0,4	4	29	15
2	6	6	0,4	6	30	16
6	10	8	0,1	10	73	17
2	10	4	0,3	10	36	18
4	10	8	0,2	10	75	23
2	6	8	0,5	6	45	25
2	10	6	0,4	10	53	28
2	10	8	0,5	10	70	40

- **Values of $S_{\min}$, $S_{\max}$, and S**

  With regard to the values obtained for $S_{\min}$, as it is equal to the number of existing pallet types, it can be said that it varied between 4 and 10 time slots, showing that as P increases, the value of $S_{\min}$ also increases.

  Meanwhile, $S_{\max}$ varies between 14 and 75 time slots. Analysing simpler instances, such as ($A = 2$; $P = 4$; $C = 4$), the maximum number of time slots obtained is 14, while in more complex instances, such as ($A = 4$; $P = 10$; $C = 8$), it reaches the highest value, 75. It is reasonable to say that S_{max} varies according to the number of customers and pallet types and, consequently, their respective demands, reflecting that there is a variation in the complexity of the problem when each customer places more or fewer orders.

The number of time slots is strongly influenced by the number of customers and pallet types. The greater the number of customers and pallets, the greater the S needed to complete deliveries, even with an increase in the number of AMRs. In many instances, S is closer to S_{min}, indicating that the system was able to operate efficiently. However, in scenarios with few AMRs and a large number of pallet types and customers, S tended to be closer to S_{max}, suggesting that the system needs more time slots to deal with the complexity of the problem.

4.4.2 Analysis of Gap Results

The previous analysis was based on the mathematical optimisation model that comprises a Lexicographic Objective Function (3.1) with three different objectives. Therefore, it was impossible to determine the optimal gap for each instance because the lexicographic function prioritises, in order of importance, the fulfilment of the three objectives and never assigns a gap. It can be said that the optimal gap represents the difference between the found solution and the best possible solution.

In order to be able to determine the optimal gap for each instance and, consequently, see the model's performance, it was necessary to simplify the objective function, turning it into a single-objective function. Hence, a second analysis was carried out, in which the new objective function only aims to minimise the total time to deliver all the orders (C_{max}). The other two objectives, which involved the number of pallets used and the unpairing of AMRs with pallets, were discarded. That said, the new objective function is represented as follows:

$$\text{Minimize} \quad C_{max} \tag{4.1}$$

By focusing exclusively on the objective of minimising the problem's makespan, the model only concentrates on the efficient allocation of resources in the shortest possible time.

It is important to emphasise that in order to analyse the gap under the same conditions as the original results, all the data for each instance was kept the same, including the value of θ, which influences the number of time slots S needed.

Table 4.11 displays the results obtained for the same instances in terms of execution time (T_{exec}), objective function solution, optimal gap, number of nodes explored, and iterations. The solution of the objective $f_1(x)$ is once again presented so that a direct comparison can be made with the new solution achieved for the objective function. The values of $S_{\min}$, $S_{\max}$, and S are not shown, as they remain the same as the original ones.

Table 4.11. Single-objective function results

A	P	C	T_{exec}	O.F	$f_1(x)$	Gap (%)	# Nodes	# Iterations
4	4	4	9	6	6	0,0	0	1161
6	4	4	9	6	6	0,0	0	585
4	4	6	9	6	6	0,0	0	2493
2	4	4	9	7	7	0,0	0	411
6	6	4	13	7	7	0,0	0	1532
6	4	6	16	7	7	0,0	198	53037
4	6	4	12	8	8	0,0	0	3416
6	6	6	27	8	8	0,0	488	100092
6	4	8	40	8	8	0,0	2900	766121
4	6	6	13	9	9	0,0	0	2943
6	6	8	25	9	9	0,0	0	22200
4	4	8	26	9	9	0,0	11	29765
6	10	4	20	10	10	0,0	0	3727
2	6	4	10	11	11	0,0	0	2406
4	10	4	22	11	11	0,0	0	9389
2	4	6	10	11	11	0,0	0	3237
6	10	6	35	11	11	0,0	0	15602
4	6	8	116	12	12	0,0	0	42081
6	10	8	233	13	13	0,0	0	82916
4	10	6	52	14	14	0,0	0	18169
2	6	6	19	15	15	0,0	0	7004
2	4	8	12	15	15	0,0	0	1321
2	10	4	22	18	18	0,0	0	7234
4	10	8	675	19	19	0,0	182	477770
2	6	8	119	23	23	0,0	0	37978
2	10	6	530	27	27	0,0	0	690
2	10	8	895	35	35	0,0	713	975068

In general, when analysing the results obtained from the mathematical model minimising only the makespan, it can be seen that the execution time was considerably reduced. The longest execution time observed was 895 seconds (approximately 15 minutes) for the instance ($A = 2$; $P = 10$; $C = 8$). This indicates that by removing the other two lexicographic objectives, the computational complexity of the problem decreased, allowing *CPLEX* to deal more efficiently with all instances, including those with higher parameters.

The gap obtained for all instances was 0%, which means that *CPLEX* found optimal solutions in all cases, even for instances previously considered more complex. When comparing the objective function values from this analysis with the values obtained for $f_1(x)$ from the lexicographic analysis, it can be concluded that the results are equal. This can be justified by the fact that in the lexicographic analysis, *CPLEX* prioritises the optimal resolution of the first objective before moving on to the subsequent objectives, meaning that the value obtained for $f_1(x)$ was also the optimal solution.

Looking at the number of nodes explored, it was noted that in the majority of instances, this value was 0, which indicates that *CPLEX* found the optimal solution immediately in the branch and bound tree, without the need to explore the solution space further. However, in some instances with $C = 8$, the number of nodes explored was higher, such as 2900 nodes, which suggests that although the model found the optimal solution, the complexity of some instances required more searches. The number of iterations was also significantly reduced compared to the previous analysis, reinforcing the reduction in computational complexity.

In summary, the 0% gap analysis for all instances shows that by minimising just one objective, the model finds optimal solutions quickly and efficiently, proving that the model is robust enough to deal with different data configurations. Although lexicographic analysis offers a more holistic view of the problem, simplifying it to a single objective function allowed for a more direct approach without compromising the quality of the results. This was not the case with the three objective function, as it turned out that the third objective ($f_3(x)$) introduced significant complexity into the problem, causing the execution time to increase dramatically and the optimal solutions not to be found within the time limit in several instances.

4.4.3 Analysis of Order Delivery Scenarios

In this section, the results of four scenarios with different combinations of the number of AMRs, pallet types, and customers are analysed. For each scenario, the AMR routes, the number of time slots used, and the efficiency of robot allocation are examined. It is important to emphasise that these results were obtained based on the mathematical model containing the lexicographic objective function. Thus, the layouts of the AMR delivery routes for each scenario are shown in Appendix A, B, C and D.

Scenario 1 – $A = 2$; $P = 10$; $C = 4$

In the first scenario, there are two AMRs, ten pallet types, and four customers. This scenario can be considered complex due to the fact that a large number of pallets need to be distributed using a small number of resources.

An illustration of the routes taken by both AMRs can be found in the figure of Appendix A.

Therefore, it can be pointed out that the number of time slots used was high, nineteen-time slots, as there were only two robots to distribute all the pallets required by the four customers. The use of a high number of time slots indicates that the system required a considerable amount of time to complete all the deliveries, as the combination of many pallet types and few robots significantly increased the makespan.

In addition, it can be seen that due to the high demand, the two AMRs were utilised to the maximum, and it can be said that the two robots alternated between customers efficiently throughout the slots. The only aspect to note is that AMR 2 did not make any deliveries in time slot 18, as this robot had to stand by, because the customer he needed to serve was Customer 3, who was already being served by AMR 1 in that same time slot. Therefore, the delivery of the items from the last pallet in his delivery order only took place in time slot 19.

The pallets used in this scenario are shown in Figure 4.15.

Figure 4.15. Used pallets in scenario 1.

Observing the behaviour of the two robots, it can be seen that, in general, there was no disassociation in consecutive time slots between the robots and the pallets they transported.

Scenario 2 – $A = 4$; $P = 4$; $C = 8$

The second scenario involves using four AMRs to distribute four pallet types among eight customers. The various routes of the different AMRs can be analysed in Appendix B.

In this scenario, the system used a total of nine-time slots to complete all the deliveries. This number of time slots proves to be relatively efficient, considering the number of customers and robots available. Most of the AMRs were utilised consistently, with the exception of a brief period of inactivity by AMR 3 in the last slot, when the remaining robots were completing their last deliveries and there were no more tasks to be assigned to this AMR.

Therefore, AMR 1 stood out for its high utilisation, with consecutive deliveries of type 'C' pallets, while AMR 3 also operated intensively, transporting type 'A' pallets in consecutive slots. In general, the utilisation of the AMRs was well distributed, which contributed to the efficiency of the process.

An important feature of this scenario was the absence of robot and pallet decoupling. For example, AMR 3 kept pallets 'A1' and 'A2' along multiple consecutive slots, and AMR 1 followed a similar pattern with pallet 'C1'. This behaviour is desirable, as it minimises the time wasted on unnecessary pallet changes between the robots and consequently maintains delivery efficiency.

In some cases, it was observed that the same AMR delivered pallets of the same type in alternate time slots. For example, in one time slot AMR 1 delivered pallet 'C2', in the next it delivered 'C1', and then returned to 'C2'. This behaviour shows that the system is efficiently calculating the number of items left on a pallet after serving a customer. If the remaining items of a pallet are not enough to serve the next customer, the robot picks up another of the same type ('C1') but, as soon as possible, returns to distribute the remaining items from the first pallet ('C2'), avoiding waste and maximising the use of pallets.

Figure 4.16 shows the pallets used in this scenario. This was one of the cases in which it was necessary to have three pallets in stock for almost every type, except for pallet type 'D', where only two were needed.

Figure 4.16. Used pallets in scenario 2.

Overall, this scenario demonstrated an efficient allocation of AMRs, with an adequate distribution of pallets over the time slots. The absence of decoupling of pallets and AMRs, and the efficient reuse of pallets reinforce the system's effectiveness in minimising waste and optimising deliveries.

Scenario 3 $- A = 6$; $P = 6$; $C = 4$

The third scenario was built in order to assess the behaviour of the model when there are more robots than the number of customers. That said, in this scenario, there are six AMRs, six types of pallets, and four customers. Based on the various results, it was possible to compile the routes of the six robots illustrated in the figure of Appendix C.

This scenario shows a completely different reality to the previous one, as it is clear that instead of the AMRs being overloaded, in this scenario there are some idle time slots for the different AMRs.

Overall, eight time slots were used, and it was observed that the utilisation of the AMRs was not homogeneous. While AMRs 1 and 2 were highly utilised, AMRs 3, 4, 5, and 6 were underutilised. In particular, AMRs 4 and 5 were inactive for several consecutive time slots, and AMR 4 was only used once, in time slot 6. This pattern of underutilisation suggests that the model may not be allocating the available resources efficiently since, despite there being 6 AMRs available, only two were used consistently throughout the process. This behaviour can be justified by the fact that the number of AMRs exceeded the number of customers with active orders, which inherently limits the possibility of simultaneous operation of all robots. Specifically, since only four customers were being served at a time, a maximum of four robots could be engaged in deliveries concurrently. In addition, the constraints imposed by the need to follow the predefined delivery order for each customer, alongside the requirement to avoid uncoupling robots and pallets limit the possibility of using all the AMRs in parallel. However, it is important to highlight that, in several time slots, even the four robots that could have been operating were not all active simultaneously. This indicates that the model did not optimise the delivery process efficiently, particularly when there was an excess of available resources relative to the number of customers being served. The observed inefficiencies demonstrate that, under certain conditions, the model's allocation strategy may not be fully effective in balancing the use of multiple AMRs, potentially leading to longer than necessary operation times.

Figure 4.17 shows the pallets used in this scenario. In this way, it can be seen that according to the demand of all customers, only one pallet of each type was needed, but for pallets of types 'D' and 'F', it was verified that two pallets of each type were needed.

Figure 4.17. Used pallets in scenario 3.

In addition, it is important to note that there was no decoupling of pallets and AMRs, which can be seen, for example, on the route of AMR 2, which delivered pallet A1 to several customers in consecutive time slots. This indicates that the system is ensuring that the robots keep the same pallet over several deliveries, which contributes to efficiency and avoids unnecessary pallet changes between robots.

Scenario 4 – $A = 6$; $P = 10$; $C = 8$

The fourth and final scenario analysed the most complex instance of the problem, involving six AMRs, ten pallet types, and eight customers. The routes taken by the various robots can be analysed in Figure D.1 presented in Appendix D.

Compared to the previous scenario, which also involved six AMRs, the number of customers in this case is higher, which results in fewer idle time slots for the AMRs. Only two moments of inactivity were recorded: in time slot 6 for AMR 6 and in time slot 13 for AMR 1 (at the end of the operation). Overall, the allocation of the AMRs to the various customers was done efficiently, with the AMRs operating almost continuously to cope with the high number of deliveries and customers. This demonstrates a more balanced allocation of resources and more efficient use of the available robots.

This scenario represents one of the instances where the execution time limit was reached, which resulted in the third objective of the lexicographic objective function not being fully optimised, i.e., the decoupling of AMRs and pallets was not avoided in all situations. This means that there were time slots when different AMRs transported the same pallet in consecutive time slots, which is a sub-optimal behaviour.

For example, in time slot 11 AMR 5 was transporting pallet ’E3’, but in the next time slot, time slot 12, AMR 3 started transporting the same pallet. The same happened between time slot 12 and time slot 13, where AMR 6 delivered pallet ’J3’ in time slot 12, but in the next time slot, AMR 2 transported that same pallet. This unpairing behaviour indicates that, despite the overall efficiency of the allocation, the model was unable to completely avoid mismatching AMRs and pallets, resulting in less efficient use at certain time slots.

Figure D.2 presented in Appendix D shows the pallets used in this scenario, with the number of pallets in stock of each pallet type varying between two and three units.

In short, the overall efficiency of the operation was very high, with few moments of robot downtime and a robust allocation of resources, showing that the system can deal with complex scenarios.

Chapter 5

Conclusions

This chapter presents the conclusions, highlighting the main aspects of the project, mentions the limitations faced during the implementation of the mathematical model, and provides some suggestions for future work.

5.1 Final Conclusions

The developed project focused on the optimisation of the order picking process assisted by Autonomous Mobile Robots in the context of a picker-to-parts system. The focal point of the executed work consisted of the attainment of a MIP model with a discrete time approach capable of optimising the routes of the AMRs, focusing on enhancing the efficiency of the picking and delivery process without violating the predefined delivery sequence for each customer. It is important to emphasise that the developed mathematical model integrated a lexicographic objective function with three main objectives: minimising makespan (total operating time), minimising the number of pallets used of each type, and minimising the unpairing between AMRs and pallets in consecutive time slots.

The mathematical model was implemented in *IBM ILOG CPLEX Optimization Studio* and went through several iterations until the model was sufficiently refined to accurately represent the problem. Several test instances were created, varying namely the number of AMRs, customers, and pallet types. Thus, the model's performance was then evaluated and validated on the basis of the results obtained, with the support of a tool developed in *Microsoft Excel*.

Based on the results obtained, it can be concluded that the model was effective in defining defining optimal routes for the AMRs, minimising the total operating time in the tested instances in the most efficient way. In most scenarios, the model was able to solve the problem efficiently, with execution times below the imposed limit of 2700 seconds, especially in less complex instances, with fewer customers and pallet types. However, as the complexity of the instances increased, the execution time also grew significantly, particularly for scenarios with eight customers and ten different pallet types, where the execution time reached the imposed limit, indicating that the increase in the number of pallets (P) and customers (C) was the main factor responsible for the increase in computational complexity.

The lexicographic objective function proved effective in optimising different aspects of the picking process. The first objective (minimising makespan) was achieved consistently, especially in more complex instances, where the introduction of more AMRs helped to reduce the total operating time. The second objective (minimising the number of pallets) was also effective, ensuring that the number of pallets used was strictly necessary. The third objective (minimising the unpairing between AMRs and pallets) showed variable results. In most instances, unpairing was minimised, but in more complex scenarios, especially when the time limit was reached, unpairing occurred in consecutive slots, indicating that the third objective was not fully optimised in these cases. Although the increase in the number of AMRs (A) contributed to reducing makespan in more complex instances, it was found that in simpler instances this increase had no significant impact. This shows that in scenarios with fewer customers and pallets, the number of AMRs was already sufficient.

In addition, a second analysis was carried out, where the lexicographic objective function was transformed into a single objective function, focused only on minimising makespan. In this analysis, the optimality gap was always 0%, indicating that the model reached the optimal solution in all cases. This shows that even in more complex scenarios, *CPLEX* was able to find the optimal solution for makespan, ensuring that the total time to complete deliveries was minimised. Simplifying the objective function proved to facilitate optimisation in more complex problems, with the third objective being the main driver affecting the model's performance when the full lexicographic function is used.

Under the scenarios analysed, it was observed that the allocation of the AMRs was efficient in most cases. The model managed to allocate the AMRs in order to minimise idle times, respecting all the constraints imposed, such as the delivery sequence, the mismatch between AMRs and pallets, and the efficient use of pallets. However, in scenarios where the number of robots was higher than the number of customers, it was found that some AMRs were inactive for several time slots. This was expected due to the fact that, in such cases, the number of AMRs is greater than the number of customers to be served, and as these cannot be served by more than one robot in each time slot, it resulted in some robots being idle.

In terms of validation, all the operational constraints imposed by the model were respected. Pallet capacities and customer demands were met in all instances, and the predefined delivery sequence for each customer was strictly followed. This demonstrates the robustness and accuracy of the mathematical model.

In conclusion, the development of the Decision Support System proved to be efficient in optimising the order picking process assisted by AMRs, proving to be a viable solution for increasing efficiency in automated warehouses. However, as the complexity of the instances increased, some limitations of the model were revealed, especially with regard to the third objective of the lexicographic function. Nevertheless, the model proved to be robust and effective in most of the scenarios tested, offering a solid basis for future optimisations in the field of automated picking systems.

5.2 Limitations and Future Research

This section presents the challenges faced during the implementation of the mathematical model. In addition, considerations that could be explored in future work will also be suggested.

After implementing the mathematical model developed and analysing the results obtained, some limitations were identified.

One of the main limitations is computational capacity and execution time. As the size of the data sets increased, the time needed to solve the problem also grew significantly. While for smaller instances the model found optimal solutions in a few seconds, in more complex instances, with a high number of customers and pallet types, the execution time reached the 45-minute limit. For these cases, the model was unable to reach the optimal solution within the stipulated time, which compromises the practical applicability of the model in more dynamic, real-time scenarios.

Another significant limitation is related to the number of pallets available in stock of each type (N), which varied between 1 and 3. This number was limited in order to maintain reasonable execution times. However, it turned out that if the capacity and demand intervals were changed, this value would increase considerably and the execution time would grow exponentially, in some cases preventing viable solutions from being obtained within an acceptable time frame.

Some suggestions that could be explored in future research are:

- **Pallet heterogeneity**

  In the considerations for developing the mathematical model, pallets were treated homogeneously, which simplifies the management and allocation of AMRs. However, in more realistic scenarios, pallets with different types of items may be involved. A future proposal is to introduce heterogeneous pallets into the model, where each item could be associated with a different SKU.

- **Impact of the Geographical Distribution of Customers**

  Another suggestion would be to incorporate the geographical information about the location of customers. Currently, the model does not take into account the physical distribution of customers, treating them as fixed locations within a dynamic picking area. Incorporating distances between delivery points would make it possible to evaluate the efficiency of robot routes in a more dynamic and realistic context.

- **Development of Heuristic or Metaheuristic Algorithms**

  The developed mathematical model proved to be effective for small and medium-sized instances, but for larger instances, the execution time became a challenge. Therefore, the development of heuristic or metaheuristic algorithms capable of offering good, although not necessarily optimal, solutions for large-scale instances would be something to incorporate.

Appendix A

Display of the order delivery process carried out by the AMRs in Scenario 1

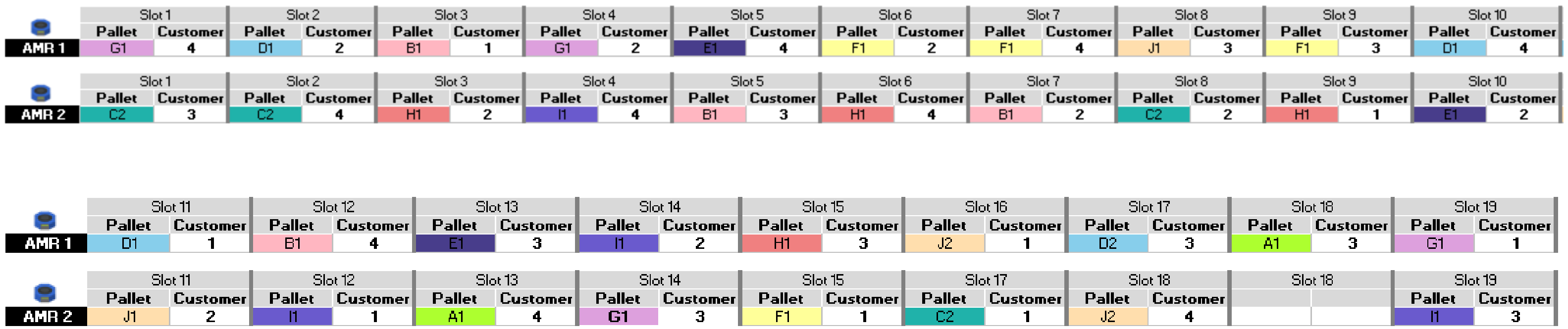


Figure A.1. AMR routes in scenario 1.

Appendix B

Display of the order delivery process carried out by the AMRs in Scenario 2

	Slot 1		Slot 2		Slot 3		Slot 4		Slot 5		Slot 6		Slot 7		Slot 8		Slot 9	
	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer
AMR 1	D3	3	C1	2	D3	2	C2	4	C1	7	C1	6	C2	3	B2	1	C1	3
AMR 2	B1	6	A3	6	B2	8	B3	5	D3	8	B1	7	A3	2	D3	7	B1	2
AMR 3	C2	8	A2	7	A2	8	A1	4	A1	5	B3	4	C3	5	A2	6		
AMR 4	A1	1	B3	3	C3	1	D1	1	B2	6	D1	4	D1	6	A1	3	D1	5

Figure B.1. AMR routes in scenario 2.

Appendix C

Display of the order delivery process carried out by the AMRs in Scenario 3

	Slot 1		Slot 2		Slot 3		Slot 4		Slot 5		Slot 6		Slot 7		Slot 8	
	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer
AMR 1	E2	2			C1	4	C1	2	D1	2	B1	1			B1	3
AMR 2	F2	3	A1	4	A1	1	A1	3			A1	4			D2	1
AMR 3	C1	1					E2	1	E2	3			C1	3		
AMR 4											D2	3				
AMR 5	D2	4											E2	4	A1	2
AMR 6					B1	2	B1	4			F1	2	F2	1	F2	4

Figure C.1. AMR routes in scenario 3.

Appendix D

Display of the order delivery process carried out by the AMRs in Scenario 4

	Slot 1		Slot 2		Slot 3		Slot 4		Slot 5		Slot 6		Slot 7		Slot 8		Slot 9		Slot 10	
	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer	Pallet	Customer
AMR 1	F1	7	G1	6	I3	4	F3	8	A3	8	A1	1	G1	5	E2	2	I3	1	C2	7
AMR 2	H3	8	I2	5	G3	3	C3	5	B3	2	H3	4	F3	1	B3	3	E1	8	H1	5
AMR 3	D2	5	D2	2	J1	1	E1	7	J1	3	J3	7	D3	7	G2	8	A1	5	I3	3
AMR 4	B3	1	E2	3	F3	7	B1	6	C3	6	D3	8	C3	4	J1	5	J1	2	A1	6
AMR 5	C3	3	C3	1	C2	2	D2	4	G3	4	F1	5	A1	2	D2	1	D2	6	E2	1
AMR 6	I2	6	B1	7	D2	8	H1	1	H1	7			I2	8	A3	4	F1	4	B3	4

AMR	Slot 11 Pallet	Slot 11 Customer	Slot 12 Pallet	Slot 12 Customer	Slot 13 Pallet	Slot 13 Customer
AMR 1	G1	7	I3	7		
AMR 2	C2	8	B1	8	J3	8
AMR 3	J3	6	E3	5	E3	6
AMR 4	A3	3	F1	3	B3	5
AMR 5	E3	4	G3	1	G2	2
AMR 6	D3	1	J3	4	H3	3

Figure D.1. AMR routes in scenario 4.

Used Pallets	A1	A3	B1	B3	C2	C3	D2	D3	E1	E2	E3	F1	F3	G1	G2	G3	H1	H3	I2	I3	J1	J3

Figure D.2. Used pallets in scenario 4.

Bibliography

Ali, S., Ramos, A. G., Carravilla, M. A., and Oliveira, J. F. (2022). On-line three-dimensional packing problems: A review of off-line and on-line solution approaches. *Computers and Industrial Engineering*, 168:108122.

Balakirsky, S., Christensen, H. I., Kramer, T., Kolhe, P., and Proctor, F. (2010). Mixed Pallet Stacking: An Overview and Summary of the 2010 PerMIS Special Session. In *Proceedings of the 10th Performance Metrics for Intelligent Systems Workshop*, pages 62–68, New York, NY, USA. ACM.

Balogh, J., Békési, J., Galambos, G., and Reinelt, G. (2010). On a Multidimensional Semi-on-line Bin Packing Problem. *International Conference on Applied Informatics*, 2(21):191–197.

Bartholdi, J. and Hankman, S. (2019). Warehouse and Distribution Science Release 0.98.1. *The Supply Chain and Logistics Institute*, pages 1–315.

Berndt, S., Jansen, K., and Klein, K. M. (2020). Fully dynamic bin packing revisited. *Mathematical Programming*, 179(1-2):109–155.

Blum, C. and Roli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308.

Bortfeldt, A. and Wäscher, G. (2013). Constraints in container loading – A state-of-the-art review. *European Journal of Operational Research*, 229(1):1–20.

Boysen, N., Briskorn, D., and Emde, S. (2017). Parts-to-picker based order processing in a rack-moving mobile robots environment. *European Journal of Operational Research*, 262(2):550–562.

Coutinho, C. P. (2014). *Metodologia de Investigação em Ciências Sociais e Humanas: Teoria e Prática*. Leya.

de Carvalho, P. R. V. and Elhedhli, S. (2022). A data-driven approach for mixed-case palletization with support. *Optimization and Engineering*, 23(3):1587–1610.

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

Elhedhli, S., Gzara, F., and Yildiz, B. (2019). Three-Dimensional Bin Packing and Mixed-Case Palletization. *INFORMS Journal on Optimization*, 1(4):323–352.

Epstein, L. and Kleiman, E. (2009). Resource augmented semi-online bounded space bin packing. *Discrete Applied Mathematics*, 157(13):2785–2798.

Faina, L. (2000). A global optimization algorithm for the three-dimensional packing problem. *European Journal of Operational Research*, 126(2):340–354.

Fragapane, G., de Koster, R., Sgarbossa, F., and Strandhagen, J. O. (2021). Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *European Journal of Operational Research*, 294(2):405–426.

Gambosi, G., Postiglione, A., and Talamo, M. (2000). Algorithms for the relaxed online bin-packing model. *Society for Industrial and Applied Mathematics*, 30(5):1532–1551.

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

Ha, C. T., Nguyen, T., Bui, L., and Wang, R. (2017). An Online Packing Heuristic for the Three-Dimensional Container Loading Problem in Dynamic Environments and the Physical Internet. *European Conference on the Applications of Evolutionary Computation*, pages 140–155.

Hernández, C., Bermejo-Alonso, J., and Sanz, R. (2018). A self-adaptation framework based on functional knowledge for augmented autonomy in robots. *Integrated Computer-Aided Engineering*, 25:157–172.

Hernandez, K., Bacca, B., and Posso, B. (2017). Multi-goal path planning autonomous system for picking up and delivery tasks in mobile robotics. *IEEE Latin America Transactions*, 15(2):232–238.

Le-Anh, T. and De Koster, M. B. M. (2006). A review of design and control of automated guided vehicle systems. *European Journal of Operational Research*, 171(1):1–23.

Li, K. and Cheng, K. H. (1990). On three-dimensional packing. *SIAM Journal on Computing*, 19(5):847–867.

Li, K. and Cheng, K. H. (1992). Heuristic algorithms for on-line packing in three dimensions. *Journal of Algorithms*, 13(4):589–605.

Löffler, M., Boysen, N., and Schneider, M. (2023). Human-robot cooperation: Coordinating autonomous mobile robots and human order pickers. *Transportation Science*, 57.

Miyazawa, E. K. and Wakabayashi, Y. (1997). An Algorithm for the Three-Dimensional Packing Problem with Asymptotic Performance Analysis I. *Algorithmica*, 18(1):122–144.

Montoya-Torres, J. R., López Franco, J., Nieto Isaza, S., Felizzola Jiménez, H., and Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115–129.

Papadimitriou, C. H. and Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*. Courier Corporation, New York.

Pardalos, P. M., Du, D. Z., and Graham, R. L. (2013). *Handbook of Combinatorial Optimization*. New York: Springer.

Puchinger, J. (2005). *Combining Metaheuristics and Integer Programming for Solving Cutting and Packing Problems*. PhD thesis, Vienna University of Technology.

Pugliese, G., Chou, X., Loske, D., Klumpp, M., and Montemanni, R. (2022). AMR-Assisted Order Picking: Models for Picker-to-Parts Systems in a Two-Blocks Warehouse. *Algorithms*, 15(11).

Srinivas, S. and Yu, S. (2022). Collaborative order picking with multiple pickers and robots: Integrated approach for order batching, sequencing and picker-robot routing. *International Journal of Production Economics*, 254:108634.

Traqueia, A., Euzébio, C., Soares, D., Pacheco, E., Taveira, E., Bernardo, I., Rios, J., Sousa, L., Bento Lopes, M., and Soares, T. (2021). *Reflexões em torno de Metodologias de Investigação: métodos (Vol.1)*. Universidade de Aveiro.

Tsai, R. D., Malstrom, E. M., and Kuo, W. (1993). Three Dimensional Palletization of Mixed Box Sizes. *IIE Transactions*, 25(4):64–75.

Ulungu, E. L. and Teghem, J. (1994). Multi-objective combinatorial optimization problems: A survey. *Journal of Multi-Criteria Decision Analysis*, 3(2):83–104.

Vijayakumar, V. and Sgarbossa, F. (2021). A literature review on the level of automation in picker-to-parts order picking system: research opportunities. *IFAC-PapersOnLine*, 54(1):438–443.

Vijayakumar, V. and Sobhani, A. (2023). Performance optimisation of pick and transport robot in a picker to parts order picking system: a human-centric approach. *International Journal of Production Research*, 61(22):7791–7808.

Wang, F. and Hauser, K. (2021). Robot Packing with Known Items and Non-deterministic Arrival Order. *IEEE Transactions on Automation Science and Engineering*, 18(4):1901–1915.

Williams, C. (2007). Research Methods. *Journal of Business & Economic Research*, 5(3).

Yoon, C. S. and Sharp, G. P. (1996). A structured procedure for analysis and design of order pick systems. *IIE Transactions*, 28(5):379–389.

Zhao, X., Bennell, J. A., Bektaş, T., and Dowsland, K. (2016). A comparative review of 3D container loading algorithms. *International Transactions in Operational Research*, 23(1-2):287–320.

Zhao, Z., Cheng, J., Liang, J., Liu, S., Zhou, M., and Al-Turki, Y. (2024). Order picking optimization in smart warehouses with human–robot collaboration. *IEEE Internet of Things Journal*, 11(9):16314–16324.

Žulj, I., Salewski, H., Goeke, D., and Schneider, M. (2021). Order batching and batch sequencing in an AMR-assisted picker-to-parts system. *European Journal of Operational Research*, 298(1):182–201.