UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Tecnica e Gestione dei Sistemi Industriali

Corso di Laurea Magistrale in Ingegneria Gestionale

Robotic Mobile Fulfillment Systems

A simulative study with three different managements

Master Thesis

Candidate

Daniele Galliussi

Supervisor Prof. Daria Battini

Co-supervisor Prof. Allan Larsen

Academic Year 2021 - 2022

Robotic Mobile Fulfillment Systems

A simulative study with three different managements

Daniele Galliussi

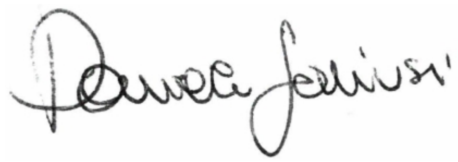
Master Thesis, October 2022.

Approval

This Master Thesis has been prepared over six months in collaboration with Management Department, at the Technical University of Denmark, DTU, during a Erasmus+ experience.

It is assumed that the reader has a basic knowledge in the areas of logistics and management.

Daniele Galliussi - student n°2027546



Date

17th October 2022

Abstract

In the last years, especially with the corona-virus pandemic, the e-commerce had increased such as never before. Analyzing the e-commerce orders, we can assert that they are generally small in batch and with strong demand's variability, while the e-commerce warehouses are large in quantity and the customer wish a very short delivery time and low prices. For all this reason, order fulfillment can be quiet challenging for warehouses and for the logistics we are used to.
This paper has such as primary objective to analyze the RMFSs with different managements where it changes the warehouse store policy. In particular, it analyze two different layouts with two different kind of pod: multi product pods, where it is stored a various type of items, without special assignment, mono-product pods, where it is possible to store only one kind of items, and with these pods we analyze two different layout: a random layout where the pods are stocking in a random position and an class-based policy location where the pods have a precise place decide in according with his class contents.
After a brief introduction part about warehouse system, with a special focus on picking operations, it describes the RMFSs, the general simulation methodology we study the three methodology described previously for understanding the performance and the best solutions.

Acknowledgements

Finally, also for me, this experience finished and it represent a useful moment to thinking and for the acknowledgments.
First of all, I would like to thank my supervisor professor Daria Battini for the thrust she placed in me in entrusting me with this very challenging thesis project that was able to stimulate me, and which, despite in initial difficulties, taught me to seek new solutions to the problems I was encountering.
I thank, in turn, professor Allan Larsen for welcoming me to DTU, for his helpfulness, for his advice that he skillfully transferred to me during this experience and and his support and supervision during these month.
Moreover, I would like to thank my parents for their support during all these years: the experiences that I was able to do in this period I made especially thank to you. Thanks mum! Thanks dad! Thanks to my sister who has always been able to listen to me and carry me around the world. Keep up the good work! And thanks to my family for their support.
I thank also Davide and Riccardo, my foster siblings, with whom I shared a lot: not only parties and funny stuff, but also for the best debriefing. Thanks for your support during the last five years!
Thanks to Luca, for your advises and for try to improve myself, always.
Thanks to Portone14 with his ”feeling at home” that allow me to be here to write this thesis and thanks to Vicenza’s people: with you I spent the best moment during my university’s study!
And, at the end, I would like to thank Andrea, Edoardo, Josh, the ”danish” buddy group and other girls and guys which I pass a lot of time in the Lyngby ”bar” in this amazing Erasmus experience.

Thanks to all,

Daniele

Acronyms

AGV Automated Guided Vehicles

AMRs Autonomous mobile robots

AS/RS Automated storage and retrieval systems

CPS Cyber physical system

IoT Internet of Things

OA Order Assignment

OPSs Order Pickins Systems

POA Pick Order Assignment

PPS Pick Pod Selection

PP Path Planning

PSA Pod Selection Assignment

PS Pod Selection

RFID Radio Frequency Identification

RMFS Robot Mobile Fulfillment System

ROA Replenishment Order Assignment

RPS Replenishment Pod Selection

SKU Stock Keeping Unit

TA Task Allocation

TC Task Creation

VLM Vertical lift modules

WMS Warehouse Management System

Se vuoi cambiare vita, sii creativo.
A volte la strada non va trovata, va inventata.

- Gianluca Gotto

List of Figures

List of Tables

Contents

Introduction

In recent years, consumers' buying behavior has changes, more people prefer to buy and sell in digital stores from the comfort of their sofa or in the comfort of their houses.
The last data of eCommerce, for Italy only for the 2022, published by NetComm [1] show that the online purchases amount to thirty-four billions of euro, whit an increase of 10% over the 2021, with a little decrease with respect to the forecast [2].
This growth confirms the trend in eCommerce, bringing the incidence of online sales around 11% of total sales, thanks to food and other typical gourmet products, design furniture, home living and beauty such as reported in Table 1.

	Apparel	**Food and grocery**	**Home living**	**Beauty**	**Total Online**
Growth	+10%	+17%	+14%	+8%	+10%
Amount	5,6 billion	4,8 billions	3,9 billions	1,2 billions	34 billions

Table 1: Amount and Growth in eCommerce 2022 respect 2021 [3]

.

The main trend in eCommerce is increasingly related with the optimization of processes, supply chain and logistic management, and all that, not only to create a better experience for the customers, but especially to keep costs down: energy costs, inflation, geopolitical instability and supply chain crisis as the last months require.
The COVID-19 pandemic has increased the variety and the frequency of customers purchase while the companies augmented their investment on digital technology looking forward to the so-called industry 4.0. Browsing the Internet you can find various data on e-commerce, such as that 38% of online shoppers will abandon their order if delivery takes more than a week [4].
From a logistics point of view, all of these information mean great challenges: enormous increase of orders and stored materials, smaller handling batches with almost the same quantity of products sold and short order management time and short delivery time.

For all the previous reasons is it easy to understand why there is the need to explore the new challenges for the warehouse management, finding new strategies for material handling operations and storing policies, looking for optimization of the process.

To solve the inefficiency of traditional picking processes and to improve the performance of the warehouse, in the first years of twentieth century, RMFS was born.The robotic mobile fulfillment systems are not well investigate in the literature, yet. For this reason, this work aimes to analyze their performance, within three different managements. In Chapter 1 we analyze the warehouse in general, with a special focus on picking process and smart warehouses.
In Chapter 2, we analyze the robotic mobile fulfillment systems, his history and functionality and some information about the study about the RMFS performance.
Going on into the paper, it is possible to find a brief introduction about the simulation model, in Chapter 3, and, in the second part of this work we start to talk about the simulation model (Chapter 4), his verification and validation (Chapter 5), his results in Chapter 6 and, in the Chapter 7 what may be the future work related.

1 Warehouse

Warehouses are an important, and in the same time also critical, player in supply chains. According to Kearney et al. (2004), warehousing contributed to about 20% of the surveyed companies' logistics costs in 2003 [5]. Warehouse assumes different meanings: they are commonly used for storing and buffering products at and between point of origin and point of consumption, but are used also as distribution centers that indicates transhipment, cross-dock or platform centre. We can summarize their main purposes in:

1. Storing raw materials, semi-finished products and exceeded materials;
2. Reduce demand variability through the use of stock;
3. Increase the customer satisfaction with a lower transportation leading time;

These aspects empathize the competitive advantages for companies and it allows to understand the importance of optimization inside the process in the supply chains.
For a better understanding of the topic, in section 1.1 we will briefly analyse the warehouse management and the operations done inside, with a special focus on picking process, while in section 1.3 we shortly introduce the main aspects of smart warehouses that, more and more, the companies choose from an industry 4.0 perspective.

1.1 Warehouse management

Warehouse management is the art of control and coordination of the day-by-day activities within a warehouse to ensure the efficiency and effectiveness for all operations inside a warehouse [6].
With warehouse management, usually, several different processes are considered. It is possible to classify them as:

1. **Warehouse layout design and optimization**: is an essential strategic decision related with storage space for goods, for working space. The main areas inside a warehouse are for receiving stock, staging, storing, picking, packing, shipping and for the offices.

2. **Goods acceptance and receipt**: all the activities from delivery notification to storage process, such as good acceptance, good receipt and good inspection for quality checks;

3. **Storage**: this kind of task can be done handily (manual storage systems) or using automatic systems and comprise the identification of the different articles and decision about their location;

4. **Picking and packing**: in this part of the processes, it is possible to include all the decision, about the fulfillment policies and the material flow to get the correct product at the end of the operation;

5. **Shipment**: here we found all the activities of packing and delivery of the goods that are leaving the warehouse.

6. **Optimizing internal processes and improving overall warehouse performance**: these activities may concern the management stock, the policies of warehouse management and the constant control of KPIs.

In all part of these it necessary to improve the overall quality of service, productivity and efficiency while minimizing costs and failures.

1.1.1 Optimization in warehouse

Optimization is the process of making a system more effective by adjusting the different variables inside it for the achievement of economical, technical and organizational goals.

It is important also to consider others problems that occurs during the different processes, especially, taking into consideration that an error during the picking process, increases the total cost with new activities such as:

- Accepting returns;
- Labour cost due to manual handling and checking of the returned item;
- Picking the replacement item;
- Re-packing;
- Re-delivery;

- Administration costs.

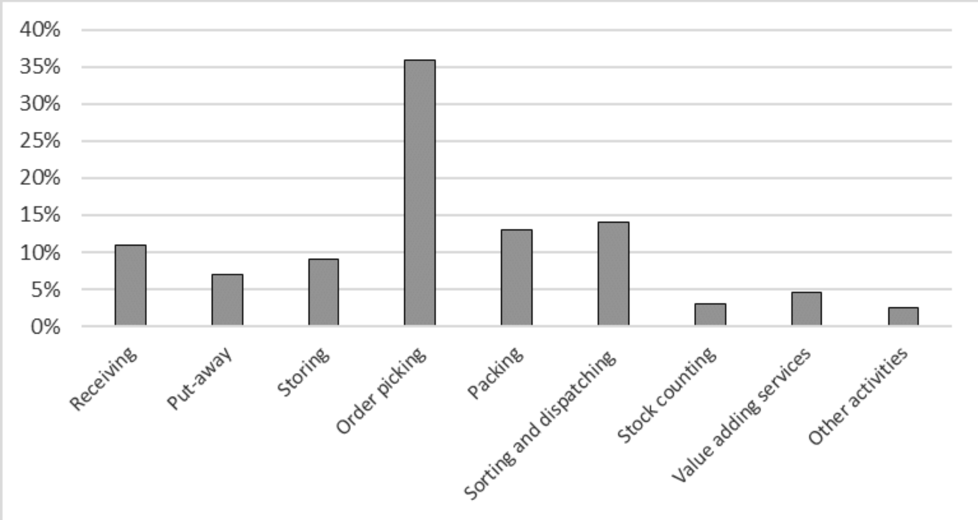


Figure 1.1: Warehouse activities as a percentage of warehouse costs [7]

The analysis reported in Skerlic et al. about the distribution cost for each different operations show that the most labour-intensive and time-intensive process is the order picking, which accounts for 36% of the total costs, but some authors estimate this value being at around 50% [7]. It is very easy to understand why the picking process is considered one of the most critical and why several authors focus their analysis and research work.

1.2 Picking process

The picking is the act of collection components and items from a storage area to fulfill an order aimed to satisfy a (internal or external) customer request.

The most common objective of order picking systems is to maximize the service level subject to resource constraints such as labour, machine and capital, where the service level is a combination of a variety of factors such as average and variation of order delivery time, order integrity and accuracy. [Goetschalckx and Ashayeri, 1989].

This definition allowed to go forward in the picking process comprehension and it is useful to split it in seven different tasks:

1. Travelling to, between and from pick locations;

2. Searching for pick locations;

3. Researching and bending to access pick locations;

4. Extracting items from storage location;
5. Documenting picking transactions;
6. Sorting items into orders;
7. Packing the items.

Also monitoring the performance of these is very important inside warehouses and can be done with seven different categories reported in table 1.1 where some measurement for monitoring are also indicated.

	Measurements
Time	Lead time Picking time Request information time
Throughput	
Human factors	Cognitive learning Ergonomic evaluation Picking errors
Quality	Order fill rate Picking accuracy
Flexibility	
Operational efficiency	Picking productivity Resource utilization
Costs	Investment Maintenance costs Order processing Inventory costs

Table 1.1: Performance categories and their measures parameters [8]

The analysis on the performance of those activities is extremely useful, in particular if we look at the measurements of time and costs. With the first measurements it possible to consider also indefinite sub-measurements of time for activities such as travel, research of components, request of information, extracting, counting, sorting and documenting but, basically, the only one that add value in the process is the extracting task.
In particular, according with Bartholdi and Hackman (2005) *”Travel time is waste. It costs labour hours but does not add value”* and the research’s work of Tompkins et al. (2003), showed in fig. 1.2, estimates it as the 50% of the total picking time.

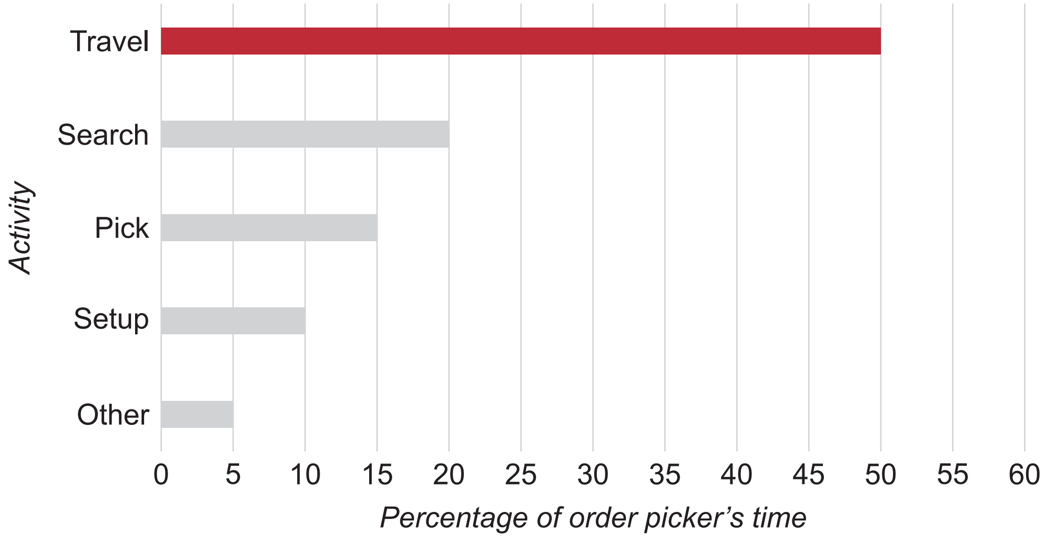


Figure 1.2: Typical distribution of an order picker's time [9]

There are different thing which may be implemented in order to improve the time performance in picking process, for example [9]:

- **Documenting**: automate the information flow, with solution as computer aided order picking, automatic identification systems, light-aided order picking, RFID systems;
- **Reaching**. The best location for the workers the waist level: system such vertical carousel, Miniload and person-abroad AS/RS are perfect solutions;
- **Sorting**. These activities depends by the picking policies, and, in particular, if one picker is assigned per order and one order per tour it is possible to eliminate;
- **Searching**: it is possible to reduce these time with pick-to-light systems that it illuminate the pick locations or with solutions able to bringing the pick locations to picker, or taking the picker to pick location (stock-to-pick solutions or person-abroad AS/RS)

Thanks to these strategies is possible to reduce the ”wasted” time overall the activity, but on travel time (or travel distance) they do not have a significant impact, therefore still the focus in optimization process.
The second important aspect in optimization is the total costs, which may include both investment and operational costs, such as show in table 1.1.

In particular, objectives which are often taken into consideration in warehouse design and optimization are [10]:

- minimize the throughput time of order;
- minimize the overall throughput time;
- maximise the space;
- maximise the use of equipment;
- maximise the use of labour;
- maximise the accessibility to all items.

These are examples, and obviously is necessary to find a trade-off between them; during the years, a lot of different picking solutions have been developed, each of them to optimize one or more of these variables.

1.2.1 Types of Order Picking Systems

Order picking involves the process of clustering and scheduling the customer orders, assigning stock on locations to order lines, releasing orders to the floor, picking the articles from storage locations and the disposal of the picked articles. As a warehouse function, order picking arises because incoming articles are received and stored in (large-volume) unit loads while customers order small volumes (less-than-unit loads) of different articles. Typically, thousands of customer orders have to be processed in a distribution warehouse per day and it attracts a lot of interest in research. Year after year a lot of different options have been developed, both manual and automated.
Jaghberr et al [8], during the 2020, have classified the existing OPS types that distinguishes it according to whether a human, robot, or no picker is used as shown in the fig. 1.3.

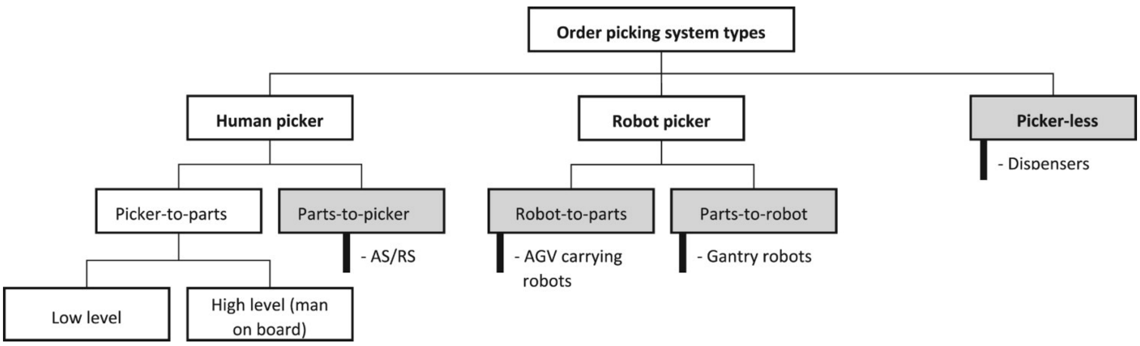


Figure 1.3: Classification of OPS types [8].

All the categories have some solutions that are completely manual, partially automated

or fully automated. Human picker OPSs can be distinguished in picker-to-parts, mainly manual systems where the picker moves around the warehouse to catch the components and in parts-to-picker systems, partly automated, where the picker is stationary and robots carry the articles in the picking station. With the same concept, in OPSs with a robotic picker, robots that move inside the warehouse area are found able to carry the items in the picking station whereabouts the human picker complete the operation with batching or possible packing. The last group of OPSs are fully automated whit no human or robot performing the picking.
Getting down to the details [10] [8]:

Human picker:

The majority of warehouses employ humans for order picking and, such as mention before, there are two logical subdivision: picker-to-parts and parts-to-pick.

Picker-To-Parts

This is the most common system [11] and it sees the picker walking or driving along the aisles to pick items from storage racks or bins (bin-shelving storage). We can distinguish two types of picker-to-parts systems: with walking operator (low-level) or with man on board of a lifting order-pick truck or crane (high-level).
It is possible to find several mission policies:

- Single order picking;
- Batch picking;
- Multi-batch picking;
- Cluster picking;
- Wave picking;
- Zone picking.

Parts-To-Picker

Parts-to-picker systems include automated storage and retrieval systems (AS/RS), using mostly aisle-bound cranes that retrieve one or more unit loads (pallets or bins; in the latter case the system is often called a mini-load) and bring them to a pick position (i.e. a depot). At this position the order picker takes the required number of pieces, after which the remaining load is stored again. This type of system is also called a unit-load or end-of-aisle order-picking system. Examples of those automatic systems are:

- Automated storage and retrieval systems (AS/RS): unit load, Miniload, shuttle

- Vertical lift modules (VLM);
- Conveyors;
- Carousels;
- Robot parts-to-picker OPSs.

Robot picker:

In this case the picker is a robot that autonomously picks the needed components. As the human picker, it can be moved to reach the parts or still fixed in a station, and also here we found robots-to-parts and parts-to-robot with the same logical.

Picker less (automated picking)

This is a process completely automated where the parts are selected and extracted using dispensers, vibro-feeder or other systems as A-frames.

1.3 Smart warehouse

In recent years, also in the logistics sector the technology have led to disruptive growth in order to respond at high efficiency and accuracy requirements. The concept of smart warehouse is used in very different fields: e-commerce, emergency departments, manufacturing and every kind of warehouses which needs to improve the customer satisfaction and reduce cost and errors.

According with Zhen at al. [12] the basic characteristics of smart warehouses are:

- **Information interconnection:** it is the base of smart warehouses and operational management and it is based on technology derived from:
  - Internet of Things (IoT),
  - Cyber-physical system (CPS),
  - Other emerging technology,

  All the information thus obtained can be shared, processed and saved by numerous operations and thus produce extra values.
- **Equipment automation:** describes the characteristics at strategic and tactical level. It represent the technical support of the smart warehouse, in fact, they can reach high automation levels for all warehouse activities. It allows to improve warehouse productivity while reducing the need for manual labor. The operations management

of smart warehouses, usually, are focused on strategic-level decisions for the equipment characteristics and on tactical-level decisions for product characteristics enabling a holistic view of technology and improving the precision of decision-making.

- **Process integration:** is one of the requirements in a smart warehouse. It tries to implement overall planning among various processes and focuses on the problem arising. In other words the objective of process integration is to achieve coordination and reduce, until their complete elimination, the discordance in warehouse operation management.
- **Environmental sustainability:** day by day sustainability increases its importance, for develop a smart warehouse we must focus on different aspects, not only on cost, such as energy cost, but also on carbon emission. All the perspectives in operations management must should should take these factors into consideration and be implement in an eco-friendly way.

Zhen at al. create a conceptual framework as shown in Figure 1.4. Here it is possible to see how the information interconnection is the base of smart warehouse, which provides information exchange channel and support for the warehouse system. The sustainability of the warehouse can be reached by exploiting the equipment automation and the process integration that are two pillars. The first one aims automatic operation through the design, deployment of automation and support at strategic and tactical levels, while the second one aims to control and coordinate al the activities and technologies in the warehouse. At the end, it is possible to define the warehouse operation management such as the theme through the four perspectives.
The previous four characteristics confer to smart warehouse the follow four attributes:

1. **Intelligence:** the information technologies allow to realize intelligent decision-making on common logistics problems;
2. **Flexibility:** the smart logistics has a higher degree of flexibility due to its more accurate demand forecasting, better optimization of inventory and more efficient transportation routing;

3. **Integration of logistics:** the use of technologies enable centralized management and strengthens the coordination of different logistics processes;

4. **Self-organization:** the real-time monitoring and intelligence decision-making permit the logistic system to work without significant human intervention

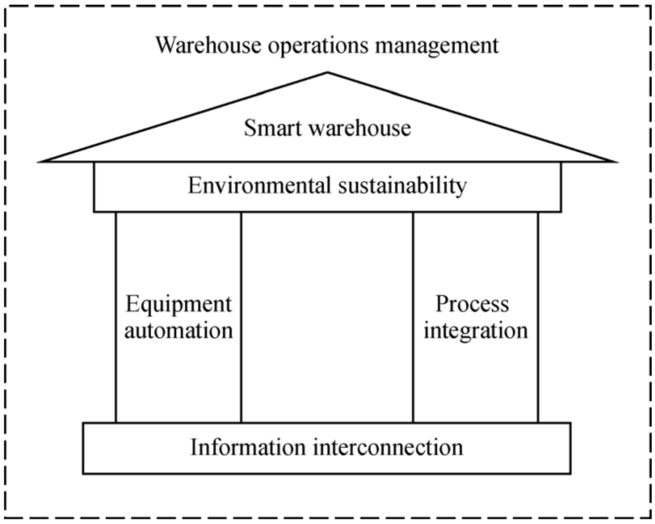


Figure 1.4: Conceptual framework of a smart warehouse. [12]

1.3.1 Information interconnections

In today's warehouse practice, RFID, warehouse management system (WMS), augmented reality (AR), reinforced learning, and other emerging technologies enhance the interconnection between different processes and entities [Dou et al., 2015; Bottani and Vignali, 2019; Sartoretti et al., 2019]. Interconnection technology provides ways to establish warehouse information collection and exchange, which could be a leading factor for future warehouse development. IoT and CPS technologies, including Radio frequency, pick to light, and pick by voice, are commonly used in the warehouse picking process.
The major affecting factors for selecting automatic identification technology include organizational, operational, structural, resources, external, environmental, and technological factors [Hassan et al., 2015].

RFID

RFID is the first technology for automatic object identification and data collection based on radio waves. Among the interconnection technologies, RFID is considered a choice for positioning, identifying, information interaction and warehouse management in smart warehouses [12]

The application mainly focuses on inventory tracking, localization and warehouse management with a significantly impacts on the warehouses' communications. The main part in RFID systems are:

- Tags: that are attached to product at the item level instead of the pallet level;
- Antenna: that are able to read the radio waves
- Reader: that collect and elaborate information from antenna.

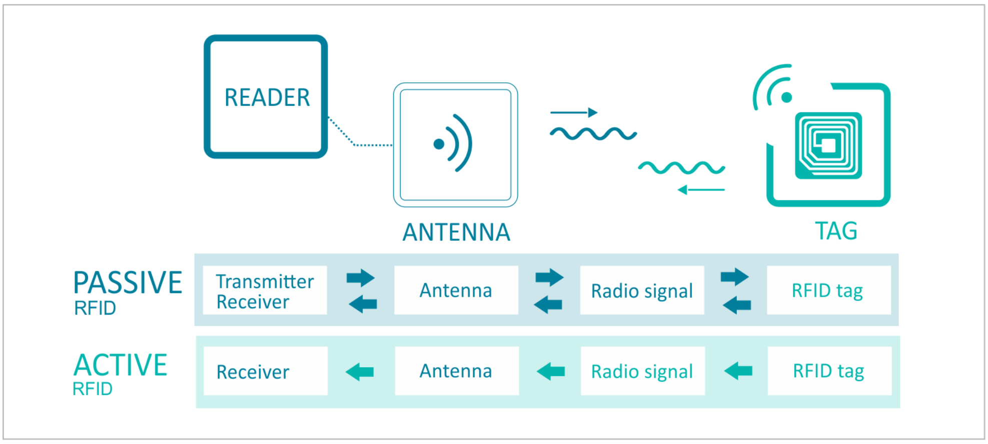


Figure 1.5: RFID working schema [13]

There are two types and they differ in the power source: active RFID systems have tags with their own transmitter and power source, mostly a battery or solar cell, while in passive RFID the antenna send a radio wave to tag that activeted it.

WMS

WMS is a computerize database that allow the warehouse control and optimization. Usually it include process as receiving, storage, order picking, packing and shipping.

1.3.2 Equipment Automation

Inside a warehouse, usually, the operations are a labor-intensive activity and the managing of it has become a critical problem for performances.

Automation allow to reduce the manual labor, the costs and the errors. Automated parts-to-picker OPSs have become more and more popular for warehouse operations (Tappia et al., 2019) and his increase is significant. The mainly systems are:

- AS/RS: are computer, and robot-aided, systems that can retrieve items or store them in specific locations

- AGV: is a driverless material handling system for movement inside the warehouse area;
- RMFS, also called KIVA system, is an automated, parts-to-picker picking system able to carry the racks with products toward warehouse and his stations.

Usually the parameters that the managers took into considerations for the automation decisions are multiple: acceleration/deceleration and speed for AS/RS systems, robot driving behaviour and battery capacity for RMFSs because it influence the warehouse's design and the performance of the systems. Other important decisions are the tactical ones, such as the storage policies.

1.3.3 Process integration

All the decisions inside a warehouse are logically interrelated and we can analyze four topics: *order processing, location assignment, resource allocation and differentiated service and blocking rearrangement and conflict-free routing*. Orders usually contain the demand information and attributes and the order sequencing determines the sequence of processing it.

Order batching is one way to reduce the inefficiency of picking, thanks to the union of different order. The opposite approach is the split order, that it plan to divide the order in two or more parts for picking, with different pickers that fulfill it. These second approach is the the one use, usually, in RMFS systems.

One important decision is about the location of the different items inside the racks of the warehouse. As always the objective of this decision is to minimizing travel time or the number of visits for each location.

1.3.4 Environmental sustainability

The smart warehouses use a lot of automation and this results in a higher consumption of energy compared by a traditional, manual, warehouse and, consequently, higher costs and higher carbon emission in nature.

These factors make us understand the importance of goals in sustainability, such as energy efficiency, cost reduction, throughput time minimization and the main decisions that the warehouse management can be able to take are:

- Trying to minimize the energy consumption with system able to improve picking efficiency and with a proper storage assistment;
- Work on the schedule of the various operations in the warehouse and find the best strategy about sequencing and scheduling.

2 Robot Mobile Fulfillment Systems

”The beauty of our system is that you don’t have to walk over to the shelves to get things—the shelves come to you.”. With these words, in 2008, Raffaello D’Andrea, one of the inventor of Kiva robotic, define the concept of Robot Mobile Fulfillment System.
With Robotic Mobile Fulfillment (RMF) system we mean a system where robots are able to lifting and carrying movable shelves called pods in the warehouse area or transport them to the pickers, who work in ergonomically designed workstations.
This kind of warehouse are very useful for e-commerce warehouses or in every type of warehouse where it is found large assortment of small products and the demands consists of a multi line small-quantity order. The system are flexible is operations and it is possible to add more movable shelves and autonomous mobile robots (AMRs). [14]
In this chapter we analyze briefly the born and the evolution of RMFS systems form the beginnings to present days. In section 2.2 we speak about the different areas and the two main process that is possible to find in this kind of warehouse, while in section 2.3 we examine the main decisions for the RMFSs. At the end of this chapter, in section 2.4 we investigate the parameters for the performance evaluation

2.1 RMFSs’ history

In the first years of the third millennium, the use of e-commerce has considerably increased, leading companies to research more efficient systems for warehouses management and picking operations performance improvement, in order to reduce costs and inefficiency.
In fact, e-commerce orders require to handle smaller quantity of materials, with a higher frequency in really short delivery time, as the customers want. For the aforementioned reasons, in 2003, Mick Mountz et al. started to study a new concept that led to the birth of Robotic Mobile Fulfillment Systems.
Mountz, together with Peter Wurman and Raffaello D’Andrea, founded the Kiva System, a company that pioneered the use a RMFSs in warehouses and facilities. This new systems

are able to fix the inefficiency related to long walking distances for the pickers, or, from a different point of view, are able to reduce picking operation's long time and high cost. The investors trusted in Kiva System, to the extent that they invested around $1.6 Million to help the company foundation and, a few years after, in 2009, the company to became the 6th fastest growing company in the US thanks to the e-commerce market growth and the important role that these system play in it. This innovative systems achieved the real success and a large dissemination when Amazon.com Inc. decided to buy Kiva in 2012 for $775 Million. [15] With the acquisition, the Kiva Systems has changed his name in Amazon Robotics and it has stop selling the AGVs and the warehouse software systems, but not the developing it.

2.1.1 RMFS in current days

Today, ten years after the Kiva acquisition, Amazon counts more than 520.000 robotic drive unit that are perfectly integrated with workers and the systems around.
On June 2022, the company presented a new fully autonomous vehicle able to guarantee safety for the workers thanks to special sensors, and this means that the specific areas where the AGV has to stay is not required. Another innovation, always developed by Amazon, is a Containerized Storage System to increase the performance, the safety and the comfort of workers. This new system is able to move a mobile box from and inside the various shelves where it is put. This robot, connected with a software, can find a container with a needed product, figure out how to move it to the employee and pick up the container once the employee has retrieved the product. This cuts down the need for employees to reach, bend or climb to retrieve items. [16].
Looking outside the Amazon world, after the Kiva Systems' acquisition a lot of new providers created very similar technology to cover the marketplace. Iam Robotics, Locus Robotic, 6 River Systems or InVia Robotics are only some examples that were evolved.

2.2 Solution overview

It is possible to say that RMFS is a very easy concept, with a complex software that manage all the components. There are two main processes: the replenishment operation that allow to store the items in the shelves and the picking operation that is useful to get the components to fulfill customers' orders, in the middle that found the AGVs who carry the shelves between this stations and the warehouse.

To describe, in detail, how this system works is helpful to use the fig. 2.1 that Xie et al elaborate from the Patent Application of Mounts at al..

2.2.1 Areas in warehouse

In fig. 2.1 there are three physical areas:

1. **Picking area**

   This area is reserved to picking stations and usually it's collocated close to the storage area. Each station basically needs: space, to host the pods from which the picker handles the items, human operator and some system, such as a conveyor to move outside the station the items collected.

   We can describe the flow in this area: the pods arrive in front of the picker, wait that the operator remove the articles that are previously ordered and after the shelves leave the area.

2. **Replenishment Area**

   For the correct functionality of the system, pods must have a minimum quantity of components stored inside. For this reason there is also the replenishment area to refill the shelves. Also in this area, we need some space to host the pods during the operation and one human worker, or smart robots, to do the activities.

3. **Storage Area**

   The storage area is the place dedicate to stock the pods when they are not in movement, or in the previous stations. Here it is possible to find full, half full or empty shelves that are waiting for one robots to be carry in the correct areas.

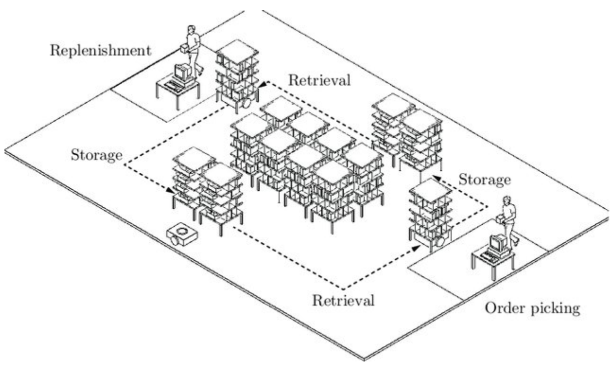


Figure 2.1: Basic layout of an RMF system [17]

2.2.2 Process overview

Figure 2.1 tracks the two main processes that happen in this kind of warehouse: *retrieval and storage operations*. The first one, retrieval process, refers to the replenishment process or the picking process. With these two different operations, after the arrival of replenishment or picking order, the AGVs raise the selected pods in their location and carry it in the station to store units inside or to pick the components in the order lines. The other process, called precisely storage process, means the operations by which AGVs bring the pods in the storage area, usually restricted to workers.

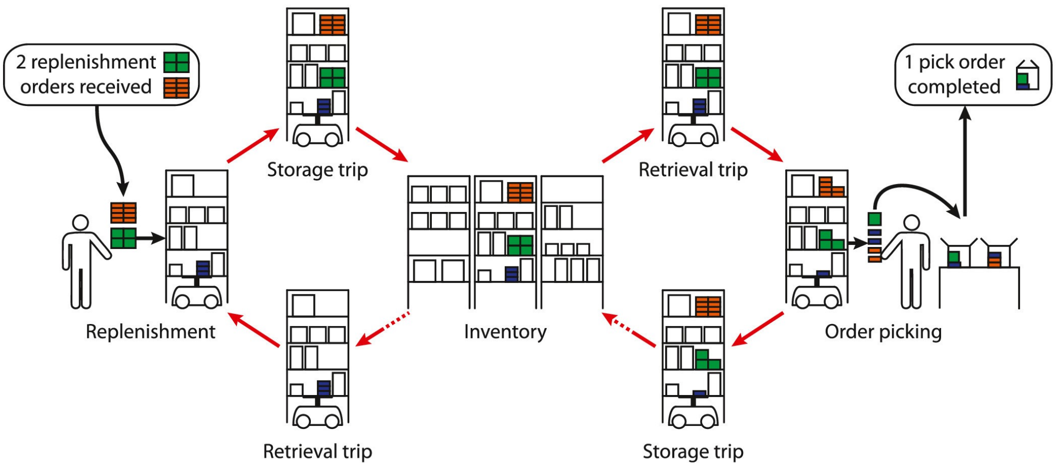


Figure 2.2: The process inside a warehouse with RMFS

In general, we can say that robots could navigate on the grid-paths through the warehouse using a waypoint system or stay in the their home-station if they have no pods to carry. Their performances impact in significant manner on the entire warehouse. For this reason, Xie et al. resume the [18] where Merschformann et al. analyze very well the decision problems [17].

2.3 Decision Problems in RMFS

As mentioned in chapter 1 there are three levels in the warehouse management decisions: strategic, tactical and operational. These three decision are repeated also in an RMFS warehouse and as represented also in fig. 2.3. Merschformann et al., in 2019, analyze the operational level and structure the decision problems in four different steps:

Figure 2.3: Hierarchical overview of decision problems and their relations [18].

1. **Order Assignment (OA)**: that provides the assignment of orders to workstations;
2. **Task Creation (TC)**, that means the creation of tasks for the robots;
3. **Task Allocation (TA)**, namely the allocation of duties to robots;
4. **Path Planning (PP)**, precisely the creation of the paths along which the robots will move.

2.3.1 Order Assignment

The order assignment means the *pick order assignment (POA)* and the *replenishment order assignment (ROA)* mentioned in table 2.1.

Abb.	Name	Description	Trigger
POA	Pick Order Assignment	Choosing a pick order from the backlog	When the order is fulfilled, creating space for the next assignment
ROA	Replenishment Order Assignment	Selection of the replenishment station for completion the operation	When the replenishment order is created and one or more stations have capacity left after the RPS assignment

Table 2.1: Details of Order Assignment (OA) [18]

2.3.2 Task Creation

Thanks to fig. 2.3, we can see that also the creation task can be split, in fact this step include *Pod Selection (PS), with Pick Pod Selection PPS and Replenishment Pod Selection RPS* and *Pod Storage Assignment (PSA)*. In general, this stage allows to understand which assignment has to be selected for transportation based on the orders inside the systems, such as picking, replenishment and storage. The TC uses the order assignment to select suitable pods and subsequently convert this in movement between storage area and workstations.

Abb.	Name	Description	Trigger
PS	**Pod Selection**	Select the suitable pod for complete the operations	-
PPS	Pick Pod Selection	Selection of pod to transport to a picking station	When an AGV for pick station needs a new task
RPS	Replenishment Pod Selection	Select a pod for the next replenishment operation	When the replenishment order is created in the system and it has sufficient capacity left
PSA	Pod Storage Assignment	Choosing a storage location for a pod	When a pod leaves a workstation

Table 2.2: Details of Task Creation (TS) [18]

2.3.3 Task Assignment and Path Planning

The TA creates the right sequence of tasks for the robots to execute and with that is also necessary to define the path planning algorithms. This last part was well investigated in [19] where L. Luo et al. determine the best path policy based on the general layout of storage area.

2.4 Performance of RMFSs

For the evaluation of RMFS' performance we have to consider a lot of different parameters and the study of literature may help. In 2020, Jaghbeer et al. presented a systematic review and content analysis to understand the most relevant performance OPSs [8]. In particular, for RMFS it is important to examine the performance of robotic part-to-picker OPSs that are summarized in table 2.3

Performance categories	Studied performance aspects
Throughput	**Throughput** (Bauters et al. 2016; Lamballais, Roy, and De Koster 2017; Lamballais Tessensohn, Roy, and De Koster 2020; Roy et al. 2019
Lead Time	**Average order cycle time** (Ekren and Heragu 2010; Lamballais, Roy, and De Koster 2017)
	Picking time (Xue, Dong, and Qi 2018; Zou, Xu, and De Koster 2018)
	Throughput time (Yuan and Gong 2017; Roy et al. 2019)
Human factors	**Ergonomics** (Lee, Chang, and Choe 2017; Hanson, Medbo, and Johansson 2018)
	Operator training (Hanson, Medbo, and Johansson 2018)
Quality	**Picking accuracy** (Hanson, Medbo, and Johansson 2018)
Flexibility	**Flexibility** (Hanson, Medbo, and Johansson 2018)
Operational efficiency	**Robot utilisation** (Lamballais, Roy, and De Koster 2017)
	Uptime (Hanson, Medbo, and Johansson 2018)
	Collision-free paths (Kumar and Kumar 2018)
	Waiting times for vehicles (Ekren and Heragu 2010)
	Average utilisation of vehicles and lifts (Ekren and Heragu 2010)
	Efficiency (Zhao et al. 2019)
	Picker and robot utilisation (Wang, Chen, and Wang 2019)
	Robot travel time (Wang, Yang, and Li 2019)
Costs	**Costs** (Boysen, Briskorn, and Emde 2017 ; Li et al. 2017)

Table 2.3: Performance aspects in literature's studies [8].

2.4.1 Throughput

In detail, the studies in literature reported in table 2.3 about throughput analyze it such as:

- Beuters et al., in 2016, analysed the performance of RMFSs capable of lifting and moving inventory pods and compare them with the performance of AS/RSs, founding that the RMFSs' throughput is higher and it's depends by the number of AGVs and by the number of SKUs for each rack;

- Lamballais et al., in 2017, found thanks to a model, that the maximum throughput is affected by the location of workstations;
- Roy et al., in 2019, did an analysis about the effect of robot assignment strategies on throughput.
- Lamballais et al., in 2020, discovered that the throughput increases when spreading inventory across multiple pods and when there is an optimal ratio between the number of different stations and when the pod is replenishment before it is empty;

2.4.2 Lead time

In particular for the lead time the studies conducted about it examine the followed aspects:

- Ekren et al., in 2010, investigated the effects of warehouse height and footprint in average cycle time;
- Lamballais et al., in 2017, using models were able to estimate average cycle time in RMFS systems;
- Yuan et al., in 2017, did the evaluation for the throughput time of an RMFS by comparing two robot-sharing policies and thier effects;
- Xue et al, in 2018, created a comparative analysis with three different picking strategies and studied the effects on picking time and the travelled distance for each robots;
- Zoe et al., always in 2018, studied the battery management strategies;
- Roy et al., in 2019, analyzed the robot assignment strategies and their effects on throughput.

2.4.3 Human Factors

The only two studies that Merschformann found research the follow aspects:

- Lee et al., in 2017, did evaluations about the ergonomics in RMFS that have higher risk factors than AS/RSs;
- Hanson et al., in 2018, studies the performances and the link between it and design, with a focus on ergonomics and operator training.

2.4.4 Quality and flexibility

The studies conduct by Hanson et al. found that robot design with regard to sensors and battery management strategies affects RMFS flexibility.

2.4.5 Operational efficiency

Operational efficiency is addressed by different authors:

- Ekren et al., in 2010, studied the effect of warehouse height and footprint on the waiting times for vehicles and the average utilisation of vehicles and lifts in an AVS/RS.
- Lamballais et al., in 2017, estimated robot utilisation in RMFS;
- Hanson et al., in 2018, found a correlation between the uptime of robot sensors and battery management strategy as well as between robot sensors and operational efficiency;
- Kumar et al., in 2018, developed a robot routing algorithm that results in a collision-free path for RMFSs;
- Zhao et al., the follow year, found that order sequencing affects efficiency in AVS/RSs;
- Wang et al., always in 2019, found that the routing strategy in RMFSs affects picker and robot utilisation;
- Wang et al., in the same research, found that different RMFS layouts affect robot travel time.

2.4.6 Costs

The costs were invesigated by:

- Boysen, Briskorn, and Emde in 2017;
- Li et al. always in 2017.

3 Simulation Approach

Since we are children playing with games, simulation is part of our lives. During a simulation, it is possible for players to imitate a process, a life or another situation. In general, it can be said that the simulation is used to model, improve design and management of different kind of systems with many levels of detail.

In companies, university and organizations the simulation is a tool that may give a significant benefit in several fields and helps in the study of real functions. Before concrete application, a simulation helps to maximize the performance, finding optimization or discovering new features, with less costs and less constraints compared to the concrete application. A key advantage of simulation modeling is that it has the capability of modeling the entire system and its complex interrelationship. The representational power of simulation provides the flexible modeling that is required for complex processes. Moreover, its results take into account the interaction of all important parameters. [20]

In this chapter we will try to understand the main characteristics of simulation and why it is useful in different fields. In section 3.3 we will analyze the problem solving approach and the general simulation methodology and while in section 3.4 we will try to understand and analyze the output of simulation modeling. The last section of this chapter introduces Anylogic™.

3.1 Simulation

The most common definition of simulation refers to the process of translating a conceptual model of some system thanks to special software program. A simulation model represents a system involving a set of real or hypothetical elements interconnected to give the systems an overall identity and behavior. Systems can be of different kinds: from atoms in a molecule to physical models such as flight simulators or role-playing game, they all have one main characteristic: they describe and capture the most important relationships among the elements inside the model.

One of the main advantages of simulation is to offer to researches the ability to extend knowledge or real or hypothetical systems and collect and analyze data. A most clear definition of computer simulation is defined by Ravindran et al. in 1987 as ”a numerical technique for conducting experiments on a digital computer which involves logical and mathematical relationship that interact to describe the behavior of a system over time.” Before getting into details, it is useful to define systems and their classification.

”A system is a composite of people, products and process that provide a capability to satisfy states needs. A complete system includes facilities, equipment, materials, services, data, skilled personnel and techniques required to achieve, provide and sustain system effectiveness.” [Air Force Systems Command 1991]

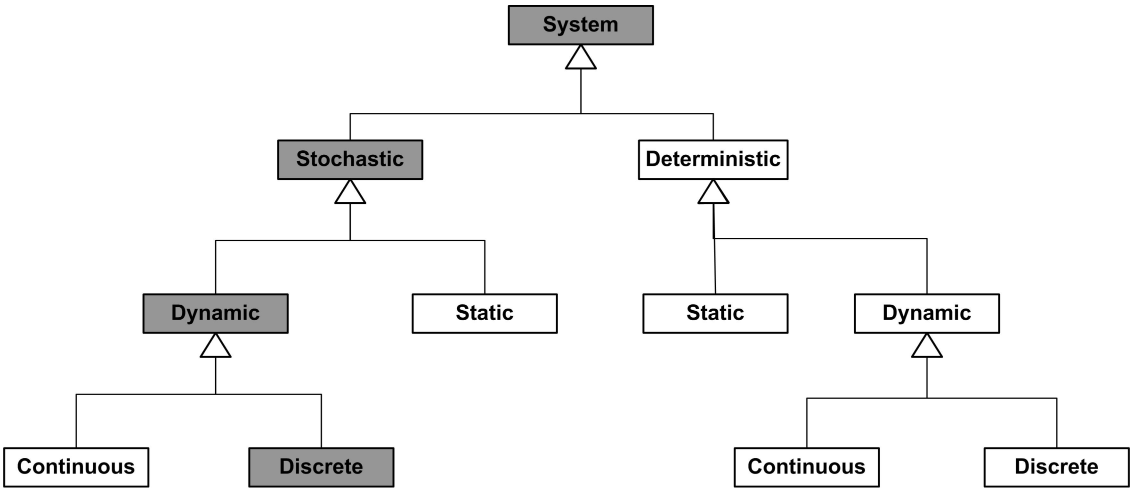


Figure 3.1: System Classification [20]

In simulation there are two different points of observation during the time: 1- Discrete event and 2- continuous. In the first mentioned, the observations are gathered at a selected point in time when certain changes, called events, take place in the system. In the continuous simulation, instead, the observation have to be continuous in time, or described for all points in time.

The second level of systems classification is based on the evolution. In static systems time is not a significant factor and in particular it is possible to say that the state does not evolve over time. On the contrary, in the dynamic systems the state change over time. The last categorization subdivides the systems by the variables in the process: systems that can be called deterministic, when the variables are not governed by underlying random process; and systems that are defined stochastic, when only some of the variables

are governed by underlying random processes.
In particular, in simulation the considered systems are stochastic, meaning the time change continuously at discrete points, or in other words at events.

3.1.1 Randomness in simulation

When talking about real-life actions, the process does not have always the same time, even though the activities are the same, and it translates in events that occur at a random fashion in time. To convert this into the simulation world we talk of phenomenons as random variables governed by a probability distribution, and this last one is decided thanks to real data, usually historical ones, or using plausible assumptions.

3.2 Why simulation?

Manson et al., in 2020, have identified some advantages and challenges of simulation [21]. In particular, they consider simulation as a ”third way” of doing science since it mixes induction and deduction. With induction, they mean the generalization from empirical observation to theory, while with the deduction the test of theory against empirical observations.

3.2.1 Advantages of simulation

1. Ability to model possible outcomes for situation that do not exist in reality, situations exist in the future or that are dangerous and expensive to perform directly in the reality;
2. Facility to combine mathematical, logical and natural language;
3. Utilization of simulation as tool to create knowledge;
4. Capability to integrate a large amount of data and information on relationships between elements in a systems;
5. Ability to incorporate random noise or a range of values for individual parameters in a way that is unavailable to mathematics;
6. Potential to examine the effects of changing inputs on outcomes and the amount of possibilities of these;
7. Less costs than real systems implementation;

8. Capability to study a systems in modules, and possibility to use divide-and-conquer strategy to solve problems;
9. Possibility to share a simulation model and simulation results;
10. Possibility to chose the abstraction level, it's easier to develop a simulation model than an analytical model. It typically requires less thought, and the development process is scalable, incremental, and modular.

3.2.2 Simulation's challenges

1. Required to calibrate, verify and validate the model. This means that the model has to fit with model and theory, determine whether the model runs well and compare the structure and output with different data from calibration process;
2. Simulation could be incorrect, mainly in case of errors and poor programming;
3. Simulation model creates a large number of outputs and it means that someone have to analyze it;
4. Simulation require a large amount of data;
5. Result depends strongly from input data;
6. Simulation returns only approximations, estimates. In other words, it do not get exact answers.

3.3 Simulation methodology

3.3.1 Problem solving

A simulation methodology is a particular set of procedures based on the general precepts of solving a problem through systems analysis as follows:

1. Define the problem;
2. Establish measures of performance for evaluation;
3. Generate alternative solutions;
4. Rank alternative solutions;
5. Evaluate and iterate during process;

6. Execute and evaluate the solution.

It is also called DEGREE methodology from the first letter of the six steps. The step number one is useful to solve the right problem, and, once this is done, to define the right performances' measures coherently with the problem (step two). Steps number three and four are used to look and evaluate multiple solutions, and are helpful to find the best solution. The last two remaining steps are the evaluation and the iteration during the process and the execution and the evaluation of the solution. Iterations are an important concept in analysis needed to recognizes if the problem solving process can be repeated until the desired degree of modeling fidelity has been achieved.

3.3.2 General simulation methodology

When we talk about simulation, the DEGREE problem solving methodology needs to be slightly adapted to the simulation, or in other words adapted to how simulation interacts with the overall problem solving process [20].
In particular, the DEGREE methodology evolves, as it can be seen in Figure 3.2 into six main phases and some under-phases for each step:

1. **Problem Formulation**
   (a) Define the problem;
   (b) Define the system;
   (c) Establish performance metrics;
   (d) Build conceptual model;
   (e) Document model assumptions;
2. **Simulation Model Building**
   (a) Model translation;
   (b) Input data modeling;
   (c) Verification;
   (d) Validation;
3. **Experimental Design and Analysis**

(a) Preliminary Runs;
(b) Final experiments;
(c) Analysis of results;

4. **Evaluate and Iterate**
   (a) Documentation;
   (b) Model manual;
   (c) User manual;

5. **Implementation**.

Problem formulation

The problem formulation is the fist step and also one of the most important: the five sub-phases are useful to understand what needs we have to solve. The output of the problem's definition is called also problem definition statement and is a narrative discussion of the problem: it has to be accurate, concise and to contain all the assumption that the analyst wants to explore and also the goal of the study.
When the problem is well defined, it is necessary to define the system with his boundaries, and, usually, it is helpful to use one pictorial representation of the major elements.
The third step of problem definition, requests to define metrics to do the analysis of the system performance with which the alternative scenarios need to be compared. Here quantitative statistic measures for the model, quantitative measures for the system (e.g. cost/benefits) and qualitative assessments are included. The focus should be placed on the performance measures that are considered to be the most important to system decision-makers and tied directly to the objectives of the simulation study. When all the previous activities end, it is possible to create a conceptual model that is a graphical description of the problem, the process and represents practical steps that need to be translated into a computer representation. Generally context diagrams, activity diagrams or software engineering diagrams are used. Lastly, at the end of all these activities, it is fundamental to organize all the information, in order to help other people to understand the problem and do the final evaluations.

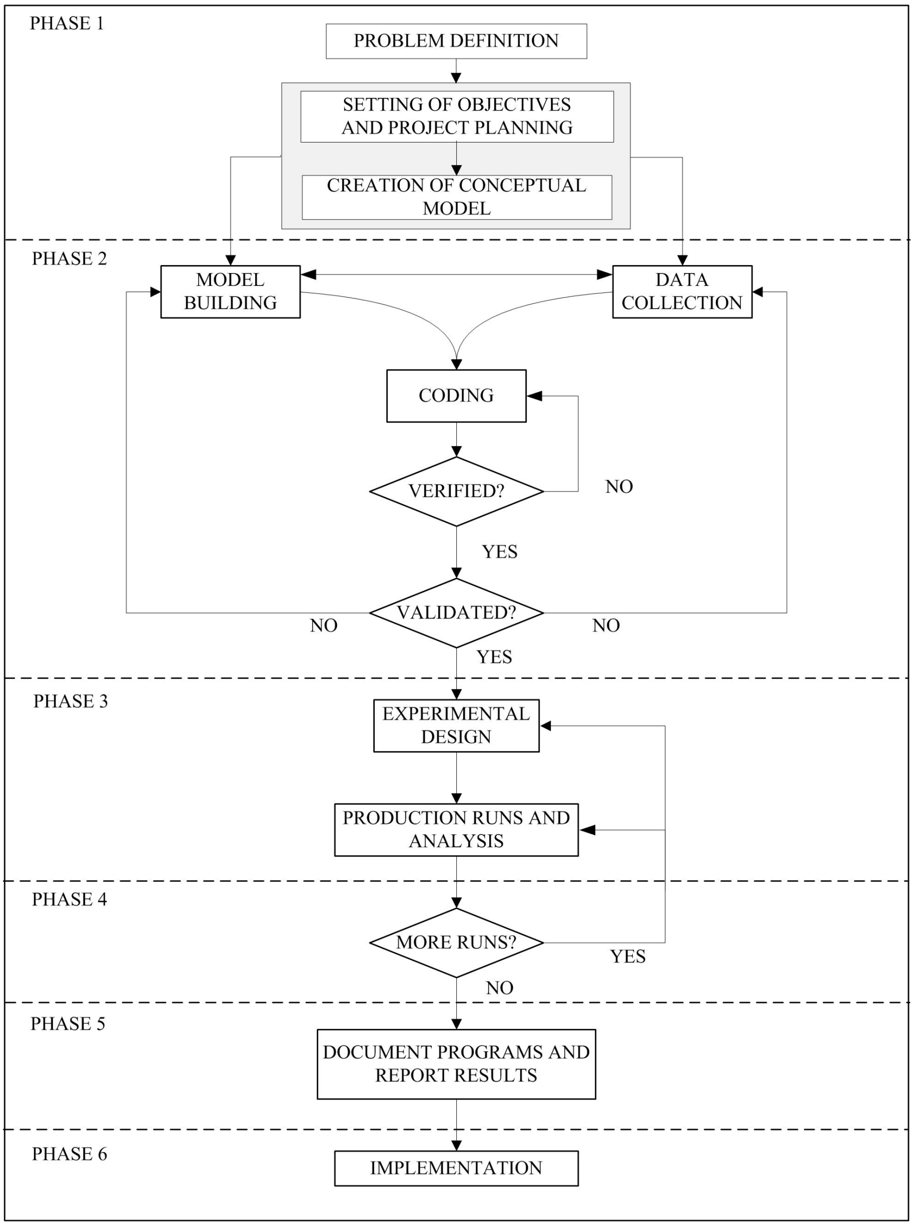


Figure 3.2: General Simulation Methodology [20]

Simulation model building

In the simulation model building phase, the second main phase of the whole process, alternative systems configurations are developed thanks to the previously developed conceptual models to evaluate the alternative solutions that need to be verified, validated and prepared for analysis. As seen in the previous steps, the model translation has to be moved into input data modeling, verification and validation. In details, the input data preparation concerns the data analysis to determine if data are enough and to classify them into several areas.

The verification is performed to determine whether the program performs as intended. Usually to do it we search errors in the simulation code, in logical or other failure. To improve the model debugging it is possible to stress the model with a sensitivity analysis or/and testing the individual modules within the code.
The validation is used to determine whether or not the simulation model adequately represents the real system. In fact, the correct input data permit to replicate the real situations as we want to demonstrate in the simulation projects.

Experimental design and analysis

After the over-mentioned phases the model use begins to perform experiments that investigate the topic of the project. The three step in this phase are *Preliminary run, final experiments and analysis of results*. The first one should be performed to set the statistical parameters associated with the main experiment and should be used to generate benchmark statistics of current system operation.
The experiment have to be replicated to generate comparative statistic, especially if the data model have stochastic characteristics, which involves a continuous process, as a cycle.
This is very important where there are a significant number of design factors that can affect the performance of the system replicated with the simulation model and it is important to use the formal experimental design or advantage output analysis techniques as statistical concepts applications.

Evaluate and iterate

The system decision maker have to define is also the specific criteria to evaluate the model and how to use the simulation results in order to analyze and rank the scenarios. Each of them, in fact, have multiple performance measures that trade-off against each other. If data are not enough it is necessary to iterate the simulation run. In this phase is also important to document the solution: a good documentation consist in two main parts.

- Technical manual which can be used by the same analyst or by others and is very important for software re-usability and portability;

- User manual used by the normal person not analyst, usually not too much expert in programming and in simulation.

In each documents we have to be clear and simple.

3.4 Output analysis

Rossetti in ”Simulation Modeling and Arena” [20] presents also a methodology to examine the output data.

As referred chapter 7 of his book, the inputs in a simulation are random, hence also the outputs are random, precisely a sequence of pseudo-random numbers.

For this reason it is important to understand how the simulation works to control it and perform some advantages during the decision making process.

Ripley, in 1987, gave a precise definition of pseudo-random number that is a deterministic sequence of number having the same relevant statistical properties as a truly random number and for this reason we can use the statistic to generate the inputs.

3.4.1 Output analysis using replications

Simulation is like a mathematical function with specific parameters in a random distribution, or, with other words, the output is a combination of the actual performance of the system and some noise, steaming from the randomness of your system.

This noise have influence on output and on the KPIs that we decide to evaluate in an unknown degree, and, for these reason, it is useful to appeal to replications. Replication is a simulation run with the same inputs but with parameters change, with the right data distribution, in each run. A good method to visualize the effect of the noise on the output result is plotting it as in Figure 3.3 where we see that the KPI in the reality is floating due the parameter noise.

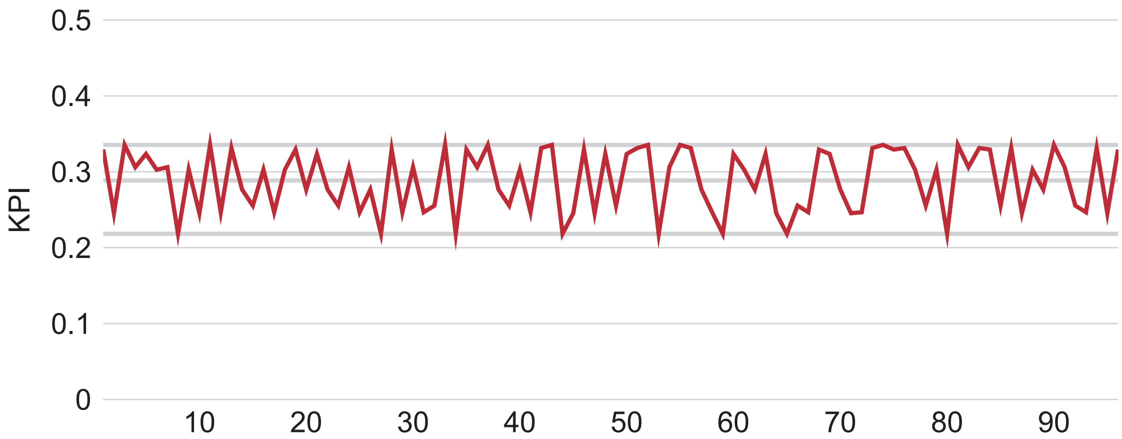


Figure 3.3: Generic KPI example

This simple diagram clearly outlines that addressing that randomness of one's output is key to establish the best possible foundation for decision making. If only one experimental run is simulated and it is assumed that this single run's output result is representative, the risk is to make decision based on a KPI that most likely will lead to wrongful decision.

Multiple replications and reporting of the output result(s) as across replication statistics will allow to direct the noise from the randomness expressing the actual performance of the system. This is only possible with the continuous across replications mean of KPI, calculated for each replication *i=1, ..., 100* by eq. (3.1)

$$\bar{X}_i = \frac{\sum_{j-1}^{i} X_j}{i} \tag{3.1}$$

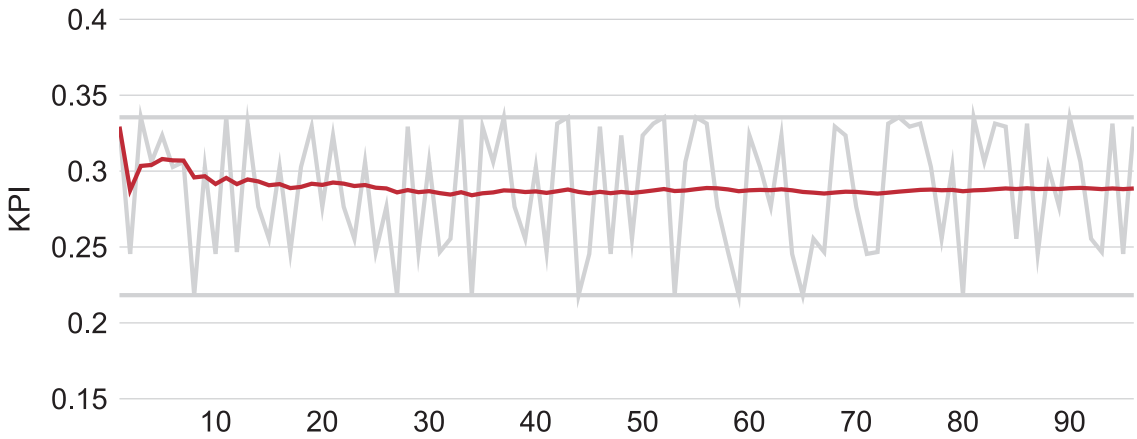


Figure 3.4: Graph illustrating the across replication mean of the KPI

In fig. 3.4, the benefit of multiple replications using the across replication KPIs for decision making is visualized: here the randomness is addressed to instead visualize the actual system performance.

3.4.2 Number of replications

To ensure confidence in decision making process multiple replication and cross replication statistics are required, but it is necessary to define the correct numbers of replications. To do this three different approaches are introduce by Rossetti (2015) and Sheldon Ross (2013).

The first step is to define the confidence level for decision making that we could define as the variation across the mean of the KPI. We define the *required half-width, h* as the value that are able to satisfy the condition reported in eq. (3.2)

$$h \leq \epsilon \tag{3.2}$$

where ε is the error that we consider acceptable.

Half-width method

The first approach described by Rossetti (2015) is based on the normal distribution and the related t-distribution: when the value distribution of a random variable follow the normal distribution it is possible to use the t-distribution to define the required confidence in the reported mean value across independent samples. To calculate it, the Rossetti's method (2015) requires a pilot experiment with ten, fifthteen or twenty replications to get an estimation for the standard deviation. With this estimation, it is necessary also to calculate the half-width in the pilot experiment, h_0 as in eq. (3.3).

$$h_0 = t_{1-(\alpha/2),n_0-1} \frac{s_0}{\sqrt{n_0}} \tag{3.3}$$

With this initial half-width the required number of replications,n, became as eq. (3.4)

$$n \cong n_0 \left(\frac{h_0}{h}\right)^2 \tag{3.4}$$

Iterative process

The second approach above mentioned is an iterative process introduced by Ross, in 2013. That approach is focused on limiting the standard deviation observed in a large pilot experiment.

To use this method, the first task is to define the accepted standard deviation h, while the second step is to conduct a pilot experiments with n_0 replications and its standard deviation. The last required element is then the multiplying factor based on your required confidence level and accepted risk, α. The multiplying factor is calculated as in eq. (3.5)

$$\Phi^{-1}\left(1 - \frac{\alpha}{2}\right) \tag{3.5}$$

With the multiplying factor calculated, the comparison made to determine whether or not the n_0 replications are enough to ensure a standard deviation below the threshold value is illustrated in eq. (3.6)

$$\frac{\Phi^{-1}\left(1-\frac{\alpha}{2}\right)s}{\sqrt{n_0}} < h \tag{3.6}$$

If the eq. (3.6) is satisfied, n_0 will ensure (1-α)% confidence in the fact that the average KPI value calculated across the replications will not have a standard deviation of more than *h*. The method must be iterate until the eq. (3.6) is satisfied.

Single formula

The iterative process require an unknown number of different experiments. To counter this, a single formula, based on the standard normal distribution and an initial estimate of std. deviation can be derived from Rossetti (2015) and Ross (2013). It is therefore possible to utilise the single formula seen in eq. (3.7)

$$n > \left(\frac{\Phi^{-1}\left(1-\frac{\alpha}{2}\right)s}{h}\right)^2 \tag{3.7}$$

where *s* is the initial estimator of the standard deviation observed and it is necessary to conduct a pilot experiments as in the *Half-width Method* and it is possible to use it if *n* is bigger than fifty ($n \geq 50$).

These three different approaches to calculate the required number of replications to ensure confidence in your decision making have now been introduced. Which method to use will depend on one's preferences and practical setup.

3.5 Anylogic™

AnyLogic™ is the leading simulation software for business applications, utilized worldwide by over 40% of Fortune 100 companies and developed by the AnyLogic Company (former XJ Technologies). AnyLogic™ models enable analysts, engineers, and managers to gain deeper insights and optimize complex systems and processes across a wide range of industries in different fields: markets and competition, healthcare, manufacturing, supply chains and logistics, retail, business processes, social and ecosystem dynamics, defense, project and asset management, pedestrian dynamics and road traffic, IT and aerospace are some examples of that [22].

One of the advantages of AnyLogic™ is that it offers a free license for students, that, however, has some limitations to it, such as:

- Maximum of 10 agent types in one model;
- Maximum of 200 embedded agents/blocks in one agent;
- Maximum of 200 dynamically created agents.

It supports agent-based, discrete event, and system dynamics simulation methodologies and it is cross-platform simulation software that works on Windows, mac OS and Linux.

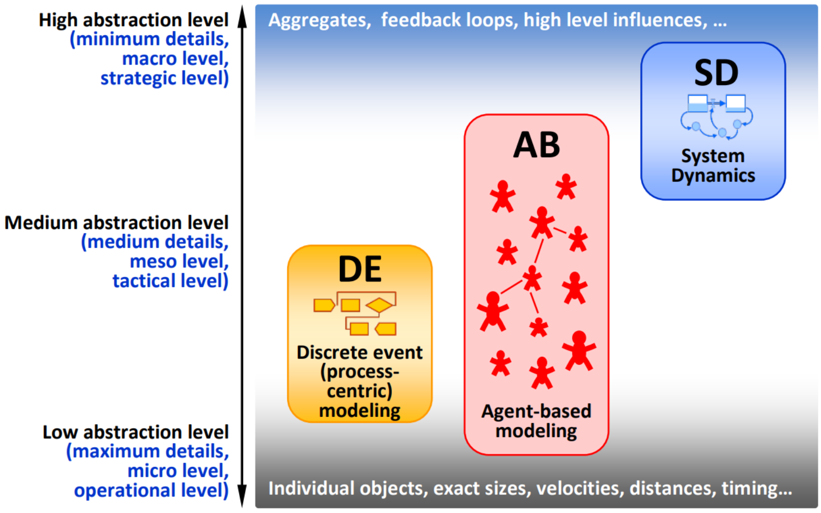


Figure 3.5: Methods in simulation modeling with discrete-event modeling. [22]

3.5.1 Discrete event modeling

A discrete-event simulation (DES) models the operation of a system as a (discrete) sequence of events in time. The idea of that is this: the modeler considers the system being modeled as a process anwhich can then be presented graphically as a process flowchart. Each block represent operations with specific characteristics (delay, queues, etc.) and each event occurs at a particular instant in time and marks a change of state in the system. Between consecutive events, no change in the system is assumed to occur; thus the simulation time can directly jump to the occurrence time of the next event, which is called next-event time progression.

Most business processes can be described as a sequence of separate discrete events. For example, one service system of a bank department, as in fig. 3.6, it's consisting of an automatic teller machine and teller lines. ATM provides people with a quick self-service for cash. More complex transactions, e.g. paying bills, are completed by tellers, allowing customers more time without inconveniencing those customers looking for quick cash. To simulate this, discrete-event simulation is often chosen.

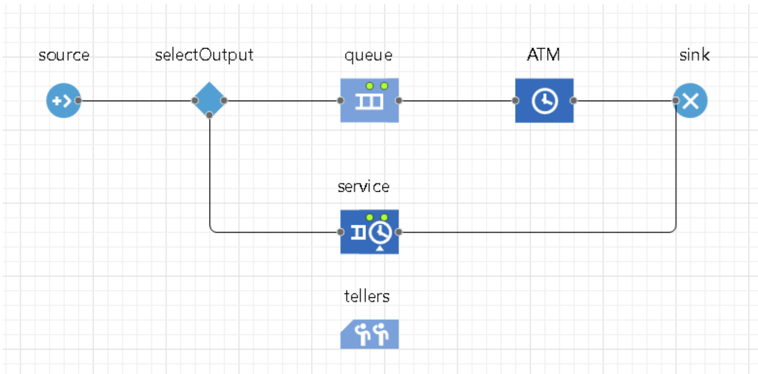


Figure 3.6: Example of simulation model of a bank with discrete event modeling [23].

Discrete-event simulation focuses on the processes in a system at a medium level of abstraction. Discrete-event simulation modeling is widely used in the manufacturing, logistics, and healthcare fields.

3.5.2 Agent-based modeling

Agent-based modeling focuses on the individual active components of a system. This is in contrast to both the more abstract system dynamics approach, and the process-focused discrete-event method.

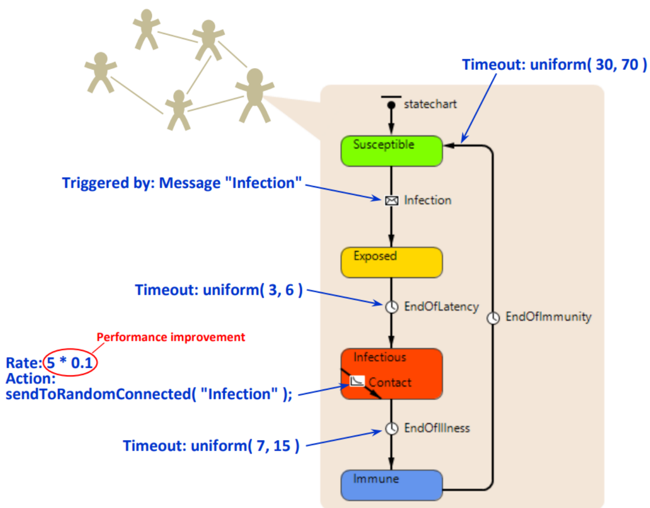


Figure 3.7: Example of agent based simulation model of epidemic [23].

With agent-based modeling, active entities, known as agents, must be identified and their behavior defined. They may be people, households, vehicles, equipment, products, or companies, whatever is relevant to the system. Connections between them are established, environmental variables set, and simulations run. The global dynamics of the system then emerge from the interactions of the many individual behaviors.

3.5.3 System Dynamics

System dynamics is a highly abstract method of modeling. It ignores the fine details of a system, i.e. the individual properties of people, products, or events, and produces a general representation of a complex system. These abstract simulation models may be used for long-term, strategic modeling and simulation. For example, a telephone network planning a marketing campaign may simulate and analyze the success of new data plan ideas without having to model individual customer interactions, as shown in fig. 3.8

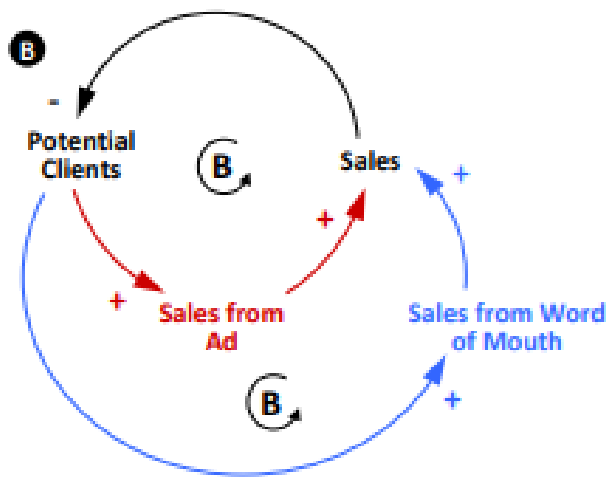


Figure 3.8: Example of system dynamics model of a new product diffusion [23].

4 My model

Simulation is a very strong tool to recreate environmental and company processes. In particular, the main focus of this work is the study of robotic mobile fulfillment systems with the aim to conduct experiments in a virtual environmental and find the best optimization of the RMFS's process.
As mentioned before, each simulation modeling requires a correct methodology. My model is developed following the general simulation methodology explained in *"Simulation modeling and Arena"* [20] : this chapter is, then, organized in the same way. In particular, I focused on the fist four steps of the general simulation methodology, since the last two phases are not within the scope of the study and therefore will not be elaborated.

4.1 Problem formulation

4.1.1 Define the problem

As an e-commerce retailer, it is of general interest to offer a high service level to their customers at the lowest operational cost. This means that the retailers must provide an effective delivery policy but with the lowest cost of resources as possible. Thus, it is desired to find a good trade-off between the resources supplied and the service level offered. For these reasons, the objective of this simulation study becomes as follows:

- Investigate how a change in the replenishment policies and the storage policies can affect the system order throughput time;
- Comparing the performance measurement for the different scenarios;
- Find the best ratio between pods and robots inside a RMFS's warehouse.

In particular, my study is focused on a system with fixed layout and different replenishment policies, such as:

1. **Monoproduct**: each items are storage only in one pods;
2. **Multiproduct**: we can find the items in random position inside the pods, and the

pods have random collocation in the warehouse;

3. **Product organized by classes**: the items are placed in the pods according to a class based policy.

These learning objectives can be related to either the tactical level for replenishment level and replenishment policies and the operational level for the real-time decisions. The defined system will use the FIFO method to decide which order will be served next, however, this can potentially weaken the overall performance since a pod may be used for multiple orders already in the system.

4.1.2 Define the system

In chapter 2, the high complexity and flexibility in the setup of an RMFS were disclosed based upon the literature review. Thus, it is required to define many of the tactical and operational decision problems in the RMFS environment when setting up the simulation model.

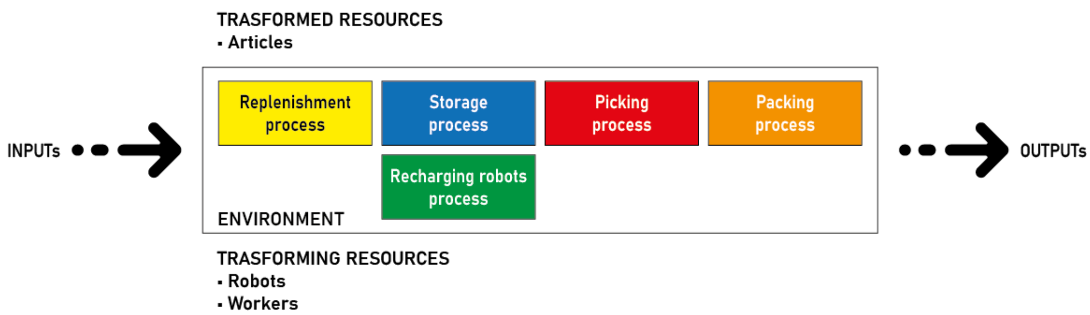


Figure 4.1: System diagram of my model

To create my model it was necessary to simplify the real RMFS, to make the system model easier: the model has now only two inputs, pieces required by the orders and the clients' ID orders, while the unique output is the packed order.

Inside the system there are the replenishment, the picking, the storage and the packing process. To fulfill the orders the core operation is the picking process, where workers standing in their own picking station, collect the correct pieces from the movable shelves carried by AGVs. After this, the pod is carried in the storage area or in the replenishment workstation, according to the space available inside them.

Every two hours of operation the robots need to be recharged by dedicated recharge stations for 5 minutes [15].

4.1.3 Assumption and simplifications in my model

During the work, I had to do some assumptions and simplification of the real RMFS system, in order to be able to replicate it with Anylogic™.
In particular, in my model, I assumed that the system is:

- Stochastic: the parameters, such as the process time in each operation, cannot be entirely predicted, and for this reason it follows stochastic distribution;
- Dynamic: that means that the state variables change over time;
- Discrete and the state variable change at discrete set of points in time.

Regarding the others parameters and decision for my model, in according with Merschformann (2019) [18], I created the same layout reported in fig. 4.2. The figure illustrates the setup of the fulfilment warehouse from a top view. It shows the replenishment stations, with yellow circles, where the workers replenish the pods with new inventory and the picking stations, the red rhombuses, where the workers pick product units to fulfill orders. Among these stations the storage area where the pods are stored in blocks of 2x4 pods can be seen.

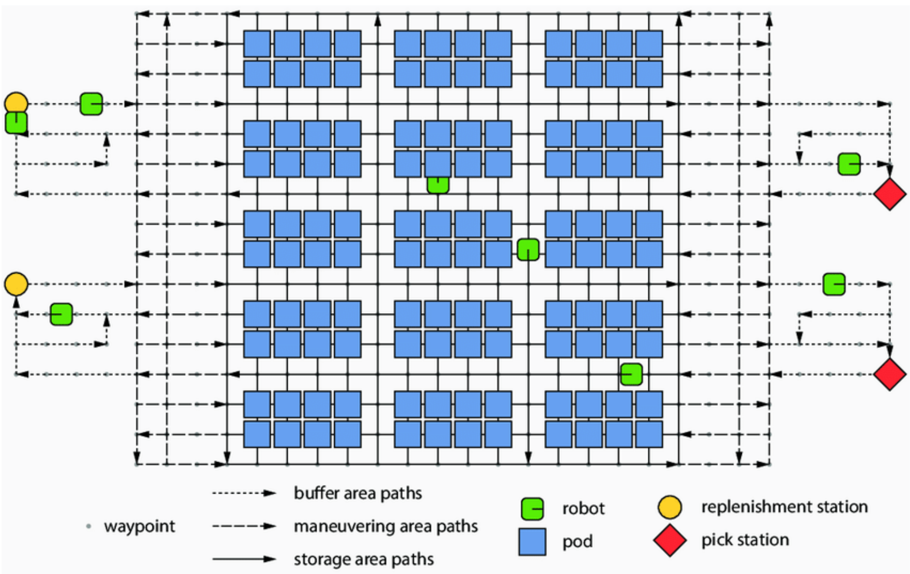


Figure 4.2: A top view of an RMFS layout [18]

One robot carries one pod at a time from the storage area, via the maneuvering area, to the buffer area of the destination workstation. During the picking process the operator picks for multiple unfinished/incomplete pick orders at the same time, following the FIFO policy for orders. For both operations, replenishment and picking, the robots need to stop with a pod at a waypoint, that represent the access point of the workstation. Travel in the aisles is single-directional to avoid gridlock and reduce congestion [18]. In my system, pods have a different storage location according to the storage policies: this meaning that when a pod, after either serving the picker or visiting the replenishment station, returns to storage location, it returns in:

- Generic nearest free location, if the model runs the first or the second scenarios above-mentioned, as in fig. 4.3a;
- Nearest free location in a precise area location decided over the kind of product that it contains if the software simulate the third scenario, as in fig. 4.3b.

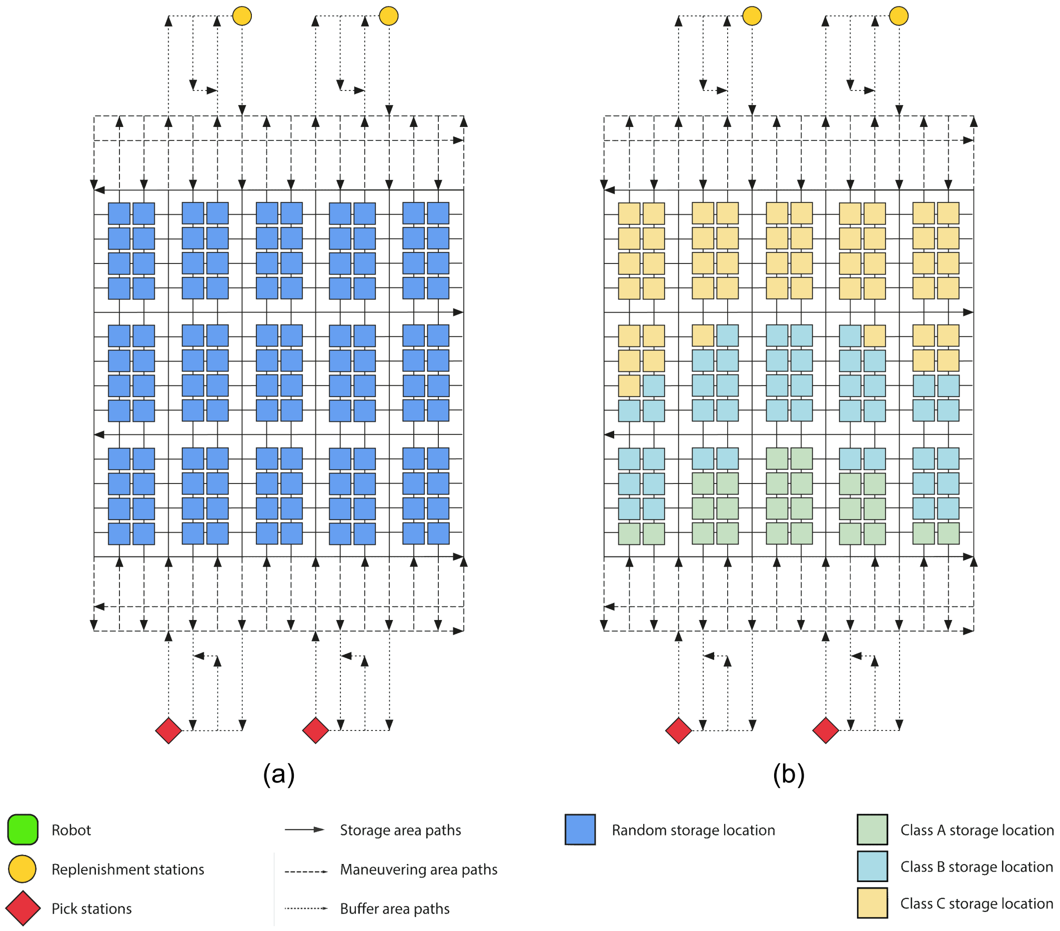
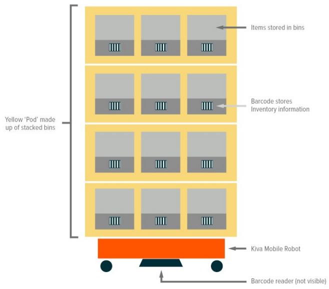
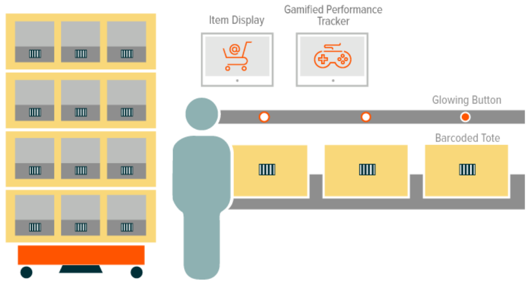


Figure 4.3: 4.3a - Layout with random pod location 4.3b - Layout with ABC storage policy.

For my study I also chose that each pod have 60 slots where it is possible to stock only one item, this means that each pods can can hold a maximum of 60 articles.



(a)

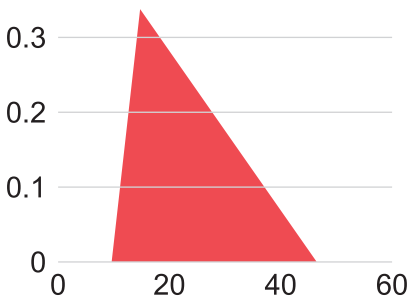


(b)

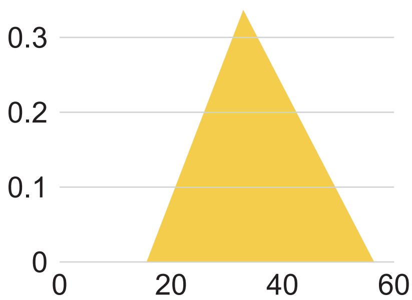
Figure 4.4a - Example of pod in Amazon Inc™ - 4.4b - Layout of a picking station [24]

Regarding the mobile robots, it applies that they carry the pods in the storage area after each picking, unless the pod requires a replenishment operation or it requires to recharge for 5 minutes after two hour's running, [15], operation that is done automatically at its own special charging station.

To estimate the time distribution of picking and replenishment operations I watched a video published by Amazon Tours [25] and I found the value reported in table 4.3 and visualized in fig. 4.5 while for the lifting and storing time I used another video, published by 6ème, in 2017, about Kiva Robots [26]. The final result I decided to use is a triangular distribution for the picking (fig. 4.5a) and replenishment time (fig. 4.5b). This since the triangle is the most common distribution when there are not too many information and data about the true distribution [27].



(a) Picking time distribution



(b) Replenishment time distribution

Figure 4.5: Time distribution in my model

4.1.4 Establish performance metrics

With the model created I will investigate the effect on performance metrics of four aspects:

1. The optimal ratio between AGVs and pods in the warehouse. In particular, my purpose is to analyze the ratio to conduct a more generic study on RMFS, not focused only in my model;
2. Different strategies of allocation of the seventy-four different SKUs on mobile shelves. Each pod can be dedicated to a unique kind of article or have a random mix of that;
3. Different strategies of storage location of pods inside the warehouse: random position or position defined by the SKU's class.
4. The optimal level of replenishment for the shelves, from pods always full to pods completely empty before being filled.

To conduct these analysis, I collect performance metrics, as those shown in table 4.1.

Performance's metrics	**Description**
ρ^{AGV}	AGVs utilization
${}^{pick}{}_{1}$	Average time to pick one order in station 1
$\rho^{pick}{}_{1}$	Picking station 1 utilization
$T^{pick}{}_{2}$	Average time to pick one order in station 2
$\rho^{pick}{}_{2}$	Picking station 2 utilization
$T^{repl}{}_{1}$	Average time to complete the replenishment in station 1
$\rho^{repl}{}_{2}$	Replenishment station 1 utilization
$T^{repl}{}_{2}$	Average time to complete the replenishment in station 2
$\rho^{repl}{}_{2}$	Replenishment station 2 utilization
$T^{order}{}_{wait}$	Average time that the orders wait
$T^{order}{}_{elab}$	Average time to elaborate the orders
$T^{order}{}_{tot}$	Average order throughput time
ω^{pod}	Average distance that each pod travelled
ξ^{pod}	Replenishment level for pods
ε^{repl}	Total visit in replenishment stations
ε^{pick}	Total visit in picking stations
ε^{batt}	Total number of recharging

Table 4.1: Performance's metrics

4.1.5 Build conceptual model

The model created has four main areas, based on the same layout proposed in Merschformann's research [18];:

1. The picking area with 2 picking stations;
2. The replenishment area with 2 replenishment stations;
3. The storage area, which hosts the 120 pods;
4. The charging area, created for the recharging of the AGVs.

My final layout is figured in fig. 4.6. With picking area and replenishment area the queue space are also highlighted: here the pods wait their turn in the workstations.
As mentioned before, When one AGV runs for more than 2 hours, it is automatically redirected in the charging area. The traffic area is necessary to avoid the collisions and to give enough space to AGVs carried by robots.

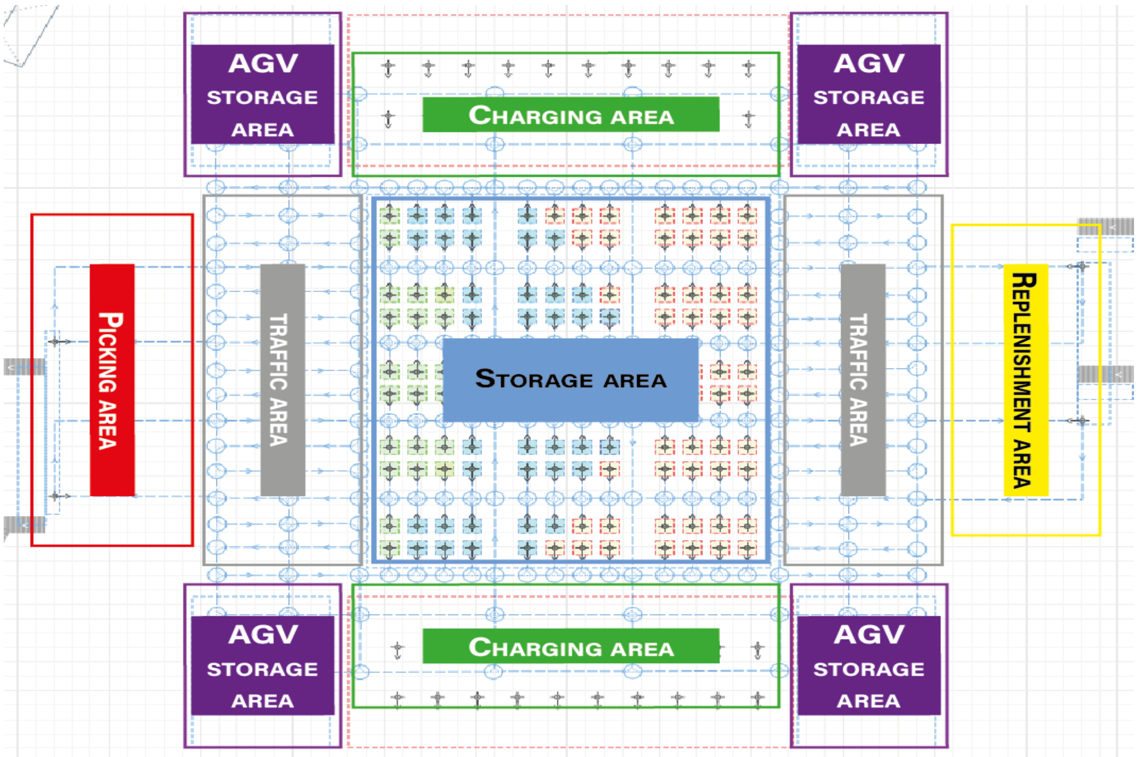


Figure 4.6: Representation of layout in my model, with focus on different areas.

In simulation is also important to define the entities, that Rossetti (2016) defined as objects of interest within a simulation whose activities are being modelled, and the resources, that

are the operating objects which constraint the flow of the entities [20].
My system has such entities:

1. Stocked materials that are modelled as SKUs inside the pods. A full pod contains at maximum 60 SKUs. The items are stored inside the pods by the workers according to the replenishment policy of the scenario and are collected by the picker, sent in the packing station and packed with other articles inside the order;

2. Order list that are the SKUs ordered by the customers. They can have form 1 to 5 lines and each line has only one items.

And it has the following resources:

1. Workers in replenishment stations and in picking stations, called also pickers, stand at their own workstation and wait the pods behind a special gate. This gate open only if there is a pod in front of that, for the safety of human operator. Each operator is able to elaborate only one piece at a time;

2. Robots. They are autonomous robots that can move inside the storage areas or in the charging area and their main role is to carry the mobile shelves between the storage area and the correct right station.

At the end, the interactions between entities and resources are described by processes. We can define them as a sequence of actions that an entity requires to complete an activity. A process is associated with an entity and represents a sequence of states that the entity experiences over a span. A process description describes the general process that an entity experiences as it moves through the system [20].
The model used has 3 main processes:

1. Replenishment. The replenishment of shelves is necessary for the correct functioning of the system. This operation happens if the pod is under the minimum storage level after the operation of picking;

2. Storage: the storage is the task that allows to stock the pods in the specific area of the fulfillment center. The shelves return in the storage area after the picking, or the replenishment operation, in the nearest free location;

3. Picking: the picking from pods allow to get the right articles to fulfill the order. When an order arrives in the system, the software matches it with the pod or the pods containing the required articles. After this, one free AGV carries the mobile shelves in the picking station, and when the operation finishes, the storage process is executed.

4.1.6 Document model assumptions

The last step of the first phase of General simulation methodology, reported in section 3.3, is the document model assumptions.
In particular, to create my model I assumed:

- If the scenario provides that the content of the pod is mixed, then the software create randomly the SKUs inside of that;
- For each SKU removed from the system, new material arrives in the queue of replenishment station and allowing the pod to be refilled;
- The orders are elaborated following a FIFO policy. This means that the model is not able to elaborate priorities;
- One order is still in elaboration until all the lines are picked;
- A new order waits for the completion of the previous one;
- Orders have from 1 to 5 lines;
- A pod has maximum of 60 SKU and a variable minimum, according with the scenario;
- Pod storage location is random;
- The robots can run only along the paths in the storage area, AGV storage area and charging area;
- The robot have to charge after 2 hours of working;
- The system choose the right station where there are less waiting pods;
- Stop for breakdown or failures of machines are not included;
- The SKUs to refill the pods arrive in a common area for both replenishment station;

- Mobile robots are chosen randomly;
- Pods after picking process can return in the storage area or go in the replenish area;
- Pods after replenishment process always return in the storage area;
- When an order arrives, the software matches SKUs and pods that are not involved in others operations;
- The order's packing waits that all the lines are picked;
- The order database is created randomly and the distribution of SKUs in all orders follow an exponential distribution, as reported in appendix A;
- There is no correlation of SKUs' type in different lines of the same order.

4.2 Simulation model building

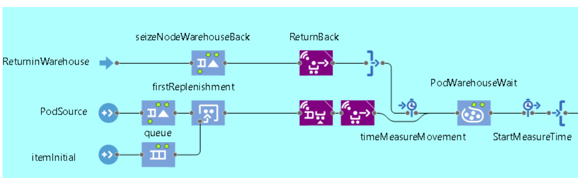
After the first phase, the general simulation methodology provides the translation from the conceptual part into a computer simulation model.

4.2.1 Model translation

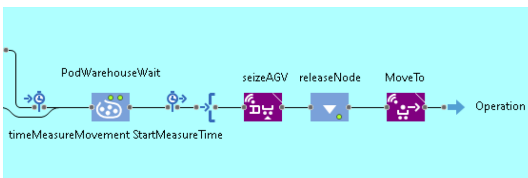
In Anylogic™, the agents of the model are created by source blocks and then flow through the block diagram, that represent a specific role and part of the process. These allow to "transform" the agents and make it possible to do the task they are built to.
For each part of the process presented in section 4.1.5 I grouped the blocks together according to the process they are involved in.

Pods' management process

This process is part of the core of the RMFS simulation model that I created.



(a) Storage process



(b) Move out process

Figure 4.7: Process flowchart of all the pod management process in my model.

In fig. 4.7a there are two *Source* blocks[1]:

- *PodSource*: where the software creates the 120 agents that represent the pods;
- *ItemInitial*: where the different SKUs are created and subsequently stored in pods. This operation is only a formal one, and in particular, is necessary to recreate the three different scenarios with the same block.

After these the pods are lifted and carried in the storage area (*PodWarehouseWait*). This block represent the real storage area mentioned before in section 4.1.5. Here each pod have his node.

The flow that start with block *ReturnInWarehouse* allows the pods to return in the storage area after the picking and replenishment process.

In fig. 4.7b what happens when an order arrive is represented: the pod which contains the largest number of products ordered is taken from the warehouse by a mobile robot and carried in the picking station with the lowest number of pods in the queue.

If there are not mobile shelves that contains all the SKUs in the order's line, then the software chooses the lowest number of pods possible for fulfill it.

To do this, each block contains a lot of java coding lines, as it can be seen in the example reported in listing 4.1.

```
// add lines of order inside in OrderContent (collection)
OrderContent(agent);

// add the same inside a collection that represent the items have to collect
OrderTypeCollection.addAll(agent.OrderContent);

// Function to move the pod with the largest number of products in the
    OrderContent collection (based on priority)
for (Pod p : PodInWarehouse) {
  // for each order try to find one pod with the item inside
  for (int o=0; o < OrderTypeCollection.size(); o++) {
    // if there is -> pod have priority > 1
    if (p.contentItems.contains(OrderTypeCollection.get(o))) {

```

[1] All block, their functionality and examples of coding are described on Anylogic™ website. In particular the blocks that I used are part of *Process modeling library* (in light-blue) and *Material handling library* (in purple) [28]

```
      while (p.contentItems.contains(OrderTypeCollection.get(o)) &&
          OrderTypeCollection.size()>=1) {
        p.priority = 1 + p.ProductsToPick.size();
        p.ProductsToPick.add (OrderTypeCollection.get(o));
        OrderTypeCollection.remove(o);
        break;
      }   // end while loop

    PodWarehouseWait.free(p);
    } // end if (p.contentItems.contains(o)) loop

  } // end for type loop
}
```

Listing 4.1: Example of coding in pod management process.

Picking process

The main process for the fulfillment of the orders is the picking process.

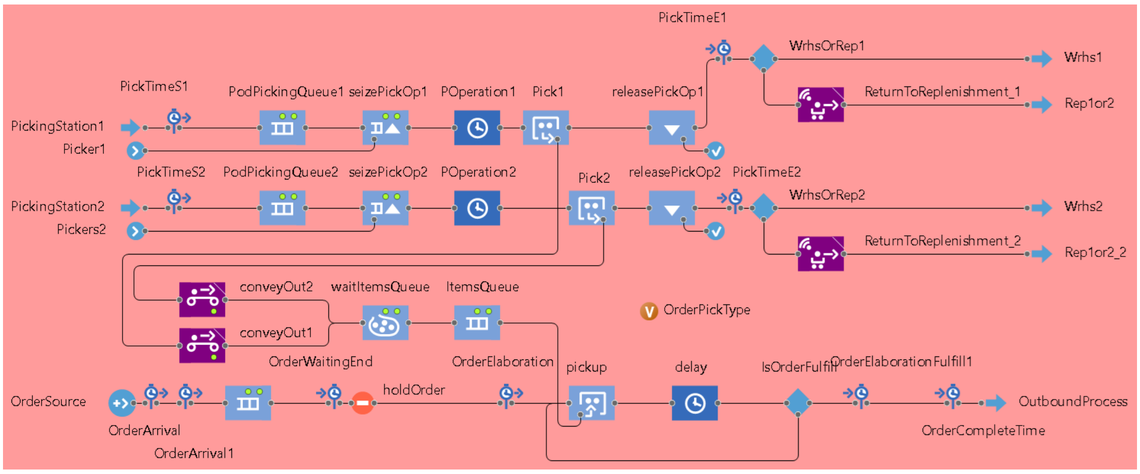


Figure 4.8: Process flowchart of order management and picking process in my model.

The process starts when an order comes and stays in queue until the pod (or pods) arrive in the station and the picker collect the SKUs.

It is possible to describe the agent that represent the order as a truck: the *pickup* block, permits to load on it the different SKUs before it leaves the system. In particular, the block called *IsOrderFulfill* creates a loop that finish when all the order's items are picked.

The orders are generated at precise time and the information that they have are:

- Time: when the order arrives in the system. The database follows an exponential distribution, with more orders in special time slots, such as evening and lunch time;
- ID order: a unique identification number for each order;
- Number of lines, that represent the number of articles in the order;
- SKUs: the kind of articles that were ordered.

Compared to the complexity of what described above, the real picking phase is simpler. The blocks *PickingStation1* and *PickingStation2* permit the pods to enter in the picking area, where they find *PodPickingQueue* which represents a real queue for exceeding mobile shelves that have not enough space to be carried in front of the gates (*Pick1* and *Pick2*) where the picker collect the SKUs.
The picker's work is modelled with two blocks, delay and drop-off: the first one is necessary to reproduce the operation time, while the second one removes the items useful to fulfill the order from pods. I used two blocks to model that process because the drop-off block do the operation instantaneously.
When all the articles are collected, the AGV moves the mobile shelves out of the station and the pods have two different possibilities: return directly to the storage area or go in the replenishment area. The decision is based on their replenishment level according to the minimum replenishment level.

Replenishment process

For each SKU removed from the system it is necessary to refill it and this operation is done in the replenishment process, reported in fig. 4.9.

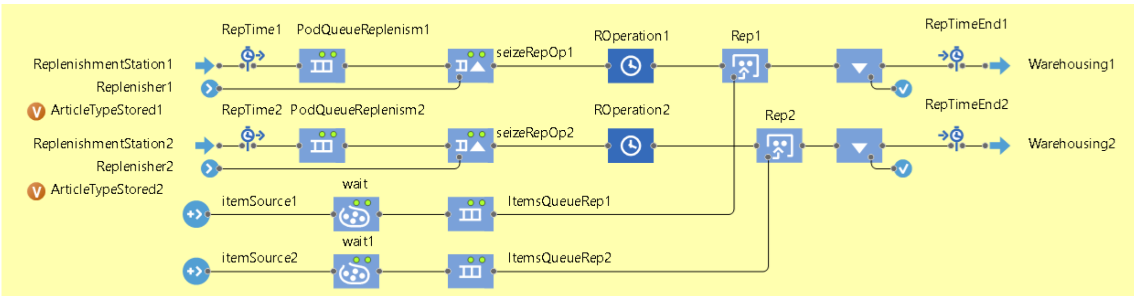


Figure 4.9: Process flowchart of replenishment process in my model.

In fig. 4.9 the first two flowcharts (in the uppert part) show pods that follow a flow very similar to picking process. When mobile shelves arrive in the replenishment area, (see section 2.2.1) if there is another one in front of the worker, they wait in the queue (*PodQueueReplenishment*) and after that they are carried in the replenishment station.
The main differences with the picking process are:

- Operational time: the replenishment is a bit slower, in fact in real Amazon warehouse, the items are stocked with a random storing policy, that means each SKU does not have a precise storage location, but the articles are closets where there is free space [25];
- Each operator puts articles on the shelves, while with the picking they collect SKUs from it.

The SKUs arrive in the station with conveyors that are able to carry the correct code according with pod's station destination. This part of the process change for each scenarios:

1. Scenario 1: the SKUs arrive in the station randomly disregarding the contents of the pod,
2. Scenario 2 and scenario 3: the SKUs arrive in the station according to the contents of the pod.

Important is to remember that each code that arrives in the stations is previously removed by the pod, with a picking process. When the refill process is done, the pod returns in the storage area following the storage policy.

Charging Process

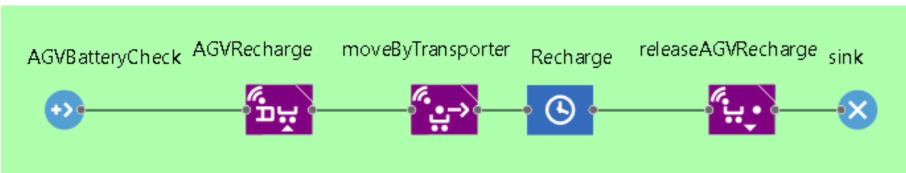


Figure 4.10: Process flowchart of recharging process in my model.

When a mobile robot runs for two hours it has to charge for 5 minutes [15], and this condition was recreate with the flow in fig. 4.10.
In particular, I created this because the library that I used to control the mobile robots in each process does not concern functions such as interruptions, maintenance or fail-

ures [28]. To solve this inconvenience, with special java-coding, I created a function that moves the robot in this part of the flow and, in particular, in charging station when the running maximum time is reached. Through this sequence the software makes the robot unavailable for charging time, in other words it is as if the robot was busy doing another operation.

Arrival of items and packing process

The last two processes that I create are the arrival of SKUs in the warehouse and the packing process. In fig. 4.11 the SKUs that are necessary for the replenishment process

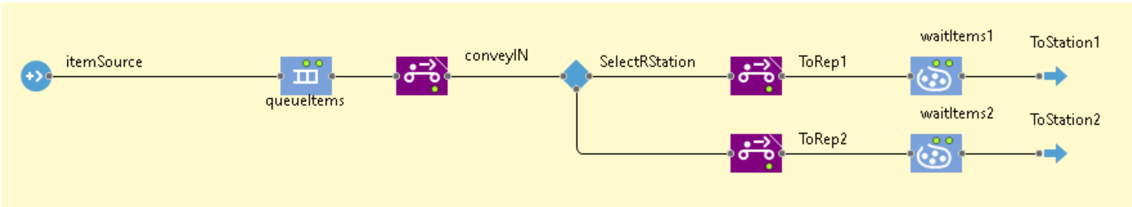


Figure 4.11: Process flowchart of SKus' arrival in my model

and mentioned in fig. 4.9 are created according to the scenario that is running. After the first accumulation area the SKUs are moving by the conveyors and, thanks to *SelectRStation*, they arrive near the operator.

The second process, the packing, is very easy. In fact, the orders leave the *order process management* only when all the lines are picked. Thanks to this the packing process is modelled with different delay blocks: *BoxSelection, Canning, PackingClosing* and *Labelling*.

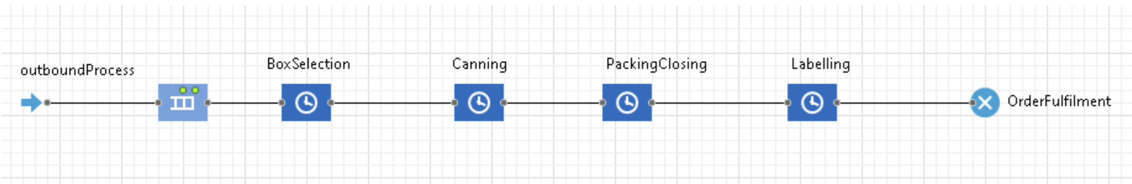


Figure 4.12: Process flowchart of packing operation in my model

These model includes more than 32000 coding lines, in part coded directly, in part coded by the software with the blocks. All those lines are aimed at reproducing the complexity of the activities that I described in the previous pages.

4.2.2 Input data modelling

The model I built has 3 types of data in input:

- Fixed parameters that are the general parameters that never change during the simulation, and are certain data collect by papers, books, catalogues or establish by me with some assumptions;
- Stochastic parameters: some data aren't certain, or could be variable such as the human operations. For that reason I model it such as statistical distribution, and their value will change continuously according to the chosen distribution (section 4.1.3);
- Variable parameters. These are the parameters that have effects on performance of the system and it will be investigated during the simulation to find the optimal result.

All before mentioned parameters are synthesized in the following tables.

Parameter	Value	Unit
Simulated duration	7	days
Number of replications	15[2]	
Size of order backload	1400	orders
N° of aisles	6	
N° of cross-aisles	4	
Storage locations	120	
N° of pods	120	
Layout	2x5 blocks	
N° of unit replenishment order	from 1 to 60	items
Space in each pod	60	articles
Picking stations	2	
Max pieces size	29 x 50 x 15	cm
Pod size	1 x 1 x 1.8	m
Robot size	25 x 60 x 10	cm
Robot acceleration	0.8	m/s^2
Robot deceleration	0.3	m/s^2

[2]The calculations are reported in Appendix B.1.1

Parameter	Value	Unit
Robot maximum velocity	1.5	m/s
Minimun distance between robots	1.5	m
Time between robot two charging period	2	hours
Time for recharging	5	minutes
Time for packing	120	seconds
Collision timeout collision	1	seconds
Class A storage locations	24	
Class B storage locations	44	
Class C storage locations	52	
Number of possible different SKUs	74	

Table 4.2: Fixed parameters' summary

Parameter	Value	Unit
Time for lifting and storing a pod	normal(0.8,15)	seconds
Time for picking a component from pods	triangular(10,46,15)	seconds
Time for putting a component in pods	triangular(16,54,33)	seconds
Initial inventory level	triangular(60, 80, 100)	percentage

Table 4.3: Stochastic parameters' summary

Parameter	Value	Step
n° of robots	from 5 to 35	1
Level of replenishment	from 0 to 1	0.1

Table 4.4: Variable parameters' summary

5 Verification and validation

In previous section 3.3 I mentioned the validation and the verification of the model that are necessary before the experimental design can be initialed. It is most importance to determine whether the developed model accuracy represent the actual system.

The first step, in accordance to the general simulation methodology [20], is to verify whether the model performs as intended. Rossetti emphasises the importance of performing model debugging, whereas Law introduced eight different techniques which supplement by each other forms the verification of a simulation model [29].

5.1 Literature review

Such as explained in the previous chapters the RMFS are not well investigated yet, in particular, the first researchers started to publish papers around 2010-2012.

Yuan et al. [30] conduct some researches about the comparison with a class based policy where products are stored in the same pod if it have a strong correlation, and they validate their model with an ABC class in a normal warehouse with a dynamic simulation.

Lamballais and De Koster [31] tried to find an optimization with three decisions variables: (i) the number of pods per SKU; (ii) the ratio of the number of pick stations to replenishment stations; and (iii) the replenishment level per pod. Their results show that throughput performance improves substantially when inventory is spread across multiple pods, when an optimum ratio between the number of pick stations to replenishment stations is achieved and when a pod is replenished before it is completely empty.

Gong et al. [32] created a simulative model based on customers classes (such as premium and normal customers) and their purchased. They used a Markov models to study this.

Xie et al. [17] conduct studyes on slit orders and optimizing operation with a special focus on the assignment of pods and orders to stations with a MIP-model (Mixed-Integer Programming model). Merschformann, Lamballais and de Koster [18] simulate both pick and replenishment process, study the order assignment, pod selection and pod storage assignment with multiple decision rules per problem. They use also a popularity of the SKUs with an exponential distribution with parameter λ=1/2 to emulate a ABC curve.

5.2 Verification of my model

For the verification part I followed the Law suggestions [29] and in particular I used the technique 1 to 4.

5.2.1 Technique 1

The first technique recommends starting with a moderately detailed model and then gradually increase the complexity, instead of developing an immediately complex model, which could lead to a tedious and extremely difficult process of determining the location of potential error in the program. Thus, the model was initially developed for each process and using randoms values and parameters. This mean that only part have in a first version the very basically functionalities: the robots carried a random pod without correlation with his contents, the picking was developed such as a picking of the same number of SKU that are ordered (and not kind of articles ordered), etc.. Gradually, when the process is working I start to code more, and add the functionalities and that means a continuous process of implementation with subprograms and gradually enhance of level of detail.

5.2.2 Technique 2

The second technique advises having more than one person reviewing the computer program, preventing the writer of a particular subprogram from getting stuck when debugging or wrongly being convinced that a statement is correct. The flowchart of each process was deeply discuss with my co-supervisor prof. Allan Larsen and more than one coded function was discuss with the Anylogic Support. In particular, that last one, is very important because Anylogic provide a very high level of coding, more than that I was able to learn in these months at DTU.

5.2.3 Technique 3

This technique states that the simulation model should be tested under a variety of input parameters (random parameters create by the software) and then ensure that the different outputs are reasonable and behave as expected. The model has, therefore, been run with various resource settings seeking to both stressing the system but also ensuring that an increase of resources resulted in an improved model output. The system was furthermore stress-tested by increasing the demanded orders. The demand has been increased until more 3000 SKUs in a day. These is more or less the same amount elaborated in one week in my scenarios and it guaranteed to test the system to it limits, and thus, ensure that it will not reach a deadlock. The demand has been increased by a reasonable percentage, where the output acted as intended.

5.2.4 Technique 4

It is here dealt with the problematics of debugging a DES program, and Law [29] states that one of the most powerful techniques for this is the so-called *trace*. By incorporating a trace the programmer is meant to display the state of the system after each event has occurred, and the exact content to display can vary from case to case. The general purpose for this is to enable the programmer to compare the state variables or certain statistical counter with hand calculations to see if the system behaves as intended. This has been implemented for each subprogram, so the changes in the system's state for the different calls of event could be controlled and verified. This technique has, amongst other things, resulted in the clarifications of the need of statement four in the process of determining, which event to be executed next. By displaying the state of the system after the execution of each event, it was made clear that the system could reach a deadlock if an order was being served by a picker and the next pod, which the order needed before becoming completed, was busy serving another order. This has been circumvented by incorporating that an order, which is initiated and currently handled by a picker, always can take over a pod serving another order not being handled by a picker yet. Even though this technique has its clear advantages for debugging codes, it also has some shortcomings. It is not always clear which information to display to highlight the error when the simulation is run, and some particular errors might not occur in the short simulation run. This can entail that the simulation has to be run multiple times and therefore, can this technique become quite time-consuming.

5.3 Validation of my model

The second step of determining the accuracy of the developed simulation model is to validate whether the model representation of the real system is accurate or not.

This part is usually done by presenting the model and its results to the persons involved with the system being studied, where a critical review of the model should take place and model outputs should be compared with outputs of the real system. Since this has not been an opportunity, the model's output has been collated with information about system performance gathered from different sites. When it is not possible to include different subject-matter experts or the personnel involved with the system, it is important to use one's intuition and experience to critically review the model output.
Therefore, it is relevant to mention the cooperation with prof. Allan Larsen and his critical reviews and intakes on the model. This is a part of the results validation, which also can be complemented by the use of animation. When simulating in general-purpose programming language this is not as easily accessible as if e.g. simulating in AnyLogic where this is an integrated part of the system. To see how the state of the system changed throughout the simulation, different statements were therefore displayed, and the text output helped to a "visual" understanding of what exactly happens when the different events are called.

6 Results and my own analysis

In this chapter I will present my own experiments with the model and results. After the creation of the model described in chapter 4, I can finally conduct experiments and bring some conclusions.

The final input I used during the scenarios can be found in section 6.1 . In section 6.1.1 I investigate the problems of the initial settings of my model while in the last section final results of the simulation are reported.

First of all, I want to resume what has been described in section 4.1.5 and what I decided to investigate, focusing in particular on two aspects:

- Different strategies of allocation of SKU on pods:
  - Multi SKUs on each pods, with a random storage;
  - Same SKU on pod;
- Different strategies of allocation of pod in the storage area:
  - Pods are stored without a fixed location in warehouse;
  - Storage is made according to the commodity class contained in the pod.

These two aspects have allowed me to create three different scenarios:

1. Scenario 1: Pods are stored with a random position in warehouse, and each pod is refilled with multi SKUs;
2. Scenario 2: Pods have a random location in the storage area, but each pod has only one SKU,
3. Scenario 3: Pods contains only one SKU and have a specific location according with the commodity class.

6.1 Final inputs

The orders represent the main input of an e-commerce system. In particular, for my analysis I used a order database created following the process reported in appendix A.1, the result of which, is in fig. 6.1

orders

	date	ordernumber	items_order	item1	item2	item3	item4	item5
1	13-06-2022 00:08:00	1	5	BlackPen	GoPro	Eraser	CDs	Book1
2	13-06-2022 00:14:00	2	4	BackPack	Book4	Mask	Curtain	
3	13-06-2022 00:17:00	3	2	Tent	Scissor	Nada		
4	13-06-2022 00:22:00	4	4	Clock	Pot	Shoes	Rug	
5	13-06-2022 00:25:00	5	5	USBPen	Towel	Cup	Tshirt	Mask
6	13-06-2022 00:27:00	6	2	Bag	Console			
7	13-06-2022 00:30:00	7	4	Curtain	Scissor	Pot	MP3	
8	13-06-2022 00:33:00	8	5	Baggage	Mask	Cup	Telecamera	ScreenProtector
9	13-06-2022 00:39:00	9	2	FinsSlipper	GoPro			
10	13-06-2022 00:42:00	10	4	Speaker	CDs	Tool	Camera	
11	13-06-2022 00:45:00	11	4	Flags	Eraser	BackPack	TableGame	
12	13-06-2022 00:47:00	12	5	Console	Console	Book3	Soap	Mask

Figure 6.1: First orders in my model

The orders' timing follows an exponential distribution during days with λ=7.2 that are able to satisfy the 500 orders/day $(appendix\ A.1)$

6.1.1 Problems and adjustments

One of the main problems of the simulation model is the congestion in the first model developed. It is possible to define congestion of robots as the crowding of robots in an area of the layout. When it happens the robots do not continue their travel and the system results blocked. This phenomenon terminates the simulation and the results are incorrect. To resolve that, I mention I worked hard with coding, paths and blocks in Anylogic: in particular, from the first layout with only one AGV storage area I created 4 different areas which recreate the same function (fig. 4.6): I used the "restricted area" block, to allow a limited number of AGVs to travel certain paths, I decrease the minimum distances between robots to 1m and 20 cm, when the mobile robots carry the pods and I increased the frequency with which the software calculates possible collisions. The real algorithm is able to avoid this kind of problems before they occurs, but it is more complex and complicated than mine. My solution, maybe, is far form that, but it's working for most of my experiments.

In the reality, if a congestion happens, the intervention of a specialized technician and a short system's breakdown will be necessary to solve it.

Other observed problems are related to the order database but it has been solved before the latest analysis. The first problem occurs when there are two or more orders at the same time, the second when there are orders as in fig. 6.2.

13/06/2022	4	3	Clock	Pot	Console		
13/06/2022	5	5	USBPen	Towel	Console	Tshirt	Console
13/06/2022	6	3	Bag	Console	Camera		

Figure 6.2: A wrong schedule orders database.

The problem of orders which have the same time is related to the software: for Anylogic it is impossible to create 2 agents (in this case, the orders) at the same moment. I solved this issue by editing by hand the time database.

The second problem detected, on the contrary, is generated when the same item is called twice but not in consequential order (referring to the figure, the "Console" items is called, then the "t-shirt" item is called and then again the "Console" has to be picked). This creates a problem in the algorithm since the code is not able to detect that the item is called twice in a non consequent line. Once the pod that contains the "console" has been assigned to the picking station, the software search the "t-shirt" to do the same operation but it is not able to update the number of "console" that have to be picked, since in the order the two elements are interrupted by the "t-shirt" item and the PPS for the first "console" has already been launched. To solve this I checked all the orders lines and run the model multiple times to find the wrong sequence or the picking stops. In these case the SKUs should be near and not with other SKUs in the middle.

6.2 Scenario analysis

The first analysis of the scenarios compares the AGV's utilization with the total time to complete the orders. This kind of analysis is the perfect compromise between the needs of the company and the desire of customers. In fact, the enterprises want to maximize the utilization of mobile robots in order to reduce cost and increase the income with more order's fulfill, while the customers want a short lead time between the order and his delivery.

Other analyses involve the number of times pods are transported to workstations and how

long they travel around the warehouse: this data allow to understand how to minimize not only the number of times the pod is transported and the time it is one the move, but it is also fundamental to reduce the charging time of mobile robots.
All simulations were successful in completing orders, so we do not analyze the total number of orders fulfilled during the period analyzed, but only the average time to complete each order.
In the end, all of these analysis were done for each replenishment level.

6.2.1 Analysis methodology

In order to do a concrete and most accurate analysis I tried to define a ”scientific” methodology to find the best solution, and for this reason, I searched some parameters that are useful to conduct this study. I choose the following parameters:

- **Cost of one mobile robot**. When a company implements a system like the RMFS or, more simply, a system with AGVs, forklift, etc. to carry the component or pickers around the warehouse, it has to do a very huge investment. In particular, to create a warehouse with mobile robots, Kiva System could cost around [15]

  - from $1 to $2 million for a ”startup kit” of robots,
  - from $2 to $4 million for a typical warehouse setup with 50-100 robots,
  - From $15 to $ 20 million for a large warehouse operation with 500-1000 robots.

  This kind of investment considers a complete set of mobile robots including the management’s software, so I searched also other information in order to define the cost of each mobile robot that is a variable cost. Following the online articles, paper and catalogues, the estimation of cost for one AGV can be around $27.000 [33].
  In my model, for the analysis I considered only the variable cost which follow:

$$C_{\text{AGV}} = \$27.000 \cdot n^{\text{robots}} \tag{6.1}$$

- **Cost of non-completion of the order**: Data show that the average value of each order executed on Amazon is around $78 (Quora, 2018). I consider that, in one day, theoretically, it is possible to fulfill a number of order equal to:

$$O_{\text{day}}{}^{\text{T}} = \frac{24 \cdot 60}{min(T^{\text{best}}{}_{\text{fulf}})} \tag{6.2}$$

Where min($T^{\text{best}}{}_{\text{fulf}}$) is the minimum average time to fulfill one order across all configurations. For each combination, instead, I calculate the orders that can actually be satisfied as:

$$O_{\text{day}}{}^{\text{R}} = \frac{24 \cdot 60}{T_{\text{fulf}}} \tag{6.3}$$

where T_{fulf} is the average time to complete the orders with the combination under analysis. Using the eq. (6.2) and the eq. (6.3) it is possible to find the cost that arises, understood as lost income, as:

$$C_{\text{o}} = \left\{ O_{\text{day}}{}^{\text{T}} - O_{\text{day}}{}^{\text{R}} \right\} \cdot \$78 = \left\{ \frac{24 \cdot 60}{min(T^{\text{best}}{}_{\text{fulf}})} - \frac{24 \cdot 60}{T_{\text{fulf}}} \right\} \cdot \$78 \tag{6.4}$$

- **Cost of battery recharging and AGV maintenance.** The first cost, battery recharging, can be calculated using the simple physics formulas in a wireless charging process. The AGV that I before mentioned has a battery of 40Ah and 48V (DC). Knowing these parameters, the cost for the recharging is:

$$_{\text{B}} = p \cdot \eta \cdot \frac{dollars}{kWh} = \frac{(48 \cdot 40)}{1000} \cdot 1.15 \cdot 0,637 \cdot \frac{5}{60} \cdot e^{\text{batt}} = 1.172\$ \cdot e^{\text{batt}} \tag{6.5}$$

  Where η is the normal efficiency during a wireless recharging process, the unit cost of energy is 0.637 $ for each kWh (PUN - Prezzo Unico Nazionale (Italy) - August 2022) and $\varepsilon^{\text{batt}}$ is the total number of recharges.
  In one year the eq. (6.5) becomes:

$$C_{\text{B}}{}^{\text{year}} = \frac{C_{\text{B}} \cdot 365}{2} \tag{6.6}$$

  I divided all by two considering that the available metrics refer to two days.
  Assuming there is a preventive maintenance policy, for the maintenance the cost is

around $ 9500 for each mobile robot [33].

$$C_{\text{M}} = 9500 \cdot n^{\text{robots}} \cdot years \tag{6.7}$$

Considered all these costs, and the usual 5 years guarantee for mobile robots and 500 orders/day for 5 years also, the final formula to find the best trade-off is:

$$min\left\{C_{\text{AGV}} + C_{\text{O}} + C_{\text{B}} + C_{\text{M}}\right\} \tag{6.8}$$

And substituting the numerical values, the eq. (6.8) becomes:

$$min\left\{n^{\text{robots}} \cdot (27.000 + 9500 \cdot 5) + \left\{\frac{24 \cdot 60}{min(T^{\text{best}}{}_{\text{fulf}})} - \frac{24 \cdot 60}{T_{\text{fulf}}}\right\} \cdot \$7117500] + [213.89 \cdot e^{\text{batt}}]\right\} \tag{6.9}$$

6.3 Scenario 1

This scenario is the one Amazon.com™ and other big e-commerce companies claim to use. In particular, they call it *"Random Stow"* and is the scenario where the storage decision is based on the free space within the mobile shelf [25]. Starting from the same assumption, I began to analyze the system replicating these conditions.

In particular, the first brief analysis of the data shows that the maximum utilization of AGVs occurs when 5 mobile robots are in the system and when the minimum level required before the replenishment is 20%. That result is not enough: following the studies reported in section 2.4 it is important to investigate *throughput, lead time, quality, human factors, flexibility, efficiency and cost*.

As for quality, taking into account how the system is constructed, I assumed it is always 100%, which means it is not possible for the operator to pick up a wrong item, since the human factor usually depends on the workstation's layout who is fixed and for that reason is not investigate in this study.

The result reported in table 6.1 shows also that the average utilization of workers is low: this depends on the number of mobile robots that carry the pods, given that while they move the pods from one point to another inside the storage area, the picker does not work. But if there was a lower number of robots, then it would not be possible to carry

Performance's metrics	Description	Average Value
ρ^{AGV}	AGVs utilization	89.97 %
ρ^{pick}	Picking stations utilization	26,60 %
ρ^{repl}	Replenishment stations utilization	16.48 %
$T^{order}{}_{wait}$	Time that the orders wait	1.084
$T^{order}{}_{elab}$	Time to elaborate the orders	3.864
$T^{order}{}_{tot}$	Order throughput time	4.948
ε^{repl}	Total visit in replenishment stations	2445
ε^{pick}	Total visit in picking stations	3580
ε^{batt}	Total number of recharging	45

Table 6.1: Model output with 5 mobile robots and a minimum replenishment level when the pod are at 20% in scenario 1.

other pods in the workstations. When looking at the AGV's average utilization, it can be seen that it is lower than 100%: this is caused mainly by the fact that the value returned by the software takes into consideration also the moments when there are not orders to fulfil and considers the mobile robot as free also during the battery's recharging timing.
Taking in consideration what I mentioned in section 2.4 and section 6.2.1 I tried to transform all the parameters in a cost.

Using the eq. (6.9) the solution with 5 mobile robots and a minimum refill level as 20%, after 5 years, have a total cost of more than 47 millions of dollars.

The best solution, found with the eq. (6.9), instead, allows to fulfill one order in 2,302 minutes, and cost 112.2 % less compared to the first solution reported. The combination necessary to reach this result is to have 10 mobile robots and a replenishment level equal to 50%. Data of table 6.2 shows lower average throughput time to order fulfill, a better utilization of workers and less number or recharges.

Comparing this two outputs, it is possible to see also that the pods come in the picking station more times when there are 10 mobile robots: that depends on the stow inside the mobile shelves. In fact, as per assumption in section 4.1.6 the pods contain random SKUs, and that influences the result.

Performance's metrics	Description	Average Value
ρ^{AGV}	AGVs utilization	81.15 %
ρ^{pick}	Picking stations utilization	68.17 %
ρ^{repl}	Replenishment stations utilization	38.54 %
$T^{order}{}_{wait}$	Time that the orders wait	0.784
$T^{order}{}_{elab}$	Time to elaborate the orders	2.518
$T^{order}{}_{tot}$	Order throughput time	2.302
ε^{repl}	Total visit in replenishment stations	1826
ε^{pick}	Total visit in picking stations	3408
ε^{batt}	Total number of recharging	75

Table 6.2: Model output with 10 mobile robots and a minimum replenishment level when the pod are at 50% in scenario 1.

Despite the less mobile robots' utilization, the second solution allows to elaborate the 44.14% more orders, and that has a strong impact on the possibles incomes of the company.

Performance's metrics	5 AGVs	10 AGVs	Variation	
ρ^{AGV}	89.97 %	81.15 %	-6.29 %	↓
ρ^{pick}	26.60 %	68.17 %	+100.74 %	↑
ρ^{repl}	16.48 %	38.54 %	+85.46 %	↑
$T^{order}{}_{wait}$	2.084	0.784	-63,69 %	↓
$T^{order}{}_{elab}$	2.864	1.518	-22.62 %	↓
$T^{order}{}_{tot}$	4.948	2.302	-44.14 %	↓
ε^{repl}	2445	1826	-25.32 %	↓
ε^{pick}	3580	3408	-4.81 %	↓
ε^{batt}	45	75	+66,66 %	↑

Table 6.3: Model output comparison

Analyzing with more detail the data in table 6.4 we see that the most strong impact is the orders' value. In fact, the lower time the order waits before his elaboration, and the lower time to pick all the lines, are translate with more orders that can be elaborate during the day, and these, replicated for 5 years have a great impact on the final result. The utilization level reported in table 6.3 permits to have a margin for the maintenance of the robots without a strong impact on performances. Looking to other data that the software returns, with the same replenishment level and 9 AGVs (for example, if we have one mobile robots in maintenance) the utilization grows up to 82.92% while the order throughput still remains around 2 minutes and 30 seconds (2.469 minutes) allowing to elaborate 7% order less.

Costs	5 AGV		10 AGV		Varation	
AGV cost	135'000	$	270'000	$	135'000	$
Maintenance	237'500	$	475'000	$	237'500	$
Charging	38'500	$	64'167	$	25'667	$
Orders' Value	41'431'000	$	88'978'000	$	47'547'000	$
Total	41'841'500	$	89'787'167	$	47'945'667	$

Table 6.4: Projection of costs during 5 years

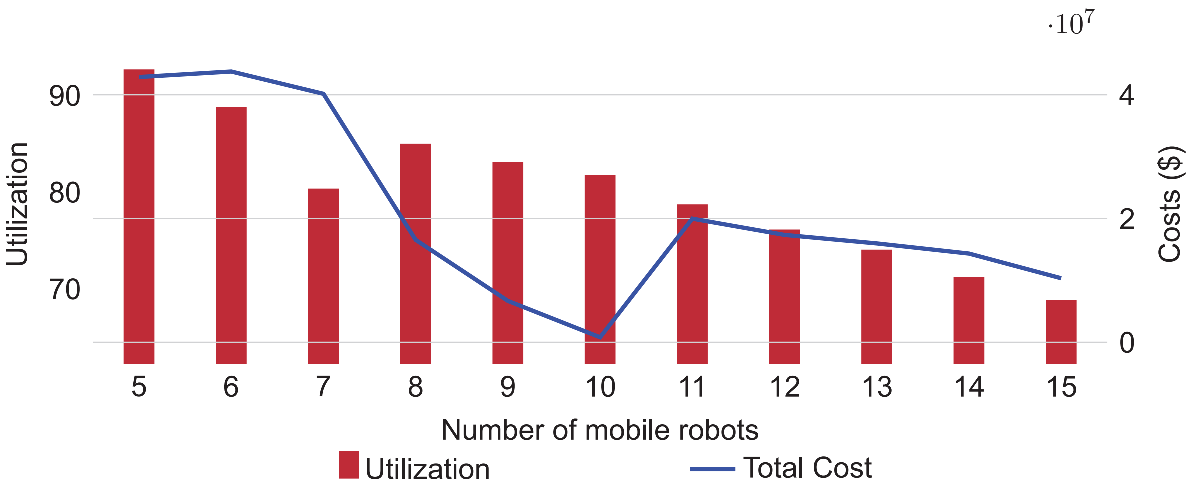


Figure 6.3: Cost and utilization graph with a replenishment when the pods contains less than 50% of their maximum capacity.

With a 81.15% of mobile robot's utilization and considering that the workers are not completely exploited, it is also possible to increase a bit the number of order during the days from the 500 order/day considered in the analysis that I realized.

6.4 Scenario 2

The second scenario plans to leave the policy of returning pods to the storage area unchanged, but it changes what each mobile shelves contains. From the previous analysis is possible to see that all the different combinations of mobile robots with the different replenishment level have a same huge number of transportation in the picking stations. This is the most intuitive result and the outputs confirm this. In fact, for each different SKUs in order's line the mobile robots have to carry a different pod in the stations in order to fulfill them.

Another easy consideration is related to the visits in the replenishment station, that is lower than in the first scenario. This result depends from the content of the pods, as it is less frequent that their supply level drops under the minimum replenishment level if they contains only one kind of articles.

Conducting analysis's steps similar to the previous ones, the combination with the higher average AGV utilization is the one with 5 mobile robots and the 50% as the limit before the need for pod refill.

Performance's metrics	Description	Average Value
ρ^{AGV}	AGVs utilization	98.20 %
ρ^{pick}	Picking stations utilization	39.16 %
ρ^{repl}	Replenishment stations utilization	25.48 %
$T^{order}{}_{wait}$	Time that the orders wait	2.406
$T^{order}{}_{elab}$	Time to elaborate the orders	3.985
$T^{order}{}_{tot}$	Order throughput time	5.391
ε^{repl}	Total visit in replenishment stations	1222
ε^{pick}	Total visit in picking stations	7330
ε^{batt}	Total number of recharging	93

Table 6.5: Model output with 5 mobile robots and a minimum replenishment level when the pod are at 50% in scenario 2.

As reported in table 6.5 it is possible to see that the AGV's utilization is very high: excessive to be sustainable (considering that they need some maintenance, there are possibilities of breakdowns, etc.) while the pickers and the other workers have a low utilization. The average order throughput is higher than the one in scenario 1 and this has an important consequence on the final performance of the system and also doubles the number of required recharges.

Performance's metrics	**Mix content**	**Same Content**	**Variation**	
ρ^{AGV}	89.97 %	96.84 %	+7,09 %	↑
ρ^{pick}	26.60 %	39.38 %	+48.05 %	↑
ρ^{repl}	16.48 %	31.50 %	+91.14 %	↑
$T^{order}{}_{wait}$	2.084	2.317	+11.18 %	↑
$T^{order}{}_{elab}$	2.864	3.314	+15.71 %	↑
$T^{order}{}_{tot}$	4.948	5.631	+13.80 %	↑
ε^{repl}	2445	1955	+20.05 %	↑
ε^{pick}	3580	7330	+104.75 %	↑
ε^{batt}	45	91	+102 %	↑

Table 6.6: Model output comparison with 5 AGV e replenishment level equal to 20%.

Performance's metrics	**Mix content**	**Same Content**	**Variation**	
ρ^{AGV}	86.57 %	98.20 %	+11.84 %	↑
ρ^{pick}	29.36 %	45.16 %	+34.98 %	↑
ρ^{repl}	15.68 %	25.48 %	+38.46 %	↑
$T^{order}{}_{wait}$	1.960	2.406	+22.75 %	↑
$T^{order}{}_{elab}$	2.861	3.985	+15.65 %	↑
$T^{order}{}_{tot}$	4.821	5.3991	+10.70 %	↑
ε^{repl}	1521	1222	-24.47 %	↓
ε^{pick}	3041	7330	+58.51 %	↑
ε^{batt}	42	93	+54.83 %	↑

Table 6.7: Model output comparison with 5 AGV e replenishment level equal to 50 percent.

table 6.6 and table 6.7 report the comparison between the scenario with 5 AGVs and refill level when the pod contains less than 20% of his maximum capacity and the hypotesys with 5 AGVs and refill level when the pod contains less than 50% of his maximum capacity. For each comparison, their percentage variation is also reported . The growth of all the parameters of utilization, time order throughput and the increment of number of charging reduce the number of orders that the system can elaborate every day with a strong impact on possible company's incomes.

Costs	10 AGV - Scenario 1		13 AGV - Scenario 2		Variation	
AGV cost	270'000	$	351'000	$	81'000	$
Maintenance	475'000	$	617'500	$	142'500	$
Charging	64'167	$	72'723	$	8'106	$
Orders' Value	88'978'000	$	63'210'498	$	25'767'502	$
Total	89'787'167	$	64'251'721	$	25'535'446	$

Table 6.8: Comparison of projection of costs during 5 years between the two scenarios

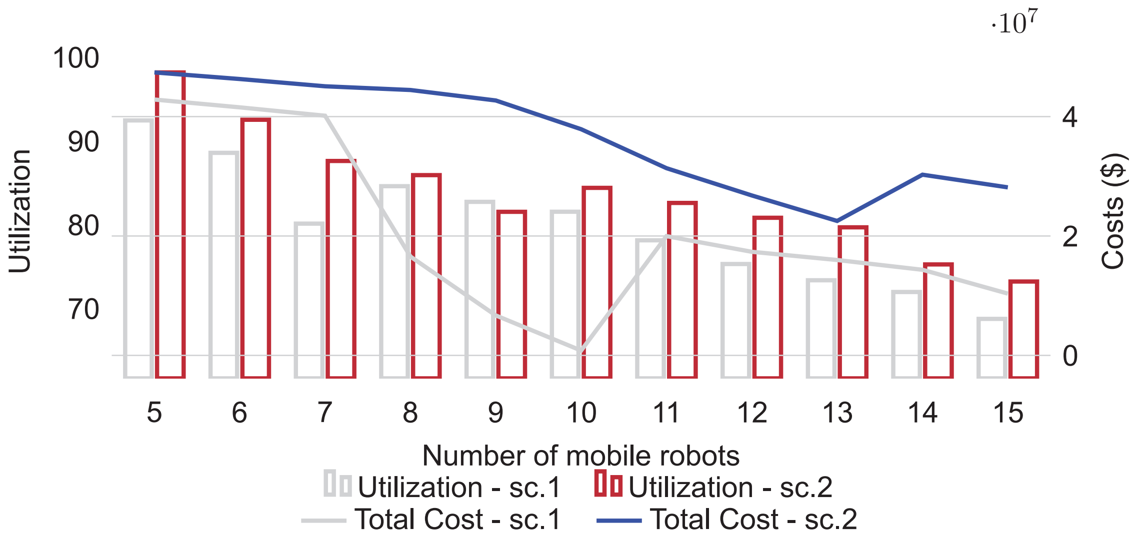


Figure 6.4: Cost and utilization graph with a replenishment when the pods contains less than 50% of their maximum capacity across the two different scenarios.

Using the eq. (6.9), the best solution with each pods refilled with the same kind of articles, is the combination of 13 AGVs and a refill level equal to 50%: the orders are satisfied in 2.746 minutes (+16,17% respect the best combination in scenario 1).

This solution increase, not only the total time to fulfill orders, but also the number of mobile robots that are required. Looking the fig. 6.4 it is also possible to make considerations similar to those made in section 6.3: a lower number of mobile robots have higher average utilization and higher costs. The last results are caused by the high time that the order wait before his elaboration: the few AGVs are not able to carry all the pods that are necessary to fulfill the order in the same time and this is translated to a significant impact on the average orders' elaboration time and on the mobile robots' utilization.
Another possible consideration when looking at the data in fig. 6.4 is that the trend of the cost curve is similar between the two scenarios, but shifted upward and to the right; this can prove the goodness of the results obtained and of the elaboration.

6.5 Scenario 3

The third, and last, scenario that I analyzed modifies the second one, keeping that the SKUs are not spread across the pods, but varying their warehouse storage location according to an ABC class based logic (appendix A.1).
As for the previous cases the maximum utilization of mobile robots occurs when the fulfillment center, that is replicated in simulation, has only 5 AGV and a replenishment level equal to 50% of total capacity of pods.
The comparison of data for this scenario has to be done with both the previous scenario, in order to return the most complete analysis and give an all-around solution that allows to do right conclusions.

6.5.1 Comparison with scenario 1

The data in table 6.9 show a general increment of the main performance' metrics such as the average utilization of mobile robots, the average time necessary to fulfill the orders and the number of transports performed to the picking and replenishment stations. A result that is less in line with these increments is the average waiting time for an order, the orders wait less in the queue, but stay more in elaboration. To check this result I increased the number of replication to 30 in order to reduce it, given the small value of only about 10 seconds, probably due to the presence of stochastic values for pickup and replenishment operations.

Performance's metrics	Mix content	Stocking with ABC policy	Variation	
ρ^{AGV}	86.57 %	94.58 %	+8.47 %	↑
ρ^{pick}	29.36 %	25.84 %	-13.62 %	↓
ρ^{repl}	15.68 %	11.07 %	-4.22 %	↓
T^{order}_{wait}	1.960	1.762	-11.23 %	↓
T^{order}_{elab}	2.861	3.583	+20.15 %	↑
T^{order}_{tot}	4.821	5.345	+9.80 %	↑
ε^{repl}	1521	3141	+51.57 %	↑
ε^{pick}	3041	7330	+58.51 %	↑
ε^{batt}	42	71	+40.85 %	↑

Table 6.9: Model output comparison with 5 AGV e replenishment level equal to 50%.

Costs	5 AGV - Scenario 1		5 AGV - Scenario 3		variation	
AGV cost	135’000	$	135’000	$	-	$
Maintenance	237’500	$	237’500	$	-	$
Charging	35’933	$	60’317	$	24’384	$
Orders’ Value	41’431’000	$	39’840’000	$	1’591’000	$
Total	41’841’500	$	40’272’817	$	1’568’683	$

Table 6.10: Comparison of projection of costs during 5 years between the two scenarios

Returning to the overall result, in general, it is possible to say that a overall deterioration in metrics is present, consistent with what has already been found in the analysis of the second scenario: it is necessary to have more time to fulfill the orders and to use more resources.

Replicating the process performed previously, the cost with this configuration becomes the one reported in table 6.10. The main difference, also in this case, is given by the total orders’ value, which is confirmed the most important parameter of comparison. Evaluating the solution that processes an order most quickly, we find the following combination:

12 AGV with a replenishment level equal to 50%.

With this solution, we found a 76,71% of average AGV's utilization, lower with respect to the first scenario (with 10 AGVs), and 4.977 minutes in average to fulfill all the lines contained in the different orders. The others metrics useful to conduct the comparison are reported in table 6.11.

It is possible to see that the workers have a bigger utilization, and this is related to the

Performance's metrics	Description	Average Value
ρ^{AGV}	AGVs utilization	76.71 %
ρ^{pick}	Picking stations utilization	57.85 %
ρ^{repl}	Replenishment stations utilization	35.44 %
${T^{order}}_{wait}$	Time that the orders wait	1.994
${T^{order}}_{elab}$	Time to elaborate the orders	2.983
${T^{order}}_{tot}$	Order throughput time	4.997
ε^{repl}	Total visit in replenishment stations	1745
ε^{pick}	Total visit in picking stations	7330
ε^{batt}	Total number of recharging	82

Table 6.11: Model output with 12 mobile robots and a minimum replenishment level when the pod are at 50% in scenario 3.

higher number of operations that they have to do: they have a lower number of items to be stored or picked from the same pod, but a largee number of pods where they have to do these operations and this mean that the operators work for more time.

In table 6.12 it is possible to see the main difference between the best solution of scenario 1 and the best solution of scenario 3 which results worst in losses scoring around $ 47'636'000. The economic value is obtained with the over mentioned eq. (6.9), while in fig. 6.5 it is possible to see the graphical comparison useful to understand the differences between average utilization of mobile robots and final costs.

The costs of the third scenario result higher, and are at their lowest when we use 13 mobile robots. This "minimum cost" is calculated using the best combination between mobile robots and replenishment level of the third scenario.

Performance's metrics	Mix content	Stocking with ABC policy	Variation	
ρ^{AGV}	81.15 %	76.71 %	-5.79 %	↓
ρ^{pick}	68.17 %	57.85 %	-17.83 %	↓
ρ^{repl}	38.54 %	35.44 %	-8.74 %	↓
$T^{order}{}_{wait}$	0.784	1.994	+60.59 %	↑
$T^{order}{}_{elab}$	2.518	2.983	+15.59 %	↑
$T^{order}{}_{tot}$	3.302	4.997	+33.92 %	↑
ε^{repl}	1826	1745	-4.64 %	↓
ε^{pick}	3408	7330	+53.51 %	↑
ε^{batt}	75	82	+8.54 %	↑

Table 6.12: Model output comparison between best solution in scenario 1 and scenario 3.

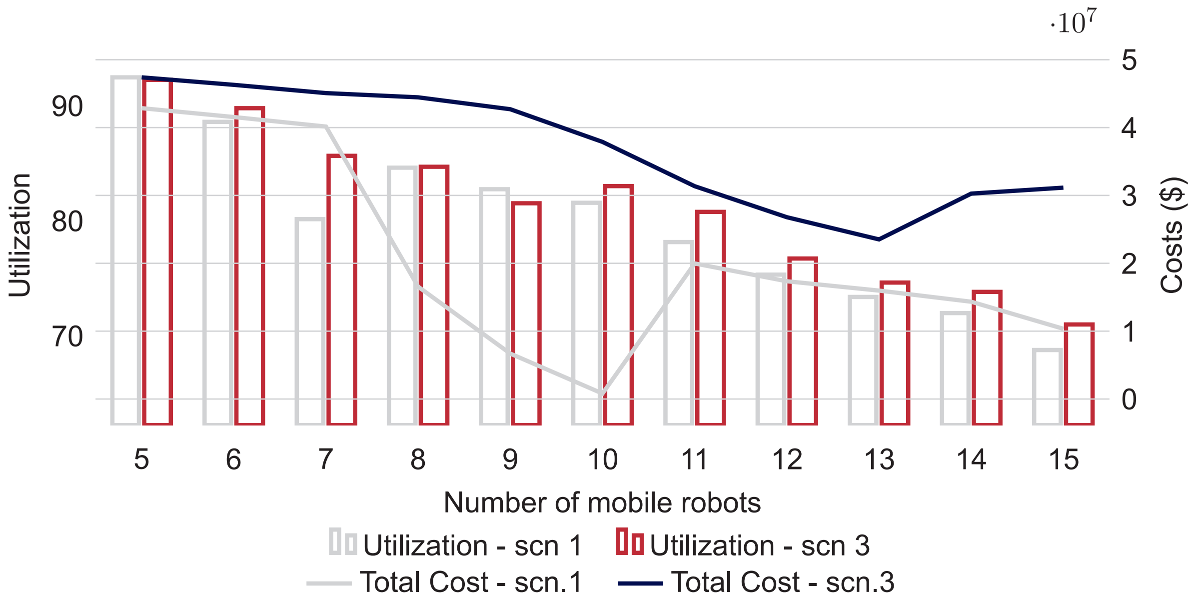


Figure 6.5: Cost and utilization graph with a replenishment when the pods contains less than 50% of their maximum capacity across the two scenarios in analysis.

6.5.2 Comparison with scenario 2

The comparison between the third and the second scenario shows that the results are closer with respect to the previous comparison. In particular, it is possible to see that there is not much difference between the overall performances of this two policies, even if the hypotesys in which the pods have a random location in the warehouse return the worst results. The comparison's data between the best solutions in the second and the third scenario, also reported in table 6.13 confirm that. Specifically, in the table 6.13 I added the average time that each pod travelled for each picking operation, a good metric that is possible to use to understand in a better way the main difference between these two scenarios. While there are not many differences in mobile robots' utilization, operators' workload and global timing, if the pods have a random location in the warehouse storage area they travel for more time (+43.17%) .

Metrics	Random storage	Stocking with ABC pol.	Variation	
n^{AGV}	13	12	-7.70 %	↓
ρ^{AGV}	78.71 %	76.73 %	-2.61 %	↓
ρ^{pick}	48.93 %	57.85 %	+15.42 %	↑
ρ^{repl}	34.97 %	35.44 %	+1.33 %	↑
$T^{order}{}_{wait}$	2.109	1.994	-8.27 %	↓
$T^{order}{}_{elab}$	2.918	2.983	+2.18 %	↑
$T^{order}{}_{tot}$	5.027	4.997	-1.01 %	↓
ε^{repl}	1915	1745	-9.74 %	↓
ε^{pick}	7330	7330	- %	≡
ε^{batt}	87	82	-6.1 %	↓
ω^{pod}	2.965	2.071	-43.17%	↓

Table 6.13: Model output comparison between best solution in scn. 2 and scn. 3.

Another consideration that can be made is the following: the outputs show that there is more or less the same utilization of mobile robots, but to have the same order time throughput with the random location it is necessary to have one more AGV. This result is transformed not only in a higher initial investment for the buying, but also in more expenses for maintenance and charging and, moreover, can also increase the possibilities of collisions.

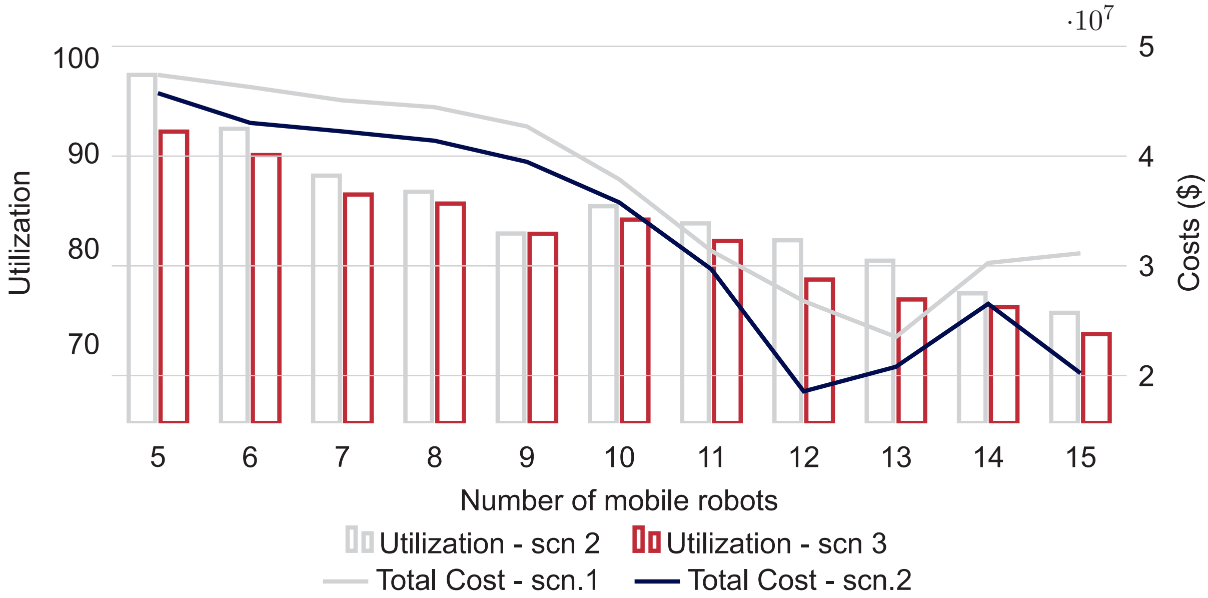


Figure 6.6: Cost and utilization graph with a replenishment when the pods contains less than 50% of their maximum capacity across the second and the third scenarios.

Costs	**Scenario 1**		**Scenario 2**		**Scenario 3**	
n^{AGV}	10		13		12	
Replenishment level	50%		50%		50%	
AGV cost	270’000	$	351’000	$	324’000	$
Maintenance	475’000	$	617’500	$	570’000	$
Charging	64’167	$	72’723	$	69.728	$
Orders’ Value	88’978’000	$	40’776’606	$	41’186’652	$
Total	89’787’167	$	41’867’233	$	42’801’728	$

Table 6.14: Costs’ projection during 5 years between the three scn. using the best combination.

In fig. 6.6 it is possible to see that the cost curve has a really similar trend in all the 3 scenarios, but scenario 2 presents the higher costs. on the contrary, if we focus on the average utilization of mobile robots, it is easy to see that the lowest utilization level emerges in the third scenario.

The projection of costs, reported in table 6.14 confirms that the second scenario is the worst, despite the similar performances compared to the third one since it needs one mobile robots more.

Using the method explained in section 6.2.1 to comparing the three scenarios we obtain the values in table 6.14.

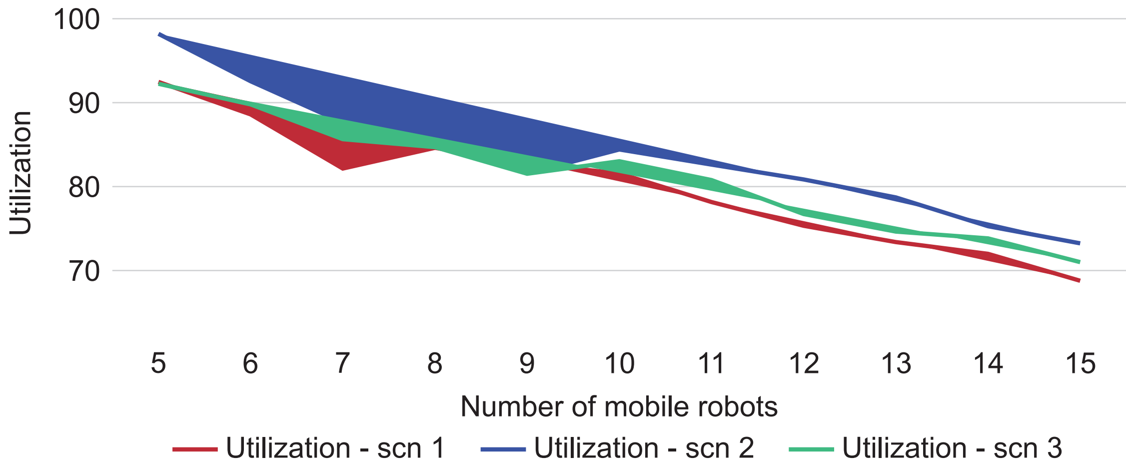


Figure 6.7: Utilization graph with a replenishment when the pods contains less than 50% of their maximum capacity across the three different scenarios.

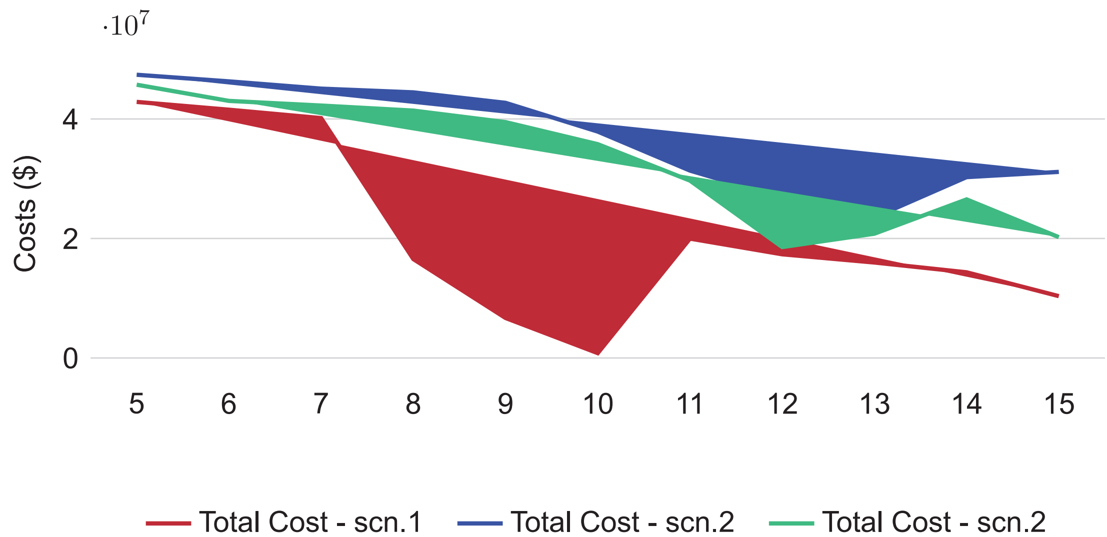


Figure 6.8: Cost graph with a replenishment when the pods contains less than 50% of their maximum capacity across the three different scenarios.

From the fig. 6.7, in reality, we see that all three scenarios exploit AGVs quite well, but the sum of the effects of other parameters performances greatly affect the order throughput outcomes. The result just mentioned can be easily observed in fig. 6.7, where it is possible also to see that the third scenario is completely located halfway between the results of scenario 1 and scenario 2, despite the fact that in any case, the first scenario has the best performances, with a huge saving and the minimum number of mobile robots.

In conclusion, resuming the objective of this study described in section 4.1.1 , the best ratio between the number of mobile robots and pods is near to 10% and the replenishment level required to have the best performance is 50%.

7 Conclusions

The aim of this thesis was to analyze the performances of RMFS through a simulating project to better understand one of the most innovative picking system, specifically a semi-automatic picking system parts-to-picker. This kind of system is revolutionizing the logistic since 2012, right after the acquisition of Kiva System by Amazon.com™, and since then is increasingly used by companies, especially for e-commerce.
For its ability to optimize processes, the topic has been studied several times in the last years, but the subject has so many applications that some aspects are still not developed in the scientific literature. making difficult to conduct all-round studies.
These are some of the motivations that led me to investigate the robot mobile fulfilment systems: I started to conduct researches in the literature, watched videos and read blogs and articles. At the end of this part, I studied the RMFs performances by a discrete-event simulative model that I build with a software called Anylogic™. In general, the simulating approach is very useful to study systems and gives the opportunity to monitor how performances change by changing parameters, and makes it possible to replicate in a virtual environment what might happen in real-life situations. Hence the reasons why simulation is more and more used in the last years.
Starting from the literature study I realized that there were not papers that analyze in the deep themes such as the management of the pods's content. Considering that it was very difficult to find real data about this kind of system I decided to create random databases and make some assumptions that helped me simplify the problems before starting with the model development.
The study was developed in three steps:

1. Study of the literature about warehouses in general, with a special focus on warehouse that use solution such as the RMFS;
2. Define the objectives of the study:
   - Investigate how a change in the replenishment policies and the storage policies

can affect the system order throughput time;

- Comparing the performance measurement for the different scenarios;
- Find the best ratio between pods and robots inside a RMFS's warehouse.

3. Develop the model and conduct the analysis of its results.

The model aims to analyze three different types of SKU storage management and pod storage location management according to the following scenarios:

- Scenario 1: Pods are stored with a random position in warehouse, and each pod is refilled with multi SKUs;
- Scenario 2: Pods have a random location in the storage area, but each pod has only one SKU,
- Scenario 3: Pods contains only one SKU and have a specific location according with the commodity class.

The assumptions made and the made-up data do not allow me to validate the simulative model by comparing the results, but after the analysis it was possible to affirm that the findings are not far from the reality.

The system was studied varying the number of mobile robots and the refill level of the pods. The minimum number of AGVs is 5 because under this value the system had a high utilization of this resource and long order throughput. Considering that I wanted to find a trade-off between the AGV's utilization and the time to pick all the lines in the different orders, I search a scientific methodology to compare the performance's metrics that I collected with the simulation.

The results that I found allow us to affirm that the best replenishment level, regardless of storage policies, is 50%. That allow a perfect trade-off between the time that the operator uses to do the refill and the unavailability of articles when they are required. As affirmed before, this result is consistent with the studies performed by Lamballais et al. (2020) [31] and permit to consider good mine result, despite level of approximation used.

Going on with the studies, I discovered that the best management achieved for each fulfill center is when the SKUs are spread across the different pods. That increases the possibilities for one pod to contain all the articles that are present in the order and this

mean less AGV utilization, less waiting time and less pod travel. The saving data from this results increase the number of orders that can be processed. This result is, also, confirmed in Lamballais et al studies. A limit of this study is that I was not able to create a full study that spread articles of the same commodity classes across the pods. From the results that the simulation returns, considering also how the order database was created, probably, this assumption would not have strong impact on the final output.
At the end, I discovered that the best ratio between the mobile robots number and the pods present in the warehouse is equal to 8%. This value, however, probably is correct only if the fulfill center have the same characteristics of this simulation model.
As mentioned before, the results explained before are in line with some studies conducted in the past, but never demonstrated in a similar way.

7.1 Future works

As a future developments, I suggest continuing studying performances of these systems, in particular, in terms of throughput, lead time and AGV utilization that are the most important performance's metrics for both the customers and the companies. This kind of study should be conducted considering also human factors, important for the workers wellness and their safety on the workplace.
The methodology I tried to build, could be implemented by considering the utilization of the resources (workers, pickers and mobile robots), the discounting of order's value and the value of the articles really ordered. Another useful consideration is relevant to the input data: for good work and to obtain more accuratd results, it would be necessary to have real information from companies that has implemented RMFS in their warehouses. Regarding the simulative model that I developed, unfortunately, it was not possible to develop a flawless model and it presents some weakness, reported in section 6.1.1 At the conclusion of this work, the for-loop is not the best way to replicate what the real management software realizes. The biggest limits of this algorithm are two: the inability to read all the order before the lines's splitting and the the incapacity to upload the assigment of the tasks. The last suggestion is to conduct similar study but with other simulation software, such as Arena, Simul8, or more generic software as Matlab or Phyton.
This should allow to compare results from different simulation's methodology and also to

use the ”library functions” that Anylogic does not have. Moreover, it would possible to linger more on the results analysis and less on the system modeling.

Bibliography

[1] NetComm eCommerce B2c. *Nel 2022 cresce l'eCommerce B2c in Italia: +14%, 45,9 mld €*. 2022. url: www.osservatori.net/it/ricerche/comunicati-stampa//ecommerce-italia-crescita-2022.

[2] eCommerce B2B. *Cresce l'eCommerce di Prodotto (+10%), trainato da food, grocery e arredamento*. 2022. url: www.osservatori.net/it/ricerche/comunicati-stampa/ecommerce-prodotti-valore-italia.

[3] Caseleggio Associati. *E-commerce in Italia 2022*. 2022. url: https://www.casaleggio.it/focus/rapporto-e-commerce-in-italia-2022/.

[4] Ed Romaine. *18 shocking ecommerce statistics about modern order fulfillment*. 2018. url: https://www.conveyco.com/blog/ecommerce-order-fulfillment-statistic/.

[5] AT Kearney. "Excellence in logistics". In: *AT Kearney, Chicago* (2009).

[6] John Longshore and Angela Cheatham. *Warehouse Management and Operations*. 2022, pp. 136–154. isbn: 9781003128939. doi: 10.4324/9781003128939-8.

[7] Sebastjan ŠKERLIČ and Robert MUHA. "Reducing errors in the company's warehouse process". In: *Transport Problems* (2017), pp. 83–92. doi: 10.20858/tp.2017.12.1.8.

[8] Y. Jaghbeer, R. Hanson, and M. I. Johansson. "Automated order picking systems and the links between design and performance: a systematic literature review". In: *International Journal of Production Research* (2020), pp. 4489–4505.

[9] James A Tompkins et al. *Facilities planning*. John Wiley & Sons, 2010.

[10] René de Koster, Tho Le-Duc, and Kees Jan Roodbergen. "Design and control of warehouse order picking: A literature review". In: *European Journal of Operational Research* 182.2 (2007), pp. 481–501. doi: https://doi.org/10.1016/j.ejor.2006.07.009. url: https://www.sciencedirect.com/science/article/pii/S0377221706006473.

[11] Rene De Koster. "How to assess a warehouse operation in a single tour". In: *Technology Report. Erasmus University, Netherlands* (2004).

[12] Lu Zhen and Haolin Li. “A literature review of Smart Warehouse Operations Management”. In: *Frontiers of Engineering Management* 9 (2022), pp. 31–55. doi: 10.1007/s42524-021-0178-9.

[13] *How does RFID work? | Aucxis*. url: https://www.aucxis.com/en/rfid/rfid-technology.

[14] Kaveh Azadeh, René De Koster, and Debjit Roy. “Robotized and automated warehouse systems: Review and recent developments”. In: *Transportation Science* (2019).

[15] Marc Wulfraat. *Is Kiva Systems a good fit for your distribution center an unbiased distribution consultant evaluation*. 2012. url: https://www.mwpvl.com/html/kiva_systems.html.

[16] Amazon. *Look back on 10 years of Amazon Robotics*. 2022. url: https://www.aboutamazon.com/news/operations/10-years-of-amazon-robotics-how-robots-help-sort-packages-move-product-and-improve-safety.

[17] Lin Xie, Hanyi Li, and Nils Thieme. “From Simulation to Real-World Robotic Mobile Fulfillment Systems”. In: *Logistics Research* 12 (Sept. 2019). doi: 10.23773/2019_9.

[18] Marius Merschformann. “Decision rules for robotic mobile fulfillment systems”. In: *Operations Research Perspectives* 6 (2019). doi: https://doi.org/10.1016/j.orp.2019.100128. url: https://www.sciencedirect.com/science/article/pii/S2214716019300946.

[19] Lei Luo and Ning Zhao. “An efficient simulation model for layout and mode performance evaluation of robotic mobile fulfillment systems”. In: *Expert Systems with Applications* 203 (2022). doi: https://doi.org/10.1016/j.eswa.2022.117492. url: https://www.sciencedirect.com/science/article/pii/S0957417422008211.

[20] Manuel D. Rossetti. *Simulation Modeling and Arena*. 2nd. Wiley Publishing, 2015. isbn: 1118607910.

[21] Steven M Manson. “Simulation modeling”. In: *International Encyclopedia of Human Geography*. Elsevier Inc., 2009, pp. 132–137.

[22] AnyLogic™. *AnyLogic Simulation Software*. https://www.anylogic.com/. 2022. url: https://www.anylogic.com/.

[23] Andrei Borshchev. *The big book of simulation modeling: Multimethod modeling with anylogic 8*. AnyLogic North America, 2020.

[24] Pedro Palandrani. *Theme lab: A look inside an Amazon fulfillment center*. 2020. url: https://www.globalxetfs.com/theme-lab-a-look-inside-an-amazon-fulfillment-center/.

[25] Amazon.com. *Amazon Virtual Tours*. 2022. url: https://amazontours.com/it.

[26] Robotique et Mathematiques - 6ème. *Robots Kiva : Amazon*. Youtube. 2017. url: https://www.youtube.com/watch?v=ULswQgd73Tc.

[27] Allan Larsen. *42417 Simulation in Operation Management - Course notes*. DTU Management, 2021.

[28] Anylogic™. *Anylogic Library Reference Guides*. 2022. url: https://anylogic.help/library-reference-guides/index.html.

[29] Averill M Law, W David Kelton, and W David Kelton. *Simulation modeling and analysis*. Vol. 3. Mcgraw-hill New York, 2007.

[30] Ruiping Yuan et al. "Storage Assignment Optimization in Robotic Mobile Fulfillment Systems". In: *Complexity* 2021 (2021).

[31] Tim Lamballais Tessensohn, Debjit Roy, and René B.M. De Koster. "Inventory allocation in robotic mobile fulfillment systems". In: *IISE Transactions* (2020). doi: 10.1080/24725854.2018.1560517. url: https://doi.org/10.1080/24725854.2018.1560517.

[32] Yeming Gong, Mingzhou Jin, and Zhe Yuan. "Robotic mobile fulfilment systems considering customer classes". In: *International Journal of Production Research* (2020). doi: 10.1080/00207543.2020.1779370.

[33] *AGV and Amr Robot Home*. url: https://www.agvnetwork.com/agv-cost-estimation-how-much-does-an-automated-guided-vehicle-cost#wellwit-under-ride.

[34] *Amazon - quarterly results*. url: https://ir.aboutamazon.com/quarterly-results/default.aspx.

A Input data

A.1 Order database

A.1.1 E-commerce orders

During 2022 Amazon.com™ publish some general information about his fulfillment centers [16]: they state more than 520.000 mobile robots in all warehouses.
Searching online others information, I discovered that there are from 2000 to 5000 pods in each Amazon™ fulfillment center, and each mobile shelf is able to contain more or less 2000 SKUs.
In 2018, the data about e-commerce orders for each day reported from 20000 to 40000 for the only italian market [4].
Starting from these publish data[1], I try to create a good input database for my model.
During 2018, Amazon.com™ has 185 fulfillment centers, in all the world, and only 7 of that in Italy.

	Range	Average value
Number of fulfillment centers in the world (2018)	185	-
Number of fulfillment centers in Italy (2018)	7	-
Number of fulfillment centers in the world (2022)	305	-
Number of fulfillment centers in Italy (2022)	15	-
Number of mobile robots (2022)	520.000	-
Quantity of pods in each warehouse (2018)	2000-5000	3500
Number of mobile robots in each fulfillment center[2]	-	1700
Number of SKUs storable in each pod	2000	-
E-commerce orders for each day	20.000 - 40.000	30.000

Table A.1: General information about e-commerce's orders

[1] I ask data during the Amazon Tour and directly to Amazon.com™ offices, but for internal policies, they cannot answer me with more precious information. However they confirmed that the data that I use are, with not to much approximations, correct.

[2] Found such as 520.000 divided by 305

From the data reported in table A.1 I have actualized the e-commerce data for 2018 to date using the economic growth publish by Amazon.com™ in his quarterly results [34] and reported in table A.2

Year	Total e-commerce orders	Growth[3]	Increment
2018	30.000		
2019	36.600	+ 22%	+ 6.600
2020	50.140	+ 37%	+ 13.540
2021	60.670	+ 21%	+ 10.530
2022	65.000	+ 8%	+ 4.330

Table A.2: E-commerce orders/day increment based on Amazon.com™ growth

Scaling now the value founded with the number of fulfillment centers in Italy in 2022, I got that each elaborate approximately 4320 orders/day. Now, from the same data reported in table A.1 compared with the dimension of my model summarized in table A.3, I discover that my model have to elaborate more or less 145 orders/day.

	My model	Ratio between warehouses
Number of mobile robots	Under study	
Ratio between mobile robots and pods	Under study	
Quantity of pods in warehouse	120	2.91%
Number of SKUs storable in each pod	60	3,33%
E-commerce orders for each day	500	11,54%

Table A.3: Fulfilment centers comparison

In order to follow the forecast growth of e-commerce orders in the current year I increase the value found in 500 orders/day.

This allow also to get lower the error create with these approximations and more closer to e-commerce reality.

A.1.2 Time distribution

The orders' timing follow an exponential distribution during days with λ=36 that are able to satisfy an average time between orders of 2/2.5 minutes. Each order can have from 1 to 6 lines, one piece for line, and the majority of the order will have only three lines. To model the generation of order's lines I chose the triangular distribution represented in fig. A.1. In total, the workers pick 3735 elements across two days simulation.

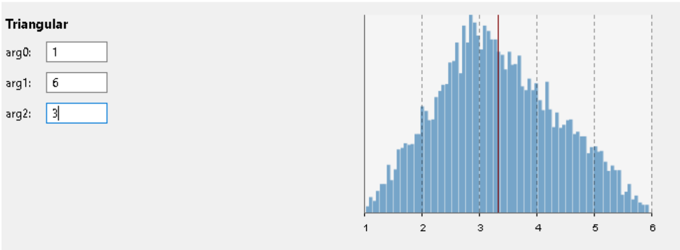


Figure A.1: Distribution of order's lines

A.1.3 Final order's database

The lines for each order was generated randomly between 1 and 5 whit Excel, while the SKUs' distribution follow an exponential distribution (λ= 1/2) as fig. A.2. The three classes

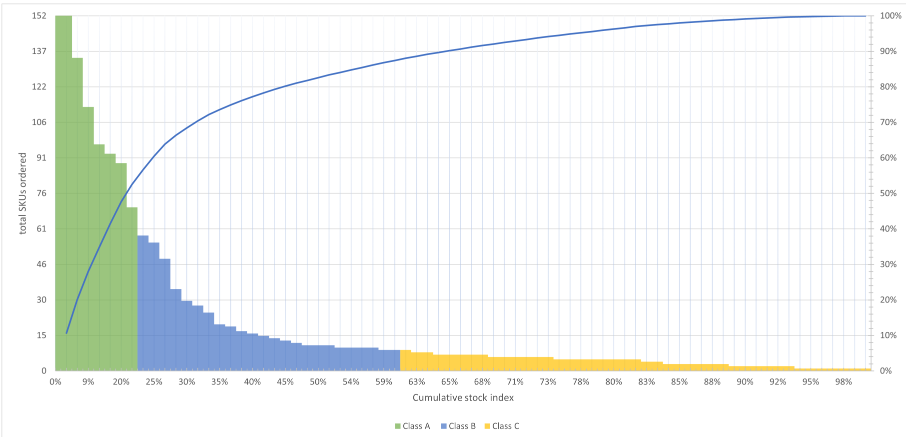


Figure A.2: Orders distribution with class subdivision.

are defined such as:

1. Class A: 6 SKUs that represent approximately the 20% of the total number access and that they constitute around of 50% of the total quantity stored in my model;

2. Class B: 25 SKUs that represent approximately the 37% of the access and that they constitute around of 30% of the total quantity stored in my model;

3. Class C: 43 SKUs that represent approximately the last 45% of the access and that they constitute the 12% of the total quantity stored in my model.

The precise data of class division using ABC logic are given in the follow table A.4.

Class	Total SKUs ordered	Cumulative stock index	Cumulative Access Index
A	2030	21.67%	53.90%
B	1225	30.83%	28.37%
C	480	44.17%	12.25%
Total	3735	100%	100%

Table A.4: Cumulative Stock Index e Cumulative Access Index for orders in my model across two days simulation.

B Output

B.1 Simulation results

B.1.1 Replication calculation

For find the optimal value of the KPIs that are define in section 4.1.4 I use the methods explained in section 3.4.1. In particular, for the first analysis each scenario run with 15 replications, such as Rossetti advises [20]. Considering all the metrics that the model returns reported in table B.1 Using the **Half-width method**, with an *half-width = 0.1* and a *Confidence Interval = 95%*, I found that are necessary, in order to have a good results, 12.04 replication (eq. (B.1)). This calculation confirm that the first results obtained with 15 replication is enough.

$$h_0 = t_{1-(5\%/2),14}\frac{0.05}{\sqrt{15}} = 0.09 n \cong n_0\left(\frac{h_0}{h}\right)^2 = 15 \bullet \left(\frac{0.09}{0.10})\right)^2 = 12.04 \tag{B.1}$$

Analyzing the same values with a practical approach we found fig. B.1, where is possible to see that the average value is very stable after the first 10th replications.

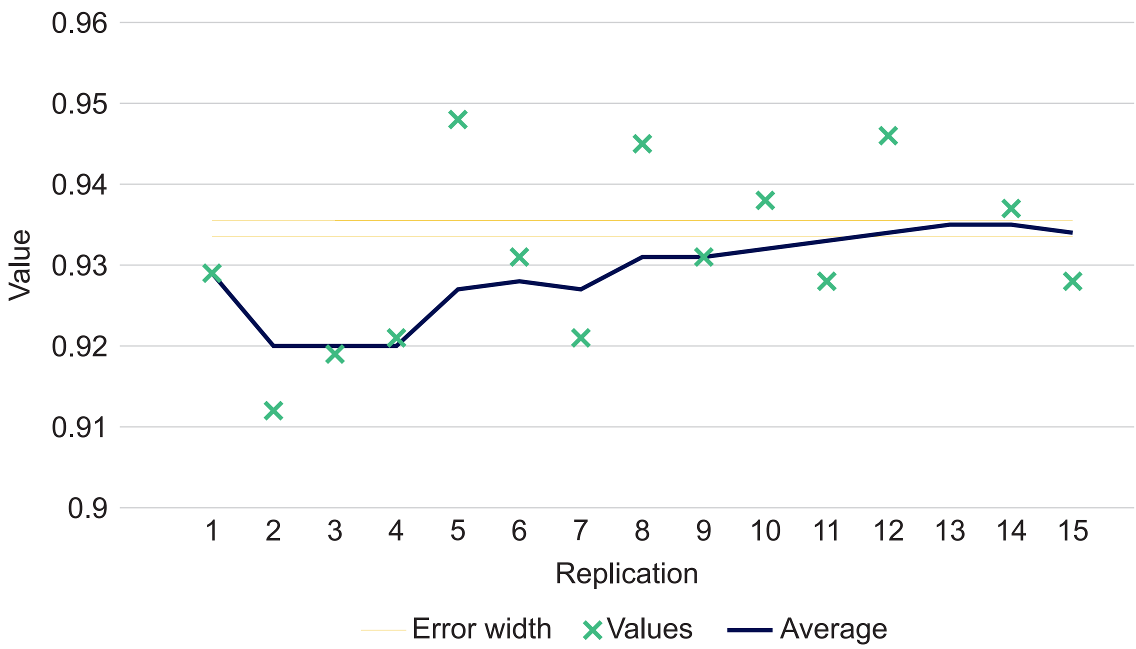


Figure B.1: Value and average across the 15 replications.

Replication	ρ^{AGV}	$\overline{T}^{pick}{}_1$	$\rho^{pick}{}_1$	$\overline{T}^{pick}{}_2$	$\rho^{pick}{}_2$	$\overline{T}^{repl}{}_1$	$\rho^{repl}{}_2$	$\overline{T}^{repl}{}_2$	$\rho^{repl}{}_2$	$\overline{T}^{order}{}_{wait}$	$\overline{T}^{order}{}_{elab}$	$\overline{T}^{order}{}_{tot}$	$\overline{w}^{pod}$
1	0.929	0.414	0.496	0.501	0.4151	0.608	0.415	0.25	10.608	0.248	1.406	5.3913.985	2.893
2	0.912	0.409	0.488	0.501	0.411	0.448	0.41	0.247	10.549	0.248	1.448	5.3853.968	2.849
3	0.919	0.406	0.495	0.498	0.4131	0.593	0.413	0.25	10.475	0.246	1.387	5.3683.954	2.875
4	0.921	0.408	0.487	0.497	0.4131	0.545	0.413	0.248	10.538	0.247	1.387	5.3033.946	2.857
5	0.948	0.413	0.488	0.499	0.4141	0.479	0.414	0.246	10.483	0.248	1.393	5.3143.947	2.872
6	0.931	0.412	0.495	0.493	0.4071	0.425	0.407	0.248	10.411	0.245	1.383	5.3673.968	2.846
7	0.921	0.417	0.488	0.497	0.4141	0.458	0.414	0.247	10.403	0.244	1.402	5.3523.918	2.892
8	0.945	0.406	0.489	0.497	0.4111	0.592	0.411	0.248	10.544	0.245	1.378	5.3763.961	2.84
9	0.931	0.414	0.491	0.497	0.411	0.535	0.413	0.247	10.466	0.246	1.391	5.2843.952	2.865
10	0.938	0.408	0.495	0.495	0.4121	0.476	0.412	0.247	10.583	0.246	1.381	5.3733.98	2.861
11	0.928	0.407	0.495	0.495	0.4111	0.424	0.411	0.266	10.504	0.247	1.391	5.293.938	2.843
12	0.946	0.458	0.487	0.547	0.411	0.564	0.414	0.249	10.482	0.244	1.395	5.3583.958	2.872
13	0.935	0.409	0.487	0.511	0.4071	0.424	0.407	0.248	10.417	0.248	1.378	5.2923.957	2.864
14	0.937	0.408	0.493	0.499	0.4141	0.499	0.414	0.247	10.586	0.244	1.399	5.3183.907	2.843
15	0.928	0.414	0.492	0.492	0.4131	0.587	0.413	0.245	10.503	0.245	1.393	5.3163.942	2.851

Table B.1: Results of a simulation with 15 replications.