RAYANE GIULIANA DE OLIVEIRA CECHIM

Performance evaluation of a new intralogistics systems

São Paulo

2019

RAYANE GIULIANA DE OLIVEIRA CECHIM

Performance evaluation of a new intralogistics systems

Trabalho de Formatura apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Diploma de Engenheira de Produção.

São Paulo

2019

RAYANE GIULIANA DE OLIVEIRA CECHIM

Performance evaluation of a new intralogistics systems

Trabalho de Formatura apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Diploma de Engenheira de Produção.

Orientador: Professor Dr. Marco Aurélio de Mesquita

São Paulo

2019

Catalogação-na-publicação

Cechim, Rayane Giuliana de Oliveira
Performance evaluation of a new intralogistics systems / R. G. O. Cechim -- São Paulo, 2019.
106 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Produção.

1.Sistema de armazenamento e recuperação automáticos 2.Pallet misto 3.Separação e preparação de pedidos 4.Armazém automatizado 5.Simulação I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Produção II.t.

RESUMO

O presente trabalho tem como objetivo realizar um benchmark de sistemas inovadores no processo de separação e preparação de pedidos, além de avaliar e comparar a performance de dois sistemas diferentes de armazenamento e recuperação automático por meio da simulação da operação dos sistemas em um armazém virtual. O estudo foi proposto por uma empresa italiana, Eurofork, fornecedora de soluções de manuseio de cargas, que atualmente está desenvolvendo um dos sistemas analisados. O sistema, ainda não existente, foi conceitualizado pela empresa de forma a agilizar a criação de pallets mistos nos armazéns, por meio da modificação da maneira que são feitos os processos de retirada dos produtos pedidos do armazém e de paletização dos mesmos.

Para o desenvolvimento desse estudo, foi realizada uma análise da literatura sobre os processos que ocorrem nos armazéns, focando principalmente no de retirada de produtos dos racks em que estão armazenados e nos sistemas automatizados existentes para a realização desse processo. A análise de literatura também abordou estudos que modelam e simulam sistemas de armazenamento e recuperação automático. Por fim, também foram verificados na literatura problemas e demandas para a criação de pallets mistos, tendo em vista que o sistema em desenvolvimento foca na criação de pallets mistos.

Após a análise da literatura, foram desenvolvidos os modelos de simulação dos sistemas e definidos os cenários das simulações para a realização de um experimento fatorial completo. Por meio de análise de variância dos resultados das simulações foi possível identificar fatores que impactam a performance de cada sistema. Por fim, pode-se também realizar a comparação da performance dos dois sistemas em diferentes condições.

Palavras-chave: Sistema de armazenamento e recuperação automáticos. Pallet misto. Armazém automatizado. Separação e preparação de pedidos. Simulação.

ABSTRACT

The objectives of this final term project are to conduct a benchmark of innovative order picking systems and the evaluation and comparison of two different autonomous vehicle storage and retrieval systems through a simulation study of the systems operation in a virtual warehouse. The project was proposed by an Italian company, Eurofork, a load handling solutions provider, that is developing one of the systems analyzed. This system, still being developed, was conceptualized by the company in order to improve the process of building mixed pallets in warehouses by changing the order picking and palletizing processes.

In order to develop this study, first it was carried an analysis of the literature about the process in warehouses, focusing in the order picking and automated systems that performs this process. The literature review also contemplated studies about automated storage and retrieval systems modelling and simulation. Lastly, topics related to demand for mixed pallets and problems building them was also studied, since the system being developed by Eurofork aims to improve the process of building mixed pallets.

After the literature review, the simulation models for the systems were developed and the scenarios for the analysis defined in order to execute a complete factorial experiment. By the analysis of variance of the simulation results, it was possible to identify some factor that impacts the systems' performances. Finally, the simulation results were also used to compare both systems' performances under different configurations.

Key-words: Autonomous vehicle storage and retrieval system. Mixed pallet. Picking process. Automated warehouses. Simulation.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AS/RS	Automated Storage and Retrieval System
AVS/RS	Autonomous Vehicle Storage and Retrieval System
CARG	Compound Annual Growth Rate
FCFS	First come First Served
I/O point	Input/Output point
SBS/RS	Shuttle-Based Storage and Retrieval System
SKU	Stock Keeping Unit

CONTENTS

1. INTRODUCTION

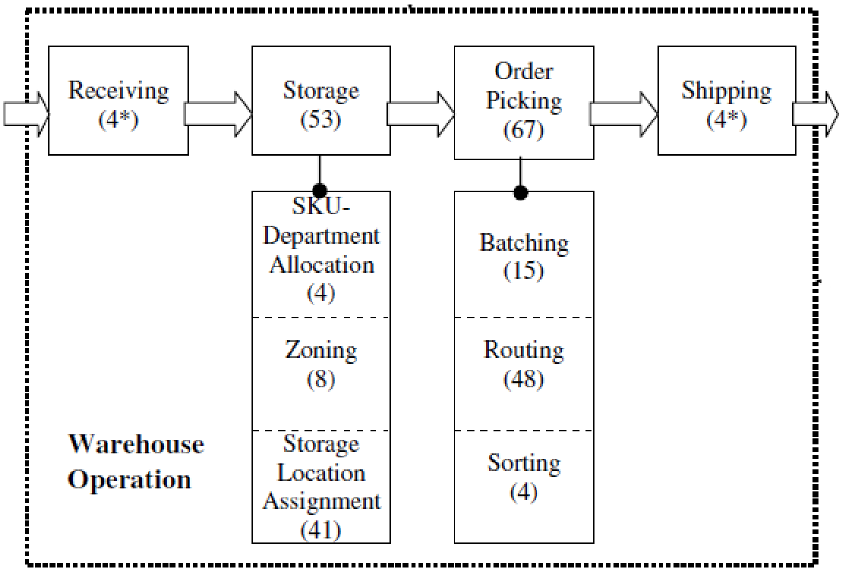
In this chapter it is explained a context of warehouse automation and the application of automated systems in order to improve the picking operation performance. Besides that, the project’s problem and objectives are presented.

1.1. Warehouses automation technologies

Warehouses are an important element of supply chains, even though some companies consider them and materials handling as non-value-added activities. Although they help to improve the efficiency and increase the customer service, the storage and transportation activities inside the warehouse does not add value to the product. The functions of the warehouse include to smother the variability resulted from several factors (for instance, product seasonality, batching in production, transportation) by buffering the material flow inside the supply chain; to consolidate different products from different suppliers according to the customer order; besides other activities as kitting, pricing, labeling and product customization. (GU; GOETSCHALCKX; MCGINNIS, 2007 and MANZINI; BOZER; HERAGU, 2015).

Higher performance from the warehouses is required form in the market due to market competition. The operation inside a warehouse consists of four main processes, they are: receiving, storage, picking, and shipping, as showed in Figure 1 (GU; GOETSCHALCKX; MCGINNIS, 2007 and MANZINI; BOZER; HERAGU, 2015). Order picking, the process of retrieving products from storage to attend a client order, is considered as the most laborious and expensive of these processes in most of the warehouses. The cost related to this process is estimated to be around 55% of the total warehouse operating expense. (GU; GOETSCHALCKX; MCGINNIS, 2007; AZADEH, KOSTER AND ROY, 2017; KOSTER, LE-DUC AND ROODBERGEN, 2007; FRAZELLE, 2002 IN GU; GOETSCHALCKX; MCGINNIS, 2007).

Figure 1: Warehouse operation scheme and the number of articles reviewed per process

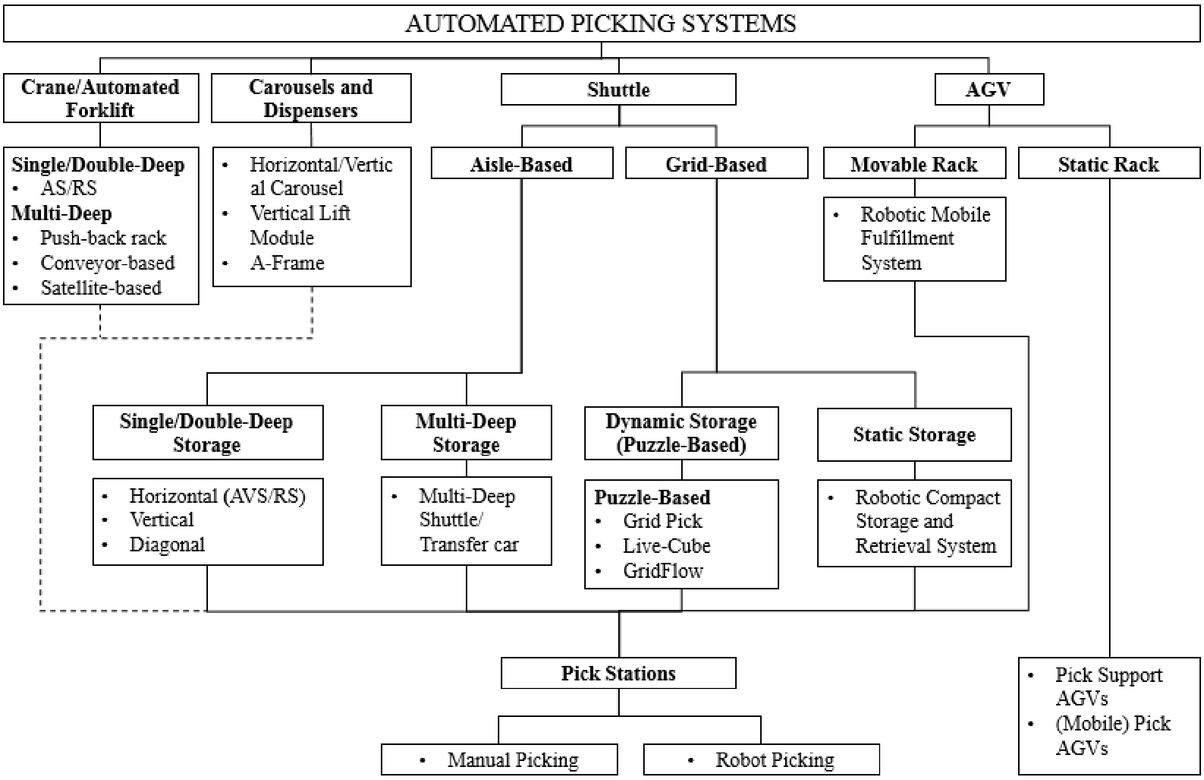


Source: Gu; Goetschalckx; Mcginnis (2007)

For this reason, the picking operation has attracted more attention from researchers than the others, as showed in the Figure 1, in which the number inside the parenthesis represents the quantity of articles focusing on each topic (GU; GOETSCHALCKX; MCGINNIS, 2007). Besides that, this process has become ideal for automation, since it is so costly and repetitive. The warehouse automation started around the 1960s, when automated storage and retrieval systems (AS/RS) were installed in German warehouses to execute the storage processes but still with manual pick stations, nowadays the picking process can be fully automated by some systems (AZADEH, KOSTER AND ROY, 2017). Just for the automated storage and retrieval system market, the compound annual growth rate (CAGR) is estimated to be 7.7% between 2019 and 2024, a growth from USD 7.6 billion to USD 11.0 billion (MARKETS AND MARKETS, 2019).

Several automated picking systems have been developed since then, and they can be classified as exhibited in Figure 2. But as pointed by Azadeh, Koster and Roy (2017), most of them have not been studied in academic literature, even though their application in warehouses is increasing. Considering this, in this study a technology in the development phase is simulated and compared to an existing one, in order to evaluate if the system's performance is improved under certain scenarios using the new technology.

Figure 2:Classification of automated picking systems.



Source: Azadeh, Koster e Roy (2019)

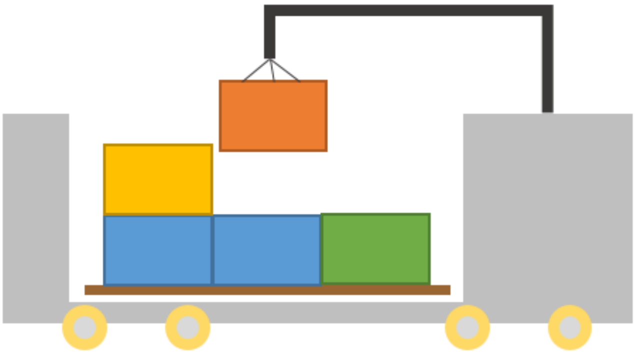
1.2. The innovative intralogistics project at Eurofork S.P.A.

The student has studied Engineering and Management for two years at the Politecnico di Torino in Turin, Italy, through a double degree program with the Polytechnic School of the University of São Paulo. There, as thesis project to get the master's degree, it came up with the opportunity to develop this study proposed by Eurofork that evaluates a new intralogistics system.

Eurofork. is an Italian company based in Turin and one of the main players on the global market of handling devices. The company provides two different handling systems for automated warehouses, telescopic forks and shuttle systems. Considering the trend of mixed pallet orders instead of full pallet, the company is developing an innovative intralogistics system, a new shuttle-based product. It is a system based on a new version of Eurofork's product, ESMARTSHUTTLE, and its objective is to bring flexibility to the picking process.

This innovative intralogistics system consists of a new autonomous vehicle storage and retrieval system (AVS/RS) that differentiates from the current one due to the presence of a robotic arm attached to the shuttle, this new shuttle is illustrated in Figure 3, while an example of the classic shuttle used in the current system can be observed in Figure 4.

Figure 3: Example of the concept for the shuttle used in Eurofork's system considered in this thesis



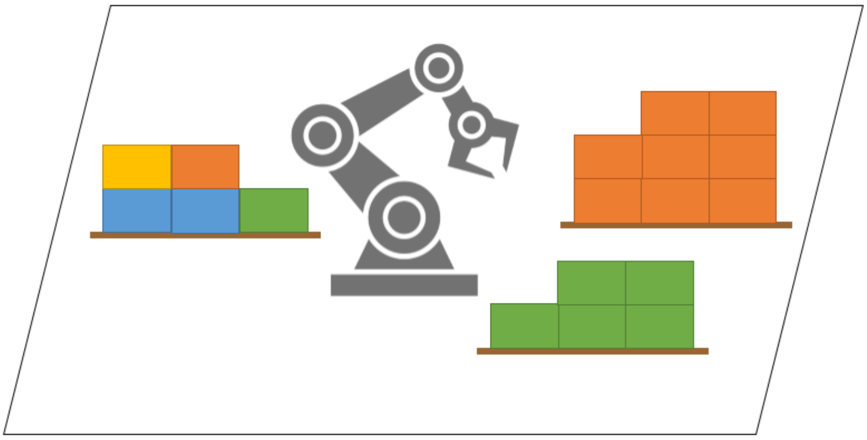
Source: the author

Figure 4:Shuttle used in AVS/RS considered in this thesis

Source: Ekren and Heragu (2009)

In the current system considered, the shuttle transports homogeneous pallets from their rack locations to a picking station where a robot picks the items ordered from homogeneous pallets and builds the ordered one, as illustrated in Figure 5. In the new system, the shuttle with a robotic arm attached will be able to pick products from a homogeneous stored pallet at its location, eliminating the need to retrieve the pallet to a picking station. Besides that, it aims to increase the system throughput capacity, since it does not require the full pallet to be retrieved and then stored again after the products are picked reducing the transportation time for these pallets.

Figure 5: Example of a palletizing station



Source: the author

1.3. Statement of the problem

Automated storage and retrieval systems and their application on warehouses have been previously studied on the literature. But since Eurofork is developing a new system, there is no information about how the proposed system performs and the impact of the modifications when compared to the current system. The company wonders if the new system will show higher performance than the current one when both presents the same number of shuttles, as expected. Besides that, they are looking for to understand how the system efficiency can be influenced by the many variables of the system, for instance arrival rates, number of stock keeping units (SKUs), number of aisles and tiers, etc.

Before starting to invest and to develop it, the company seeks to estimate how efficient it will be compared to an existing autonomous vehicle storage and retrieval systems under different configurations. The objective of this performance evaluation is to verify if the system indeed presents better performance than the new one under different scenarios and also identify possible critical variables that Eurofork needs to pay attention to in order to create a system that outperforms the existing ones.

Since the company is still on the early development phase of the product, it has not tested a real case application of the new system, so a simulation project in a virtual warehouse will be used to compare its operation under the existing system and the new one. Considering this, the problem analyzed in this thesis is a comparison of two different types of automated storage and retrieval systems under different scenarios. In order to facilitate the reading of the thesis the system conceptualized by Eurofork is addressed as Eurofork’s system or new system, while the other one is addressed as current system or current AVS/RS.

Besides that, since the concept of what the system will do is defined, but the technology is not, an analysis of the literature will be done in order to verify the current state of the art of the technologies applied to build mixed pallets, focusing on those that do not demand complex infra-structure for the picking process. In this way, it can be verified if something similar could be applied on Eurofork’s project.

1.4. Objectives

This thesis has as final objective the evaluation a new shuttle-based storage and retrieval system that allows to build pallets with flexible composition, the so called mixed-pallets. To do so, there are two different objectives that will be developed in this study. The first one is a benchmark in order to verify features of other innovative picking system that could be applied to the Eurofork’s one. The second one is focused on the performance evaluation and comparison of the two automated and retrieval system, the current and the new one, through a simulation study.

The objective of analyzing available technologies is to help Eurofork to develope the concept of new shuttle structure that allows the system to build the mixed pallets in the aisles of the rack. Essentially, this study will provide insights to the producer about possible technologies that can be applied on its product, since the company has not established the exact features of the system.

Although this study will analyze available technology and provide the insights, the main objective of this thesis is to evaluate if the new system is viable, in other words if it performs better than the current one under some conditions. To achieve this objective, a virtual warehouse storage and its operation will be modeled in a simulation software, Simpy, getting as output the performance measures. The indicators collected, cycle time and throughput capacity, will then allow the comparison to verify if the new system has a superior performance in relation to the old one and will help to understand how to improve the new one’s performance. Besides that, since in the simulations it was considered only one vehicle, the resources utilization rates were collected to verify opportunities of improving the system by adding new vehicles to the system.

1.5. Outline of the thesis

This chapter of this study presented an overview of the systems analyzed and the objectives of the final term project. In the chapter 2, it is presented a literature review about topics related to the study. Then, in chapter 3 the benchmark developed by presenting some innovative technologies analyzed as well as some insights for the development of the new system. The system’s modelling, simulation and evaluation are divided into: the conceptual and computational modelling is presented in chapter 4, the scenarios and results in chapter 5 and a discussion about the outcomes is developed in chapter 6. The last chapter contains the conclusions of this final term project.

2. LITERATURE REVIEW

In order to understand better the automated systems operation in the warehouse, the literature about order picking processes, warehouse automation and mixed pallets was analyzed. The analysis was used further to develop the simulation model of the autonomous vehicle storage and retrieval systems considered in this project.

2.1. Order picking operation and its automation

Order picking/selection is defined as “The major activity in most warehouses. It involves the process of obtaining a right amount of the right products for a set of customer orders” by Koster, Le-duc and Roodbergen (2007) in their literature review about the order picking activity. The process of the order picking operation is: first the clients’ orders are grouped and scheduled, then the stock on locations are designated to each product ordered, next the orders are released to the floor and finally the items are picked from storage locations and disposed (KOSTER, LE-DUC and ROODBERGEN, 2007).

Order picking process is considered an expensive, labor intensive and repetitive process. (FRAZELLE, 2002 in GU, GOETSCHALCKX and MCGINNIS, 2007; AZADEH, KOSTER and ROY, 2017). In addition to these aspects, in some cases problems related to this activity can also be poor ergonomics and the difficulty to attend the high demand for high-quality employers to work in shifts. (AZADEH, KOSTER and ROY, 2017). These among other factors, such as advent of e-commerce and warehouses operating 24/7, are indicated by Azadeh, Koster and Roy (2017) as reasons of the increasing application of automation in warehouses.

Another factor related to the increase of automation in warehouses is pointed out by Westernacher (2017), it is the higher complexity and frequency of orders. According to the report, due to the high complexity of orders, human operators can make more mistakes and consequently incur in more costs, while automated systems are more accurate. In addition to it, the automation of warehouses improves the time-efficiency of the operation, important element considering that the warehouses are getting more frequent orders.

The automation of warehouses is becoming more popular and important. In 2016, automated material handling to support goods-to-man picking were used by more than 10% of warehouses in the United States (PRG, 2016 in WESTERNACHER, 2017). Besides that, the forecast for the end of 2021 is an increasing of 15 times in the number of robots operating in warehouses (TRACTICA, 2017 IN WESTERNACHER, 2017). Westernacher (2017), a report about trends in warehouse automation, points out that technology has been applied with great extent in order to increase the productivity.

From the beginning of automation in warehouses in the 1960s to the present, much has been developed in terms of automated systems, especially in the last decade. As showed in the classification of automated picking systems made by Azadeh, Koster and Roy (2017) in Figure 2, there are several different technologies available in the market. However, despite the increase of options, the lack of flexibility is still a problem according to the Annual Warehouse and Distribution Center Operations Survey and might be the reason of a 4% drop of respondents that used automated picking systems in 2015 compared to 2016 (SUPPLY CHAIN MANAGEMENT REVIEW, 2016).

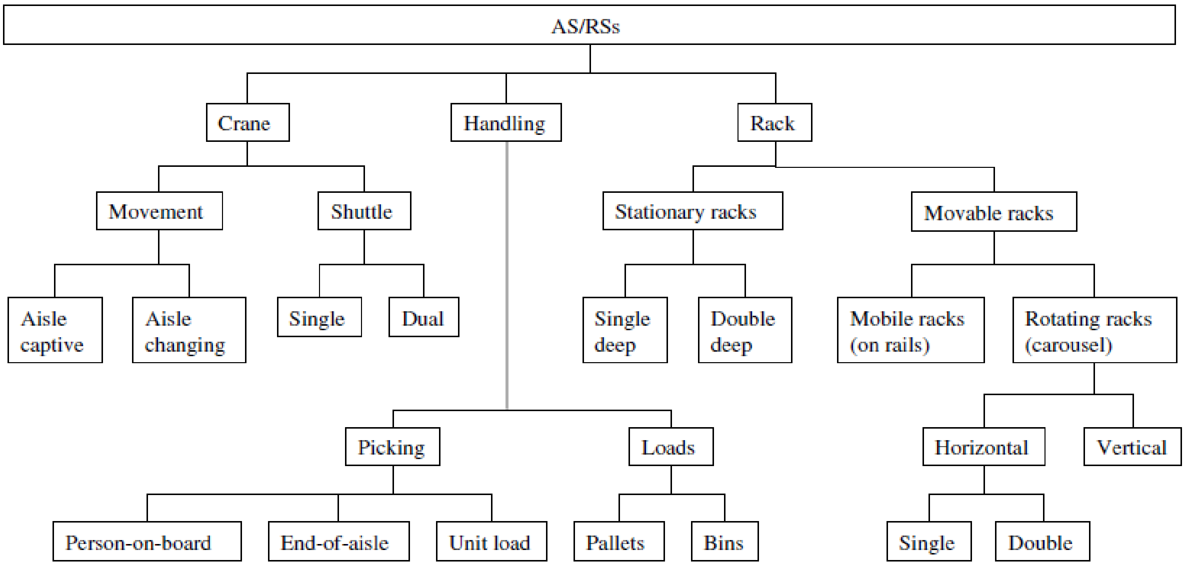
2.2. Automated storage and retrieval systems

Automated storage and retrieval systems (AS/RS) are defined by Roodbergen and Vis (2009) as "warehousing systems that are used for putting products in storage and for retrieving those products from storage to fulfill an order in both distribution and production environments". They were developed in the 1950s as systems to store inventories in pallet warehouses, since then their application in warehouses has increased until it became possible to use it also for order picking activities and consequently became popular in warehouses (ROODBERGEN; VIS, 2009 and AZADEH, KOSTER AND ROY, 2017).

The AS/RS presents many benefits, among them it improves throughput, the accuracy, ergonomics, worker safety and labor productivity, besides that it also allows highest possible storage density, requires less floor space, provides real-time inventory control and enhances product security (MATERIAL HANDLING INSTITUTE, 2019). Despite these advantages, the authors also mention some of the downsides of this kind of system, the need of higher investments and the less flexibility to change the physical layout (ROODBERGEN; VIS, 2009).

Automated storage and retrieval systems are basically composed by racks, where the products are stored; cranes, instrument that transports, picks up and drop off the items; aisles, the space between the racks where the cranes can circulate; input/output points (I/O-points), place where the retrieved products are left and the ones to be stored are picked; pick position, where operators can select the ordered items from retrieved loads. There are many possible configurations for these components of the system, resulting in several types of AS/RS, as showed in Figure 6 (ROODBERGEN; VIS, 2009).

Figure 6: Possibilities for automated storage and retrieval systems



Source: Roodbergen and Vis (2009)

As showed in Figure 6, there are two alternatives in relation to the crane's movement and structure of the equipment. With respect to the equipment, it can be able to transport only one load, called single shuttle, or two loads at once, the dual shuttle. About the equipment movement, the cranes move inside the aisles, in the case of aisles captive cranes the equipment moves on a single aisle and consequently each aisle needs one crane. The other option is aisle changing cranes, this type can transport products in different aisles. There is a special case of the latter called autonomous vehicle storage and retrieval systems (AVS/RS) (ROODBERGEN; VIS, 2009).

In AVS/RS, there are two possible options in relation to the shuttle movement, it can be tier-to-tier or tier-captive. While in tier-to-tier systems the vehicle can travel from one tier to another using the lift, in the tier-captive the vehicle operates only in one tier and the lifts transports only the loads, not the vehicle (MANZINI et al., 2016). Since there is one vehicle for each tier, the tier-captive system requires more vehicles and thus is more expensive, but it also presents higher performance compared to tier-to-tier because of the independence of the vehicle and lift movement (MARCHET et al., 2012; EPP; WIEDEMANN; FURMANS, 2016). Because of this higher efficiency, most of the solutions provided by companies are tier-captive single-aisle configurations according to Lerher et al. (2015).

With regards to the handling unit part of the system, there are three options for the picking process and two for the type of load stored and retrieved. The picking activity can be performed by a person transported by the crane and that picks the requested items at the stored load's location, called person-on-board. If it does occur in the storage location, it can be at a picking area where the retrieved load is dropped off and the items of the customer's order are picked, this configuration is called end-of-aisle, or the AS/RS can deal with unit loads only, in this case the customer order the whole load, thus there is no need to select and separate items from it. Besides that, the load transported, stored and retrieved by the system can be pallets or bins, if it is the latter the system is called miniload AS/RS (ROODBERGEN; VIS, 2009).

Finally, there are different configurations for the rack, it can be single or double deep and stationary or movable. The first one allows only one load to be stored at a rack location, while there is space in the second one to store two loads (ROODBERGEN; VIS, 2009).

Roodbergen and Vis (2009) also discuss the operation and design decisions for the system. The system can operate in single command cycle or in dual command cycle, in the first case the crane cycle refers to one storage action or one retrieval order per time, different from the second one in which the crane executes both a storage and a retrieval process in one cycle. Besides the decisions of various options showed in Figure 6, when developing the automated storage and retrieval system to be built, there are many other parameters to be defined about the design of the system, as showed in Table 1. There are many studies applying simulation models to evaluate the impact of some of these parameters on system operation or on its improvement, these researches are discussed in section 0.

Table 1: Desing decisions for AS/RS

Class of problem	**Decisions to be made**
System configuration	• Number of aisles; • Height of the storage racks; • Length of the aisles; • Equally sized or modular storage locations; • Number and location of the I/O-points; • Buffer capacity at the I/O-points; • Number of cranes per aisle
Storage assignment	• Storage assignment method; • Number of storage classes; • Positioning of the storage classes
Batching	• Type of batching; • Batch size; • Selection rule for assignment of orders to batches;
Sequencing	• Sequencing restrictions; • Scheduling approach; • Sequencing method
Dwell-point	• Type of positioning; • Location where idle cranes will be placed

Source: Roodbergen and Vis (2009)

2.3. Modeling automated storage and retrieval systems performing picking processes

There are two methods to model warehouse systems: the analytical one and the simulation based. They differ in relation to accuracy and time required to completion. Modeling using simulation results in more accurate and realistic outcomes, but it requires much time to conceptualize and design a detailed and accurate model. With respect to the analytical based models, they run faster, and the estimated performance measures are acceptable, even though they are not as accurate as the ones obtained using simulation. Consequently, when several configurations need to be analyzed, the analytical technique is preferred, at least in the first phases, to reduce the number of scenarios (AZADEH et al., 2017).

Several studies have been developed to analyze or improve warehouses operation, they apply different methods and tools, and encompass various issues related to the warehouse performance. Among the problems explored, there are storage design optimization, operation analysis and performance evaluation under various scenarios, as showed in the literature review done by Azadeh et al. (2017). This study focused on researches about automated storage and retrieval systems as Eurofork aims to evaluate a new shuttle for this kind of system. Güller and Hegmanns (2014) states that "Especially, automated storage and retrieval systems are difficult to model analytically because they incorporate interactions of many subsystems.", the authors indicate simulation as an important tool to analyze this type of system.

Ekren and Heragu (2009) use simulation to help defining the design, they highlight the importance of defining the correct design for autonomous vehicle storage and retrieval systems in the first time, since it is hard to change it later as consequence of its lack of flexibility. Taking it into account, the authors developed a regression analysis in order to define the best combination for the rack configuration, varying the number of aisles, tiers and bays to optimize the systems operation. To do so, they developed a simulation model of the system on Arena 12.0 and collected the average cycle time of 21 scenarios simulated. Then, the researchers were able to do regression analysis with the data collected to establish a function relating the number of tiers, aisles and bays with the average time for the storage and retrieval processes and optimized this function to define the best values for each variable.

Ning at al. (2016) also applied simulation model to support the system design definition. To find the best rack design, they developed an efficient simulation model that can be reconfigured varying the number of racks and elevators. The authors simulated 81 different rack configurations under 15 different retrieval rates within 48 hours, evaluating the throughput and cycle time of each case. As simulation is very time consuming, the model was created using eM-simulation package, due to its faster simulation process and simulation modelling, and object-oriented technique, to accelerate the process. A modified constant velocity was applied taking into consideration the acceleration/deceleration effect in the simulation. This allowed a faster collection of more accurate results. A chart was built with the simulations' results, the outcome was an optimal curve with all the optimal solutions under all the retrieval rates.

In a study developed by Lerher, Ekren and Sari (2015) not only the impact of different rack configurations (number of tiers and aisles) were taken into account, but also the influence of velocity profiles of the shuttle carriers and of the elevator's lifting tables on the performance of a shuttle bases storage and retrieval system performance. A discrete event simulation was used to realize the performance evaluation by obtaining the number of transactions (throughput) occurred in timespan on each scenario. The cases were created considering 3 values of tiers and of aisles and two different velocity profiles for the shuttle carrier and the lifting table. The authors concluded that the system's performance (throughput) is linked to the multiplication of two factors: the number of aisles and the throughput of the elevator. In addition, they verified that even though the bottlenecks are the elevators' lifting tables in general, when the number of columns is too high, the shuttle carries can be the bottleneck depending on their velocity.

The design of experiments done by Lerher (2017) applied simulation to analyze the impact of the shuttle-based storage and retrieval system's characteristics on its throughput in order to discover the optimal design too. The factors considered were number of columns, velocity of the shuttle carrier and of the elevator's lifting table, acceleration/deceleration of the shuttle carrier and of the elevator's lifting table. The simulation was modeled using the Visual Basic for Application program language from Excel and the elevator's lifting table and shuttle carrier's movements were simulated considering the real velocity-time dependence. Through the study, the author found the options of design that the system shows the best performances.

Another study to define the best design using simulation was developed by Ekren and Heragu (2011), the difference in this case is that they search for the optimal combination of lifts and autonomous vehicles (AVs) considering some pre-defined rack configurations. The simulation analysis was done using Arena 12.0, considering seven rack arrangements, two arrival rates and for each rate there were 9 or 10 possible combinations for the number of lifts and vehicles per zone. The data collected from the simulation to evaluate the system's performance were the average cycle time of storage and retrieval transactions and the average utilization of AVs and lifts. Through the simulation's results, the authors were able to comprehend the system's performance under several scenarios and discover the best combination.

The best value of lifts and vehicles for pre-defined scenario was also developed by Ekren et al. (2010). However, in this study the objective was to analyze the impact of several variables and their interaction on the system's performance. For this reason, first a simulation was developed to find the best combination of the number of lifts and vehicles, then a design of experiments was carried to understand the factor's influence on the system and finally the best value for each variable was defined through Turkey's test analysis. The authors evaluate the influence of four factors (dwell point policy, scheduling rule, input/output location and interleaving rule) on three performance measures (average cycle time for storage and retrieval transactions, average vehicle utilization, and average lift utilization) under different arrival rates.

Another type of study in the literature that uses simulation to improve the process is the evaluation of the system's performance under different circumstances. Güller and Hegmanns (2014) applied a discrete event and multi-agent simulation model of a mini-load multi-shuttle order picking system to verify how it performs when the order structure varies. In order to do the evaluation, data about system's throughput and average cycle time per order were collected and analyzed. Bruno and D'antonio (2018) employed simulation and analytical tools to analyze the warehouse's performance under various scenarios of the manufacturing operation. The evaluation in this case was made considering the system utilization, average time spent in queue by the unit loads and average number of queued unit loads.

Besides the application of simulation to define better designs or improve the process, this method was also used by Marchet et al. (2012) to analyze the performance of AVS/RS for product tote with tier captive configuration. The analytical model developed by the author was validated with an Arena simulation of the system and used to estimate the transaction cycle time and waiting time. As indicated by the authors, this model is useful during the "conceptualization" phase of the warehouse design, since it allows to obtain the performance measures in a short time.

Simulation was a tool also applied to compare different solutions in the literature. Zhou and Mao (2010) developed an analytical model to optimize the storage location assignment considering the energy consumption to load and unload the units. Then, the solution found by the analytical model and the original storage configuration were modeled and simulated on Flexsim software, obtaining data about the stay time, congestion time, unloading quantity and no-load time. The comparison between both results showed that the optimized one was indeed more efficient.

2.4. Flexible final product composition

A rainbow or mixed pallet is a pallet constituted by multiple products in order to attend a single customer (SHULTZ, 2003 in MONGOLD and JOHNSON, 2006). The objective of offering this kind of pallet is to better attend clients that need to order smaller quantities than a full pallet. According to Andel (1998 in YAMAN and ŞEN, 2008), at that time there was already a trend of ordering uniform pallets to attend a certain period to more frequent request of mixed pallets to supply the same interval. This might happen in the cases in which customers want to order more frequently or because the demand of a certain item is too low to order a full pallet (HAMMOND, 1999 in MONGOLD and JOHNSON, 2006; YAMAN and ŞEN, 2008). Besides that, mixed pallet can also help manufacturers in the penetration of small size markets (BARRESE, 2002 in YAMAN and ŞEN, 2008).

Yaman and Şen (2008) helped a beverage producer to solve the problem of the need to build mixed pallets. In this case, the producer was selling many brands, some of them not very popular, for this reason the demands of certain products were not high enough to order full pallets of them. Due to producer's constraints to offer customized pallets, the solution found was to sell a few standard mixed pallets. In this way, the customers could select an option that would better fit their demand. This solution proved to be efficient, it resulted in savings of inventory and backlogging costs compared to the alternative of just selling full pallets.

In the study developed by Yaman and Şen (2008), there were some constraints that turned the option of building customized pallets to each client not feasible; according to the authors there was lack of technology and the manual operation would be too complicated. As discussed in the article, the automation of the operation requires high investment and the execution of it manually. Besides, being labor intensive can result in more costs due to damages and mistakes. These factors can discourage warehouses and distributor centers to execute operations to build mixed pallets, even though it might be beneficial.

Besides the mixed pallets usage increase due its flexibility, there is also a trend of consumers doing more complex and frequent orders (WHITEPAPER, 2016). In this way, handling and shipping case and split cases instead of pallets are becoming more usual in distribution centers. According to the Annual Warehouse and Distribution Center Operations Survey results, the percentage of respondents that deal with a mix of full pallet, case and split case on the inbound grew from 4% in 2015 to 48% in 2016 and when specifically looking at the flow involving split cases, the percentage of respondents in 2016 that handles it on the inbound is 58% and on the outbound is 69% (SUPPLY CHAIN MANAGEMENT REVIEW, 2016).

Considering the exposed, Eurofork's project aims to attend better the warehouses facing an increase of mixed-pallet demand. To do so, its developing technology, an automated storage and retrieval system, is expected to build mixed-pallets more efficiently than the existing technologies.

3. INNOVATIVE ORDER PICKING SYSTEMS

The main purpose of this thesis, as mentioned before, is to evaluate and compare the performance of two different automated storage and retrieval systems, one of them in the development phase by Eurofork. As a step of Eurofork's project, the company needs to define the structure and features of the new shuttle used in the system. Considering this, the company asked for some technologies insights as part of this thesis study, as it can help to define the shuttle model and be considered in the simulation model. This section presents the information collected about innovative order picking systems that can build mixed pallets.

The research about automated order picking systems was conducted searching about innovative picking methods and picking methods for flexible products on Google Scholar, Google Patents and database of scientific research, such as Web of Knowledge website. In the following, the results found are presented in the following section (3.1). In section 3.2 , the insights from the research are discussed.

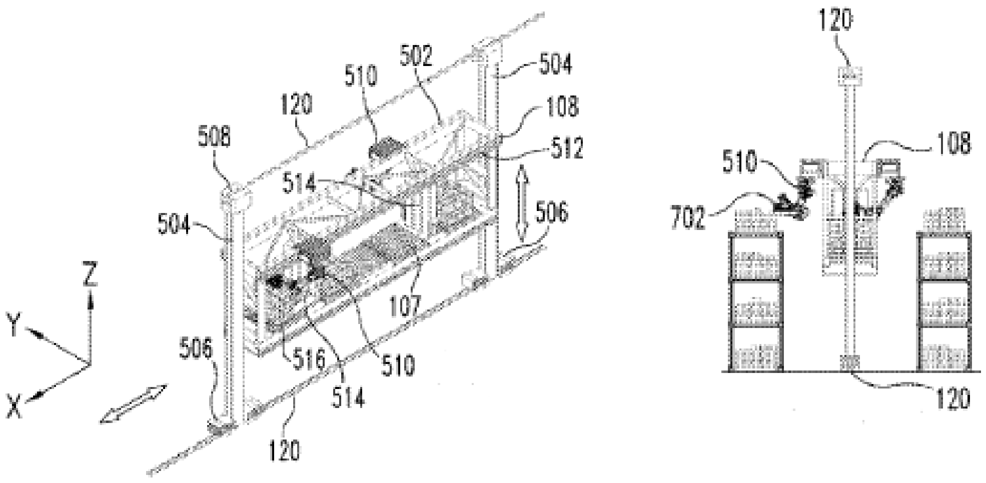
3.1. Research results

1. Robotic automated storage and retrieval system mixed pallet build system by Bastian Solutions

Bastian Solutions, a Toyota Advanced Logistics company, specialized in material handling equipment patented in 2014 a robotic automated storage and retrieval system to build mixed pallets. This is an automated storage and retrieval system (AS/RS) that consists in a carriage that can move horizontally and vertically, the carriage contains robotic arms to execute the picking operation. Besides that, the carriage has space to transport more than one pallet at the same time. The carriage can be observed in Figure 7 and Figure 8.

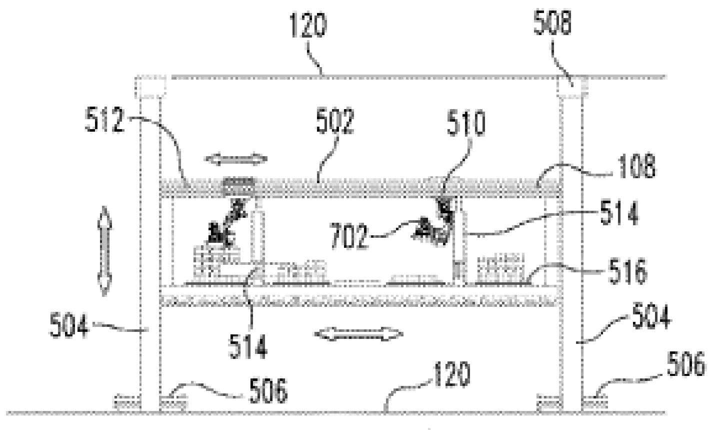
The system builds the pallets while the picking operation is still being executed, the robotic arm positions the product picked in the mixed pallets transport by the carriage. Besides that, it can also be applied to the inverse operation, pick products from the mixed pallet and put it back in the rack. According to the patent documentation, this technology saves work as it does not require to break down the pallets, buffers and sequence each product. In addition to it, it also reduces the required floor footprint, because the warehouse does not need space designated to break the full pallets and build the mixed ones, and it is more cost competitive. (BASTIAN AND CALLOWAY, 2014)

Figure 7:Perspective view (left) and end view (right) of the robot AS/ RS



Source: Bastian, II et al. (2014)

Figure 8: Lateral view of the AS/RS.



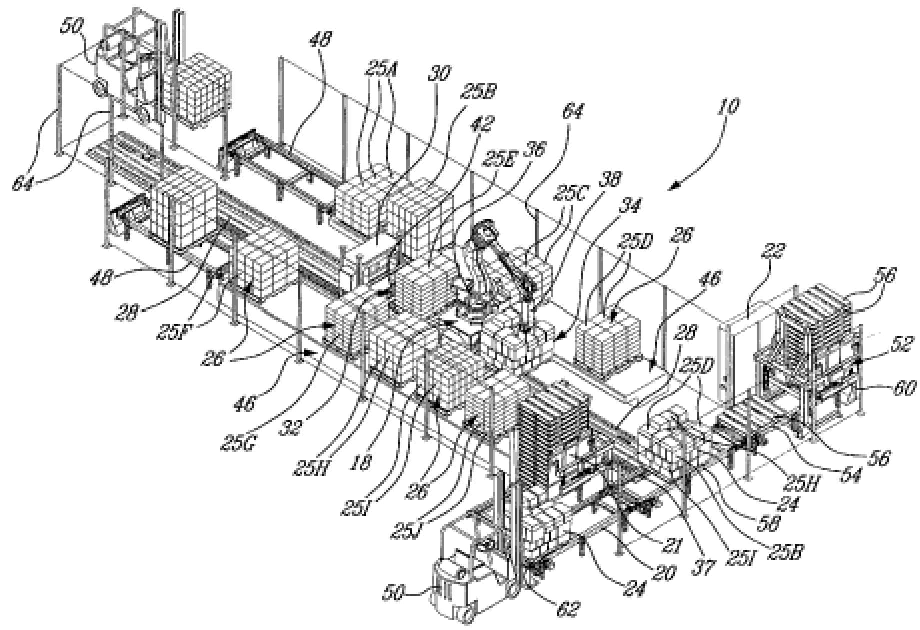
Source: Bastian, II et al. (2014)

Furthermore, as reported by the patent authors, in order to give more flexibility to this system, there are different tools for the robotic arm that can be used interchangeably, for instance vacuum gripper or shrink wrap roll. Besides that, the carriage also contains turntables to make the wrapping process easier and to facilitate the allocation of the products onto the pallets. At last, the robotic arm also includes depth sensors and vision system to support the picking process.

2. Robot in a pallet support shuttle by Axium Inc.

This system patented by Axium Inc. is formed by a shuttle that can carry the pallet and contains a robot able to pick products from full pallets placed in adjacent stations. The system illustrated in Figure 9 is less flexible than the previous one presented, considering it can only move horizontally along the path. Consequently, the warehouse still must retrieve the pallets and put them along the feed conveyors located closed to the shuttle’s path. Yet, the robot allows to pick items of different sizes from the pallets and different tools can be used, such as vacuum, side grippers or fork tool, giving flexibility to the process. (MORENCY et al., 2008)

Figure 9:Picking robot in a shuttle system



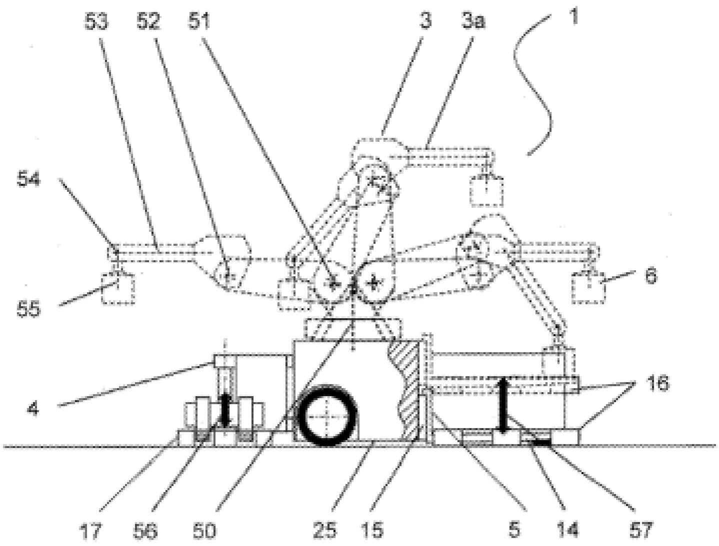
Source: Morency et al. (2008)

3. Vehicle with robot device, by Jungheinrich

This autonomous order picker patented by the German intralogistics provider Jungheinrich in 2017 is composed by a vehicle with robotic arm, a pick-up tool and a carrying device. The robotic arm picks a product from a full package and position it in the carrying device in order to build a mixed pallet. It is also similar to the first system described previously, but in this case the vehicle moves around the racks and the pick-up device is used to take the pallet from the rack and return it back to its position after all the items ordered were collected. It can also execute the inverse operation, get the product from the pallet carried in the vehicle and put it onto the pallet storage. (MAGENS et al., 2017)

The vehicle described is displayed in Figure 10.The pick-up device, identified as number 4 in Figure 10, is an interesting feature of this order picker as it allows the operation without requiring free space above the pallet stored in order to enable the robotic arm to get the item. Besides this, the vehicle can also move vertically, using a lifting device, and horizontally, so this system saves space in both directions.

Figure 10:Vehicle patented with the robotic arm in different positions



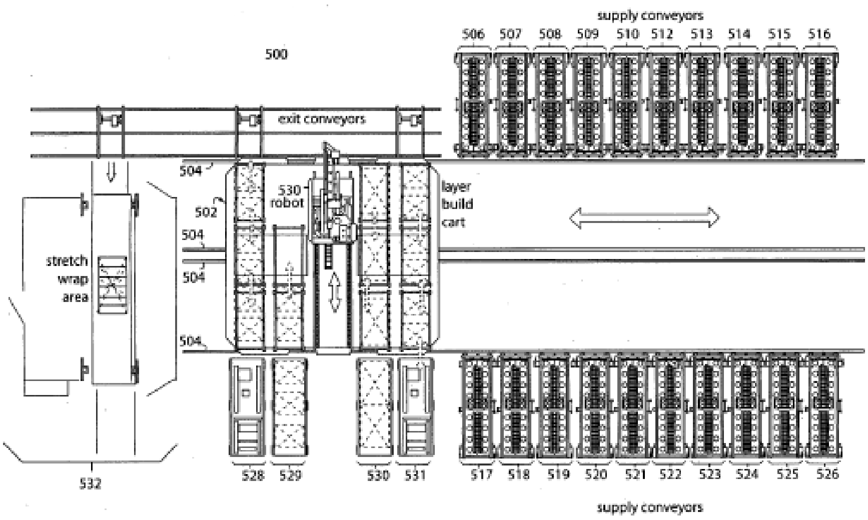
Source: Magens et al. (2017)

4. Layer build cart with robot

This patented system to build rainbow pallets consists in a cart with a robot that pick layers from homogeneous pallets and put it in the new rainbow pallets that are being built. The difference in relation to the previous systems described is that the cart has space for many pallets, so it can be supplied with full pallets and build more than one mixed pallet at the same time, optimizing the process. Besides the flexibility due to the fact that it allows to build mixed pallets, the authors of the patent also state that it presents other advantages, it incurs in lower costs and higher accuracy. (MITCHELL, 2010)

The system is illustrated in Figure 11, it can be observed the empty spaces for the pallets with an “x” in the cart indicated by the number 502. These empty cells are supplied by the conveyors or filled with the mixed pallets being built by the robot indicated by the number 30. When the order is completed, the pallets are released by the exit conveyors and goes to the wrapping area, as showed in Figure 11.

Figure 11: System of a layer build cart with robot

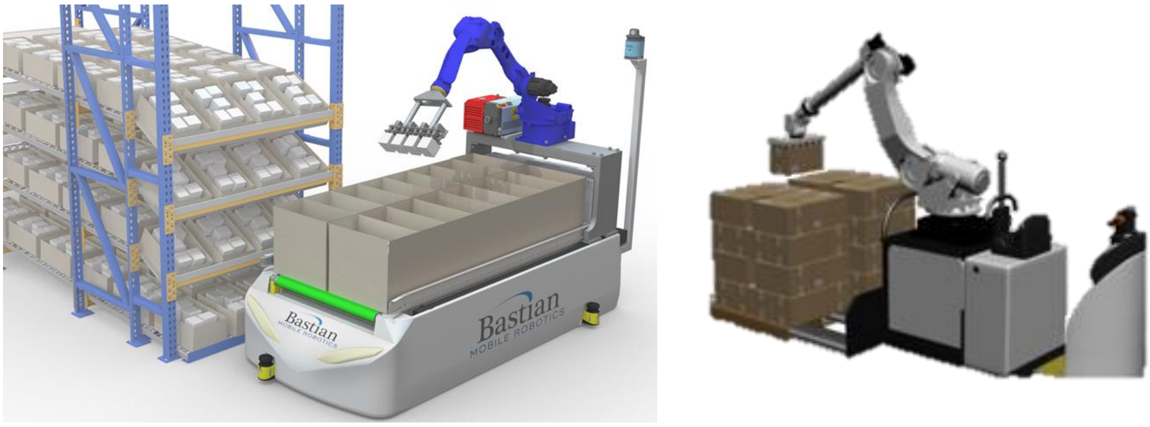


Source: Mitchell (2010)

5. Mobile robotic arms for picking process

There are a series of mobile robot options in order to support the picking process of mixed orders in the market. The robotic batch picking, and the robot palletized mixed pallet produced by Bastian Solutions are some examples. Both robots aim to reduce the time and work required to build a pallet by not demanding the pallet to be retrieved and taken to the build area, to do so the picked items are placed on the vehicle and transported along the process until its completion, as can be seen in Figure 12. Besides that, similar systems as the one displayed in Figure 13 are also on alternative on the market, the difference of this system compared to the previous ones is that the items are only transported from one pallet to another, so there is no need to retrieval the whole pallet but the robot have to return every time to the place where the mixed pallet is located.

Figure 12: Robotic batch picking (left and AGV with robotic arm (right)



Source: Bastian Solutions (2019)

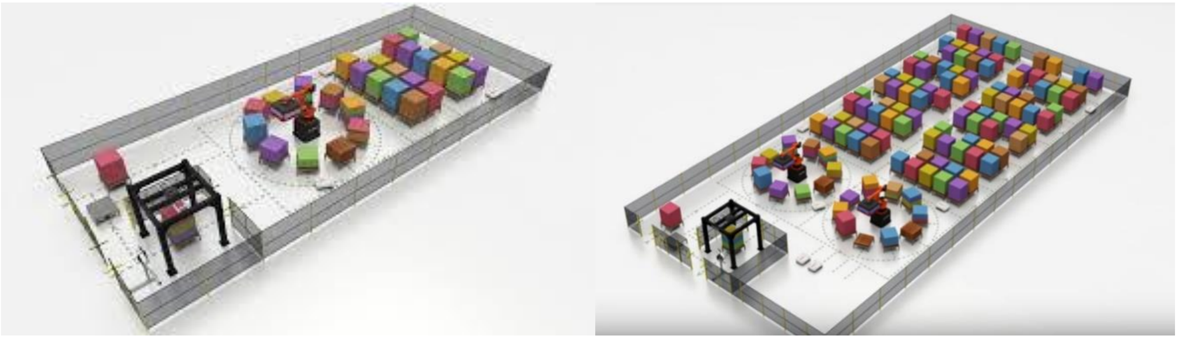
Figure 13: Robotic palletizer mixed pallet

Source: Bastian Solutions (2019)

6. CarryStar, Swisslog

CarryStar is an automated order fulfillment system developed by Swisslog, a swiss company specialized in logistics automation. In this system there is a storage area for the pallets and the build pallets process occurs in a separated zone. The space organization is illustrated in Figure 14, the pallets received by the system are storage in the back of the building and the build pallets region is the circle sector located between the storage and the entrance. The area designated to build the mixed pallets can be observed with more details in Figure 15.

Figure 14: CarryStar simple structure (left) and expanded one (right)



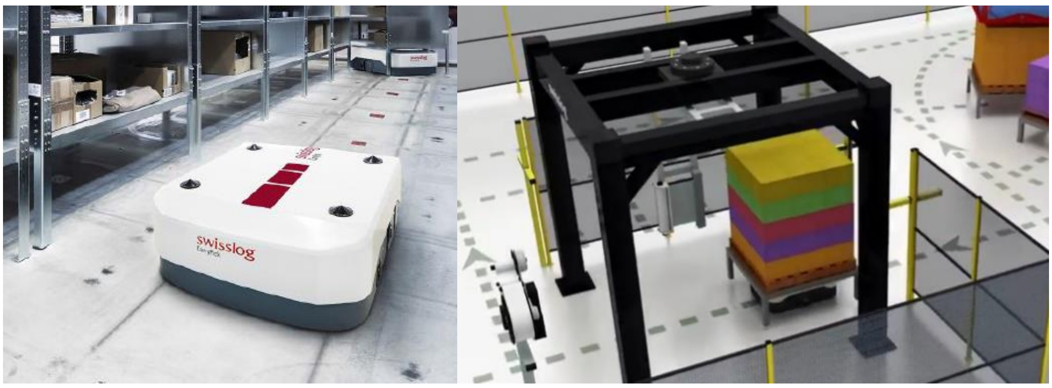
Source: Swisslog (2019)

Besides these areas cited previously, the system is also constituted by Carry robots, automated guided vehicles that transport the pallets around the area, and the wrapping machine. The Carry robots move around following the paths previously defined, as shown in Figure 16. The wrapping machine is located at the exit of the system, the robots move the pallets that are ready to this apparatus where they are wrapped and then taken to be shipped. The robot and the wrapping machine can be seen in Figure 16.

Figure 15: Build pallet area

Source: Swisslog (2019)

Figure 16: Carry AGV (left) and Carry AGV taking the mixed pallet to the wrapping area (right)



Source: Swisslog (2019)

According to MHD (2018), a magazine specialized in transport and logistics news, this system presents several advantages. First, it is scalable and modular, as showed in Figure 14, consequently the system can be used in operation of different magnitudes (small, mid-size and large). Besides this, it also can be expanded easily because it requires minimal fixed infrastructure. In addition, this system is also advantageous because of its flexibility, the technology allows to fulfill flexible orders efficiently, the robot can pick around 200-300 layers per hour to build the mixed pallets. Another flexible feature is that the robots can be defined depending on volume and type of products the company works with.

Another advantage of the CarryStar is the decrease in the time and work needed to complete the picking process. The CarryStar cell is connected to the warehouse storage and receives the pallets from it, but it also has a minor storage closer to the area where the pallets are built. In this way, after the ordered items are picked from a pallet, it does not have to be retrieved all the way back to its original position, it can be storage in the CarryStar cell until the next request.

In addition to that, since the area has the minor storage that are supplied with pallets from the warehouse storage, it does not interfere in the storage organization and operation. Consequently, the company is able to use better the space available, vertically and horizontally, even though the Carry robot does not pick pallets above the ground level.

7. TORU, Magazino

TORU is a picking robot, showed in Figure 17, it is able to pick boxes with its vacuum gripper, store them and then transport them to a workstation and can to the inverse operation. According to the producer, it can pick objects in a range from 8 up to 250 centimeters above the ground and store up to 16 products at the same time (MAGAZINO, 2019). In order to be more reliable, it has also sensors to allow the surrounding comprehension and take it in consideration when making decisions.

The system is flexible considering the warehouse can use as many robots as needed, reducing the fleet it when the demand is low. This allows to save costs in infrastructure, since it can be easily integrated if there is plans of expansion in the future, for example.

Figure 17: TORU robot picking an item (left) and the robot with the storage filled with products (right)

Source: Magazino (2019)

8. Skypod

Like CarryStar robots, presented before, Skypod are small robots that carry the products from the storage to the build pallet station. An important feature of this technology is that it is similar to the AVS/RS operation, but it does not require any complex structure to carry the robots, as lifts, because it moves horizontally and vertically by itself, as shown in Figure 18. This system is advantageous because of its flexibility, the number of robots operating can change easily depending on the work demand without requiring structure modifications.

Figure 18: Skypod robot (left) and skypod robot picking a product in a high tier (right)

Source: Exotec (2019)

3.2. Insights

After the research about innovative order picking systems, the advantages and disadvantages of each of them was evaluated in order to support the Eurofork's project to verify the opportunities for the company's project. With this information, the company can verify if there is a customer need not being satisfied by existing systems and also what other companies are doing that was not considered for the Eurofork's new system. The advantages and disadvantages are presented in Table 2.

Table 2: Advantages and disadvantages of different innovative order picking systems

Technology	Advantages	Disadvantages
1	• Moves horizontally and vertically; • Builds mixed pallet during the picking process; • Different tools for the robotic arm; • Multiple pallets at the same time	• Size of the carriage; • Robotic arm external to the carriage; • Requires a wide aisle; • One carriage per aisle.
2	• Different tools for the robotic arm; • Picks products of different sizes; • Build many pallets at the same time.	• Does not use the vertical space, cannot pick product in higher tiers; • Products must be retrieved first.
3	• Requires less space above the pallet for the robotic arm movement; • More flexibility in relation the robot movement, can move among aisles; • Occupies less space in the aisle; • Can use several in one system, according to demand.	• Moves vertically, but a small distance, so does not use the vertical space; • Build one pallet per time.
4	• Build multiple pallets; • Connected to the wrapping area.	• Very large, requires a wide aisle; • Need to retrieval products from storage.
5	• There is an intermediary storage • Connected to wrapping process; • Build multiple pallets at the same time; • Flexible, intermediary storage area and number of robotic arm can be increase or decrease; • Does not interfere in the storage operation.	• Needs to retrieve items first; • Requires large area for the intermediary storage; • Does not use vertical space for intermediary storage because the robot only picks pallets on the ground level.
6	• Flexible to move around the storage area; • Can increase the number of robots; • Robotic arm system to picks items from the storage.	• Does not build pallet, only store them and transfer to another area; • Limited space to store items; • Does not use vertical space, only pick products up to 250 cm from the ground.
7	• Picks products from storage, not the full pallet; • Flexile movement, can move around the aisles.	• Limited space; • Does not use vertical space.
8	• Flexibility, can increase the number of robots; • Moves horizontally and vertically	• Carry small items; • Does not build orders while picking.

Source: The author

Considering the technologies analyzed, their characteristics, advantages and disadvantages, it can be concluded that the companies are starting to focus on developing flexible technologies in order to attend the demand of mixed pallets or unit loads.

Besides that, it was observed that the technologies 1 to 4 present similarities with Eurofork's shuttle concept of picking products from the pallet at the storage locations. However, none of the vehicles were developed to be used in storages of high density, since they occupy a large space and does not travel to higher levels. Some insights from these technologies are related to vehicle's capacity, as most of them can build and transport more than one pallet at a time. Eurofork could develop a shuttle that builds and transports more than one pallet, for example, in order to improve the system's performance.

Moreover, the fourth technology described presents a device that could be useful for the new system. Considering the little space in the storage rack, the robotic arm will not have much space for its movement, what can limit the operation and consequently increase the robot picking time. As an alternative, the shuttle could remove the pallet from the rack location as the vehicle described in item 4 does with the pick-up device. In this way it would have more space above the pallet, since it would be in the aisle, facilitating the picking activity. However, this process would also impact the picking activity duration, as the pallet would be removed and then restored to its original location, so an evaluated would be necessary to verify it this procedure would be more beneficial than prejudicial to the whole process.

In relation to the Carry Star technology, this system is similar to the current AVS/RS considered (storage + palletizing area), but it contains an intermediary storage to speed up the process, as the pallets with ordered SKUs are close to the robotic arm. The downside of this system is the area required to the intermediary system, that could be used for storage. Considering this, Eurofork's system might be a better option for warehouses in which there is not much space available, but for warehouses with more space the intermediary system could be an option. Therefore, in the future it is important to evaluate and compare both systems operations, as this alternative might be applied in the warehouses as a solution to build mixed pallets.

The interesting insight from the sixth technology describe, Toru robot, is related to the arm robot tools. This robot can pick boxes in different positions, if it moved a little in the rack, for example, and is able to get a box from a lower level without removing the boxes that are above it. Since the robotic arm are going to pick items from one pallet inside the storage rack, these features could improve the picking process and consequently the systems performance.

Finally, with respect to the Skypod technology, the insight is related to the possibility of building a smaller shuttle also with the robotic arm attached. This vehicle, with the same concept of the Eurofork's system, could be applied in warehouses in which the orders are unit loads. In this case, it would be necessary to destack the pallet in order to store unit load neither to retrieve pallets from the storage since the robotic arm would pick the requested items.

Through the analysis of innovative picking orders systems, it was verified many opportunities for the Eurofork's systems, as it fulfills existing gaps of other technologies. Besides that, some features from the technologies observed could be applied by the company in order to improve the vehicle's configuration and consequently the system operation.

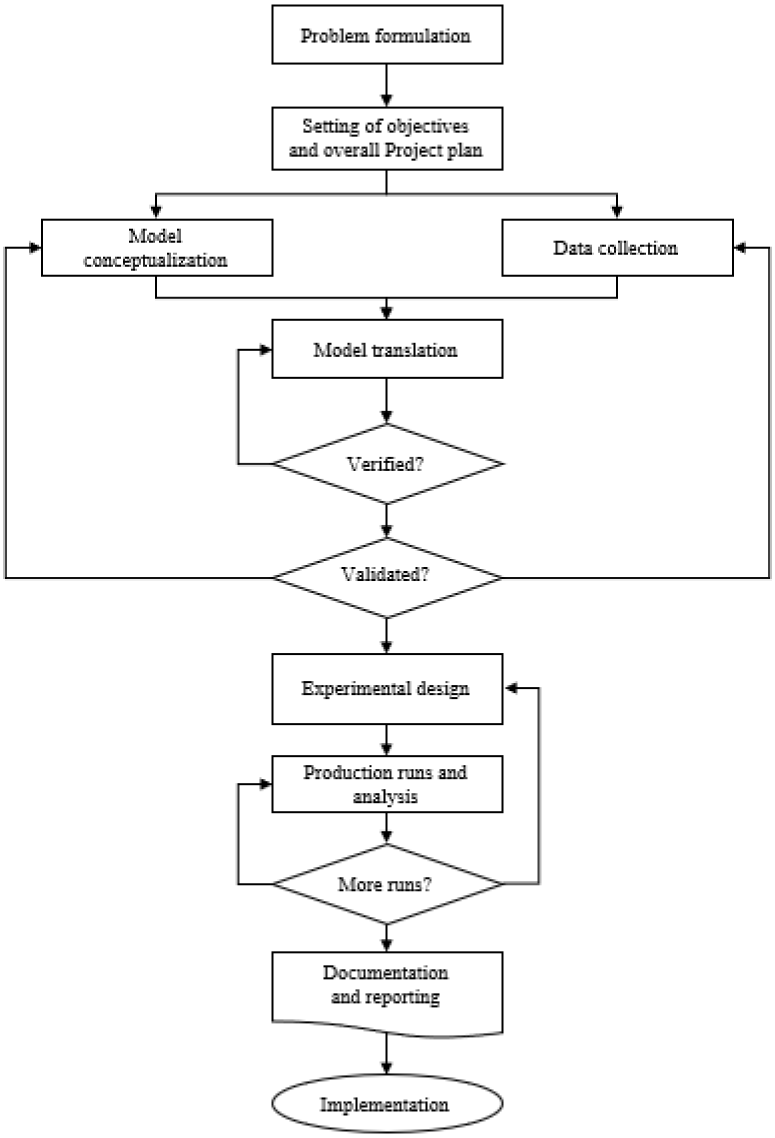
4. METHODOLOGY

In order to compare the two autonomous vehicle storage and retrieval systems – the current one and the new one - this thesis is going to apply the methodology of a simulation study proposed by Banks et al. (2004), showed in Figure 19. The steps to be followed are: problem formulation, setting of objectives and overall project plan, model conceptualization, data collection, model translation, model verification and validation, experimental design, production runs and analysis, documentation and reporting and finally implementation. Since this study aims to evaluate the performance of the new system, this thesis finishes with the analysis and recommendations. In the next sections each step of this simulation study is developed.

The problem formulation has already been discussed in section 1.3 (Statement of the problem), the objectives also have already been defined in section 1.4 (Objectives). In chapter, both systems' operations and simulation models are explained, as well as the model conceptualization. The current system is discussed in section 5.1 and Eurofork' system in section 5.2. The model verification and validation are discussed in section 5.3, the experimental runs results and analyses are presented in chapter 6. Besides that, since this study aims to only compare the performance, it is not going to be implemented.

In order to analyze the simulation results, a factorial experimental with replications will be executed. It was defned in experimental design 3 factors to be evaluated, each factor with 3 levels. For each scenario simulated 10 replications were run, totalizing 270 simulations. Besides that, 4 different performance measures were collected: cycle time, orders completed (throughput capacity), vehicle and lift utilization. Then, an analysis of variation (ANOVA) was executed with the simulation's outputs, using the software Minitab. This method allowed the verification of which factors has significant effect on the system's performance and also the evaluation of which system performed better under the scenarios' conditions.

Figure 19:Steps in a simulation study



Source: Banks et al. (2010)

5. MODEL CONCEPTUALIZATION AND TRANSLATION

A simulation of a virtual warehouse will be built to evaluate and compare the performance of two different types of automated storage and retrieval systems. The study purpose is similar to the one developed by Mital (1992), in which the author compared two methods to build pallets, manually by operators or automated by an industrial robot, through experimental simulation.

Mital (1992) compared the performance of these two alternatives considering the cycle time to execute the process of building the mixed pallet and also the cost of each method. Different from Mital (1992), in both systems considered in this thesis – current and new systems - the palletizing activity is performed by robotic arms, as mentioned before, and an economic evaluation was not feasible due to the lack of financial data as the technology is still in the development phase.

Besides that, the lack of data also impacted the simulation, since there is no data to compute the time required to execute the operation of transferring items from one pallet to another. In this way, while Mital (1992) evaluated the time required to put all ordered items into a pallet, in this thesis four different values were set as transferring time for both systems. In this way, it is possible to evaluate this factor impacts on the system's efficiency and enable the estimation and comparison of both systems' performance depending on the time required by each. Nevertheless, due the difference of distance and space between the robotic arm and the pallets, the transferring interval is different and should be evaluated when the exact data is available.

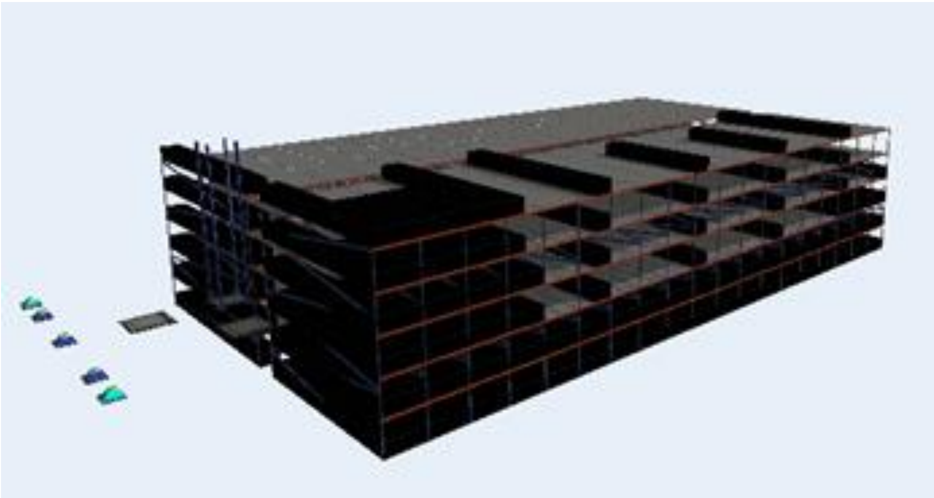
5.1. Current AVS/RS

5.1.1. Components and system

In the current AVS/RS system considered, following the classification of AS/RS systems developed by Roodbergen and Vis (2009) displayed in Figure 6 (chapter 1), the system is shuttle-based with single command, end-of aisle picking, the loads are pallets and the rack is stationary and multiple deep. Besides that, system is tier-to-tier, the shuttle can move from one tier to another using the lift. The system is composed by a storage rack, as the one showed in To facilitating the comprehension of the current system, a simplified scheme of the operation to build a mixed pallet is illustrated in .

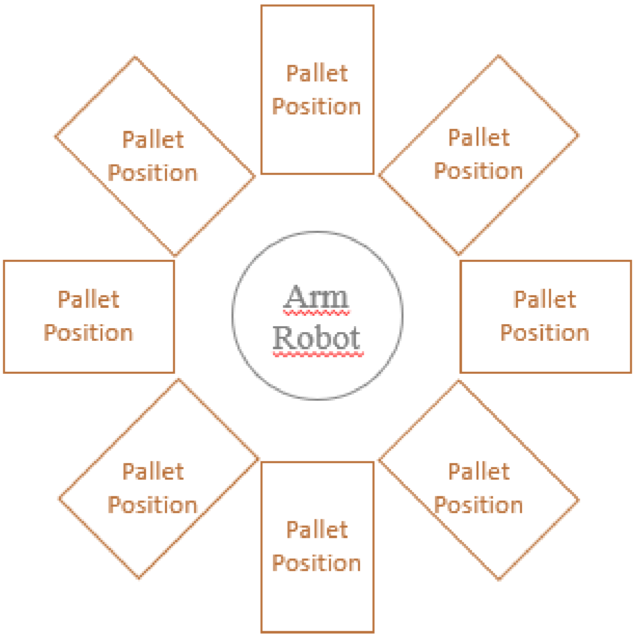
Figure 20; the stored pallets, each containing one single SKU; shuttles, such as the one presented in Figure 4 (chapter 1); one lift, to move the shuttle between tiers; and the palletizing area with the robotic arm and slots for pallets, retrieved and mixed ones, as the example displayed in Figure 21. To facilitating the comprehension of the current system, a simplified scheme of the operation to build a mixed pallet is illustrated in .

Figure 20: Storage rack



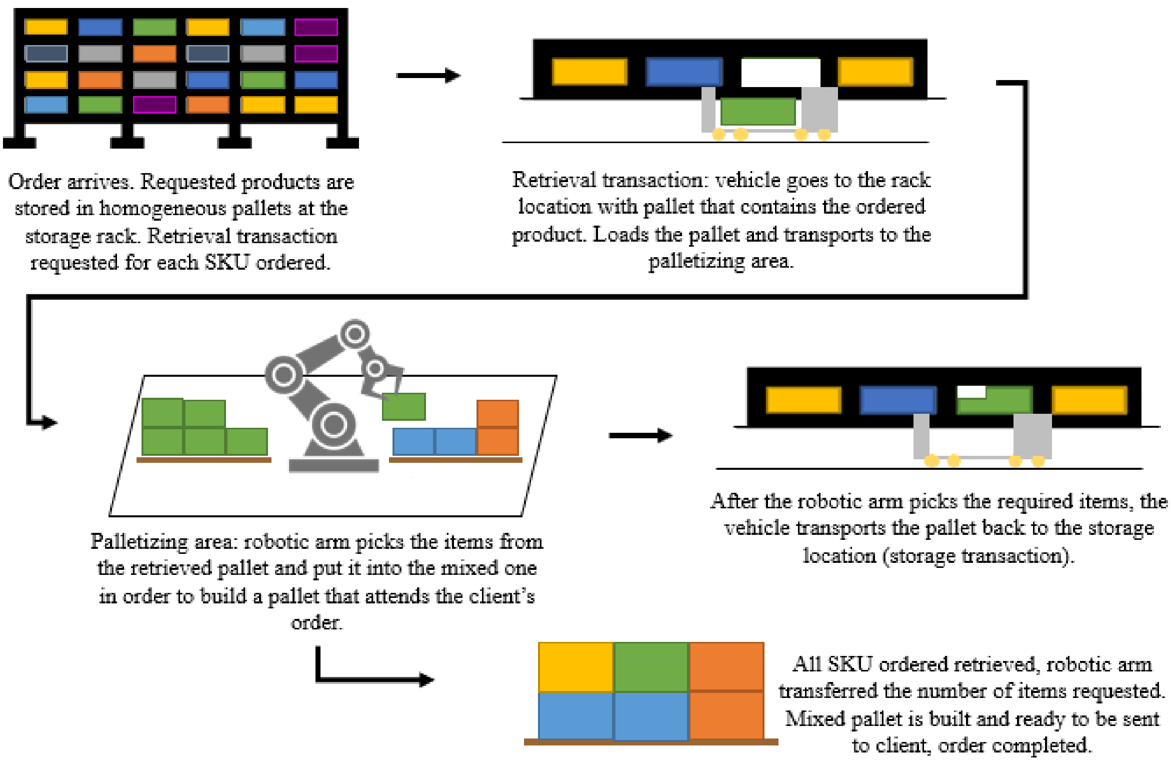
Source: Eurofork

Figure 21: Example of a palletizing area layout (top view)



Source: the author

Figure 22: Simplified scheme of the current system's operation



Source: the author

The storage rack is fulfilled with homogeneous pallets that come from outside the system. In the model it is assumed that the storage is always full, and the storage transactions are not considered, as done by Marchet et al. (2012). According to the authors, since retrieval process is a critical activity, the storage transaction can be postponed and completed when the system presents less workload. Moreover, a random storage policy was defined when the system is initialized, after the initial definition, it is assumed a dedicated storage, so each product is stored in its fixed position until the end of the simulation. In the case of the current system, every time a pallet is retrieved, a new one of the same type is allocated at its rack position.

Every time an order arrives on the system, a retrieval transaction is created for each SKU ordered. For each transaction, a shuttle is requested to pick the pallet from its rack location and to transport it to the palletizing area. If the vehicle is idle, it remains at its dwell- point, that in the model was assumed to be outside the storage rack and close to the lift entrance at the ground floor. When requested, the vehicle solicits the lift, at the moment the lift is available at the ground floor, the shuttle goes into it and it is transported by the elevator to the tier where the pallet to be retrieved is. At the tier, the vehicle leaves the lift and moves to the channel where the pallet is, picks it and finally does the inverse path, requesting the lift to go down and discharging the pallet in the palletizing area after it leaves the lift at the ground floor.

During the shuttle process explained, there are three different transferring moments: the shuttle going inside/leaving the lift, shuttle loading/unloading the pallet and shuttle movement from the elevator to the palletizing area. For the simulation model, due to lack of data from a real system, the time required to the shuttle go into/leave the lift and to load the pallet from the rack location was taken from the literature. Ekren and Heragu (2011) defined the time required to shuttle enter or leave the lift as equal to zero and the to load/unload the pallet into the shuttle as 14 seconds.

With respect to the interval during which the load is transported the palletizing area after the vehicle unloads it in front of the lift's entrance, for this project it was assumed as equal to zero. In fact, this value depends on the configuration of the warehouse and the technology used to do the transportation, for example conveyors. However, since the pallet transportation from the lift's entrance to the palletizing area is independent of the vehicle travelling process to retrieve items from the storage, it was assumed that the real interval would not be high enough to impact the systems performance. Therefore, it is reasonable to assume the interval as zero, since it does not interfere in the simulation results.

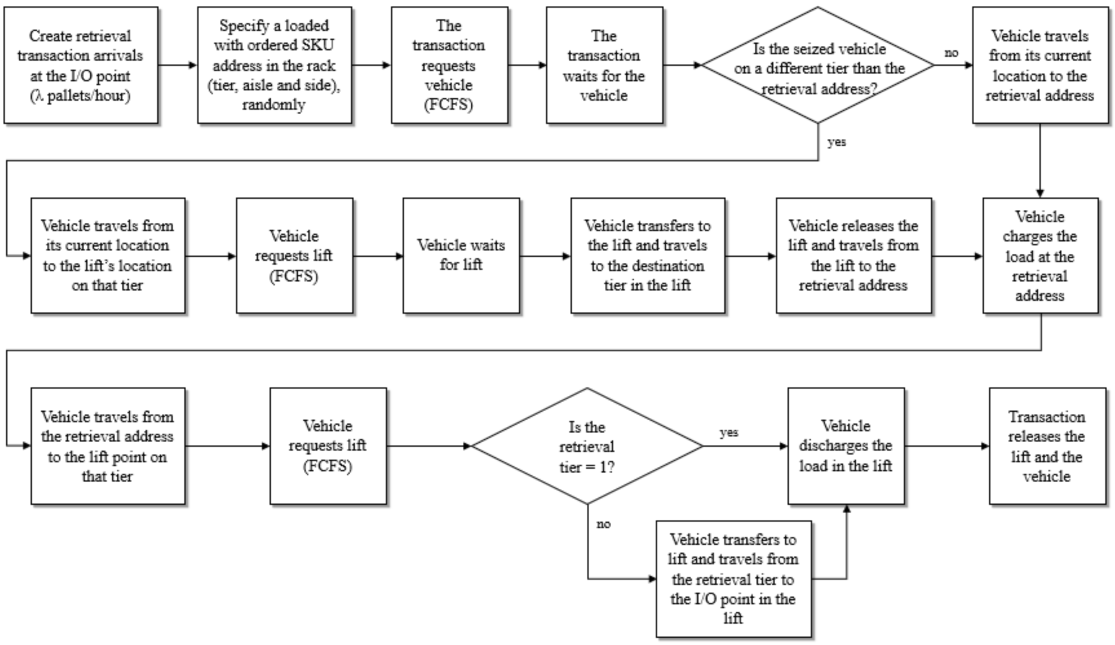
In relation to the lift, in the model was considered one single lift per aisle. The lift process, as described before, is the transportation of the shuttle up and downward. In the model after the shuttle leaves the lift, the latter it remains at the same tier until a new request arrives.

Finally, the last component of the system, the palletizing area, is the zone where the robotic arm builds the ordered pallet (as illustrated in Figure 5 in chapter 1) , by picking the ordered items from the retrieved pallet and putting into the pallet that is going to be shipped to the client. In this area, besides the robotic arm, there are the slots where the pallets, mixed and retrieved, are allocated. In the example showed in Figure 21 the slots are positioned as a circle around the robotic arm, but as showed in the chapter 3, there are many possible configurations for this zone. Moreover, in the simulation model it was considered one single palletizing area with different slots depending on the scenario, but this number can also vary, as verified in the CarryStar systems described in chapter 3.

5.1.2. Model

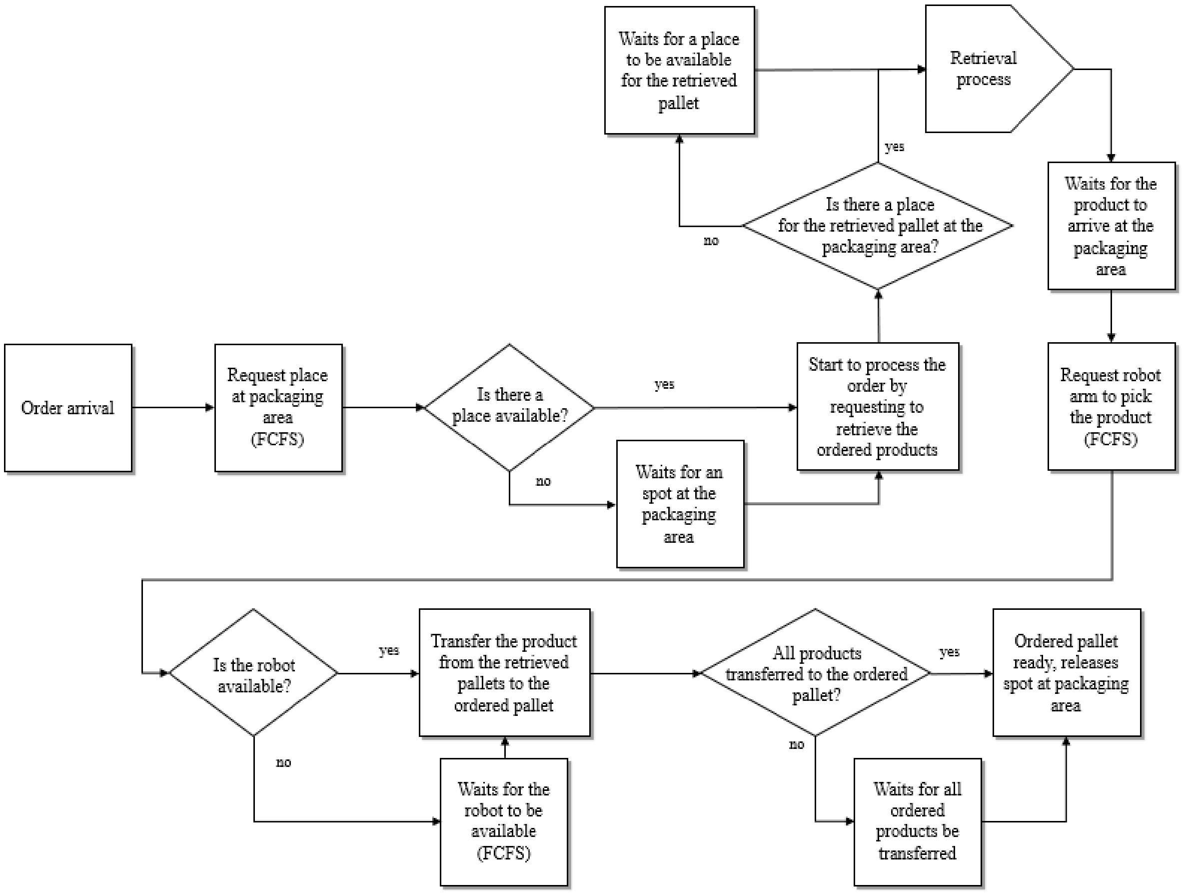
According to Tappia et al. (2019), "little research is available modeling the interaction between the storage and order pick systems". Considering this, in order to model the system and processes described, the simulation developed for this thesis followed the simulation flow charts for retrieval transaction from Ekren and Heragu (2011). Some adaptations were made in the model due the palletizing process, since the authors simulated only the storage and retrieval transactions, were made considering the simulation flow chart of the simulation developed by Tappia et al. (2019) that models a shuttle-based storage and retrieval system (SBS/RS) integrated to an automated parts-to-picker order picking system through conveyors. In Figure 23 it is displayed the retrieval transaction flow chart, while the flow chart presented in Figure 24 illustrates the process from the moment the order arrives until the pallet is completed with all items requested.

Figure 23: Flow chart of the retrieval transaction in the current AVS/RS



Source: The author, adapted from Ekren and Heragu (2011)

Figure 24: Flowchart of the model for the current AVS/RS operation



Source: the author

The model developed in Simpy is composed by four main functions that are related to the processes of the simulation: Arrival, Palletizing Area, Retrieval and Build Pallet. Besides that, there are other five auxiliary functions: Resources, Storage, Order, Travel Time Vehicle, Travel Time Lift, Record Data. Besides that, there are 5 different resources: vehicles, lift, robot, slots for mixed pallet and slots for retrieved pallet. The code for this simulation model is presented in the Appendix section.

When the simulation is initialized, first it is defined which SKU is stored in each position of the storage rack by the function Storage. For this, it was assumed that the probability of an SKU be produced and stored is equiprobable. So, the function receives as arguments the quantity of SKUs available in the storage and the probabilities of each one to be ordered. Then, for each position, it is selected randomly which SKU is going to be stored. As a result, the function returns a list with the storage rack information, the SKU and the quantity of items stored, in case of system 2 it is always the quantity of a full pallet. In Figure 25 it is illustrated how the function is called and what it returns for a storage rack with A aisles, T tiers and C columns.

Figure 25: Example of Storage function being called and its output

```
storage = Storage(QTY_SKUS, PROB_SKUS):

print storage

storage = [[Aisle 1, Tier 1, Column 1, Side 1, SKU x, #Items],
           [Aisle 1, Tier 1, Column 1, Side 2, SKU y, #Items],
           [Aisle 1, Tier 1, Column 2, Side 1, SKU z, #Items],
            .
            .
            .
           [Aisle A, Tier T, Column C, Side 1, SKU s, #Items],
           [Aisle A, Tier T, Column C, Side 2, SKU t, #Items]]
```

Source: the author

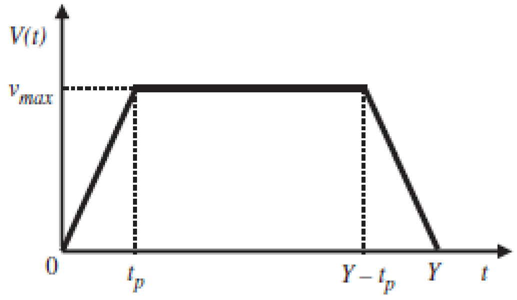
After the storage is defined, the Arrival function is called. This function creates the arrival of the orders, following Poisson distribution with a pre-defined arrival rate for the scenario, and defines the ordered quantity of each SKU available in the storage, calling the auxiliary function Order. Then, it is defined from which rack locations the pallets with the ordered SKU are going to be retrieved. In the end of the Arrival function, the Palletizing Area one is called receiving as argument a list with all rack positions with pallets to be retrieved and the quantity to be picked from each pallet.

The Palletizing Area function starts requesting the resource “slot for mixed pallet”. Then, when it is available, for each SKU ordered it is requested the resource “slot for retrieved pallet” and with this resource available the retrieval transaction order is created, calling the Retrieval function and sending as argument the rack location where the pallet to be retrieved is.

The Retrieval function follows all the steps illustrated in the flow chart display in Figure 23. It begins requesting the resource vehicle, when the resource “vehicle” is available it starts the process already described, requesting the resource “lift” and travelling to the tier and channel of the rack location, loading the pallet and then returning to the first tier and leaving the lift. For each lift process, the vehicle waits for the resource to be available and the resource “lift” is only released after the vehicle leaves it. In relation to the resource “vehicle” it is released in the end of the process when the load is discharged at the palletizing area.

The travel time for each lift and vehicle movement is computed by the auxiliary functions Travel Time Lift and Travel Time Vehicle. To compute the travel time, first is calculated the distance based on the resources’ initial and final position (tier and column). Then, it is calculated the time to travel the distance considering the acceleration and deceleration impact on the velocity through time, illustrated in the velocity profile displayed in Figure 26 and described in equation (1), as done by Marchet et al. (2012), Lerher (2017) and Lerher et al. (2015) in their models.

Figure 26: Equipment velocity profile



Source: Marchet et al. (2012)

$$V(t) = \begin{cases} a_h t & if\ t \le t_p \\ v_h & if\ t_p \le t \le Y - t_p \\ -a_h t & if\ t \ge t_p \end{cases} \tag{1}$$

Since this velocity profile was considered, there is a travel time equation if the travel distance the equipment is going to cover is shorter than the interval required to the equipment reach its maximum velocity and then decelerate to velocity zero and another equation if it is longer, as showed in equations (2) for vehicles and (3) for the lift. These equations are used in the model to compute the travel time of the vehicle ($TV_{v,c}$) and lift ($TV_{l,t}$) movements.

$$TV_{v,c} = \begin{cases} 2\left[\frac{2v_v}{a_v} + \frac{D_{v,c} - 2\left(\frac{v_v{}^2}{2\,a_v}\right)}{v_v}\right] & if\ Y > 2v_v/a_v \\ 2\left[2\sqrt{\frac{D_{v,c}}{a_h}}\right] & if\ Y < 2v_v/a_v \end{cases} \tag{2}$$

$$TV_{l,t} = \begin{cases} 2\left[\frac{2v_l}{a_l} + \frac{D_{l,t} - 2\left(\frac{v_l{}^2}{2\,a_l}\right)}{v_l}\right] & if\ Y > 2v_l/a_l \\ 2\left[2\sqrt{\frac{D_{l,t}}{a_l}}\right] & if\ Y < 2v_l/a_l \end{cases} \tag{3}$$

After the retrieval process is completed, the pallet is transferred to the palletizing area, vehicle is released, and function Build Pallet is called. This function requests the resource "robot", it executes the robotic arm process of picking the quantity ordered from the retrieved pallet and putting into the mixed one and then release the resource. After all items from the retrieved pallet are picked, the resource "slot for retrieved pallet" is released and if the mixed pallet is ready (with all items ordered) the resource "slot for mixed pallet" is also released.

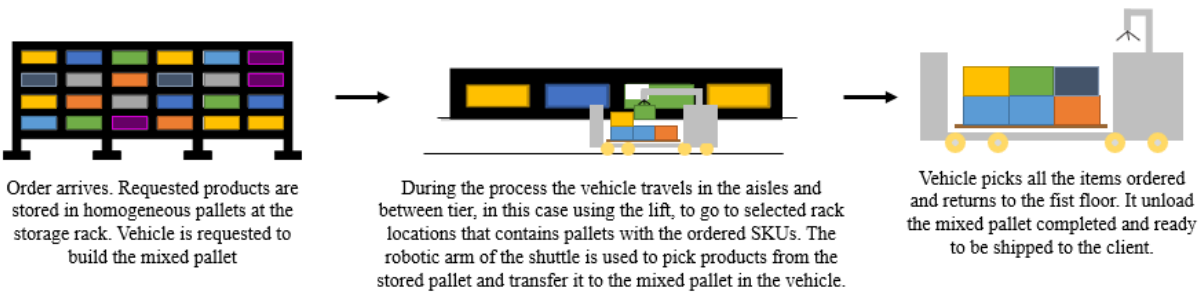
Finally, after the simulation finishes, the function Record Data is called. This function records all information collected about vehicle and lifts processes and the duration of each order transactions in tables in an Excel file. Examples of the tables with the data are showed in the Appendix, the data for each process was used to do the model verification and the data about the order transactions were used to compute the system's performance indicators for each simulation. For each scenario, ten replications were executed, then an average was computed to obtain the system's performance indicators for the scenario. Six indicators were considered: the cycle time from the moment the order arrives, the cycle time from the vehicle started the process, the waiting time for the vehicle, lift and vehicle utilization and the total of orders completed.

5.2. New system

5.2.1. Components and operation

The Eurofork's system, as the current system, can be classified as shuttle-based system with single-command, pallets as loads and the stationary, multiple-deep rack and tier-to-tier, considering the categories considered by Roodbergen and Vis (2009), displayed in Figure 6. However, in relation to the picking process it cannot be classified into any category defined by the authors. In this system, the picking and palletizing process occur at the same time in the storage rack, therefore the item is going to be picked at the storage location as person-on-board, but the activity is performed by the robotic arm not by an operator. The scheme in Figure 27 illustrates the picking and palletizing operation in the new system.

Figure 27: Simplified scheme of the new systems operation



Source: the author

With respect to the system's composition, since it does not need a palletizing area to execute the process of transferring products from one pallet to another, the system is composed only by the storage rack, lift and shuttles. The storage rack is the same as the one in the current system, displayed in To facilitating the comprehension of the current system, a simplified scheme of the operation to build a mixed pallet is illustrated in .

Figure 20, and the system also contains only one lift per aisle. In relation to the shuttle, as explained before, there is a robotic arm attached to it, as showed in the example illustrated in Figure 3.

Regarding the storage rack, as considered for the current system model, it is randomly fulfilled in the beginning of the simulation and it remains with the same SKU in each location through the entire time. Also, the storage transaction is not included, but it is assumed that the products are always available in the storage, as done by Marchet et al. (2012). The difference in this system with respect to the quantity of items in each pallet is that, while in the current every position contained a full pallet the entire time, in this system the quantity is updated as items are picked from the pallet. When the quantity reaches the value of zero, it is set again to the number of items available in a full pallet, because it is considered that a new pallet occupied the position.

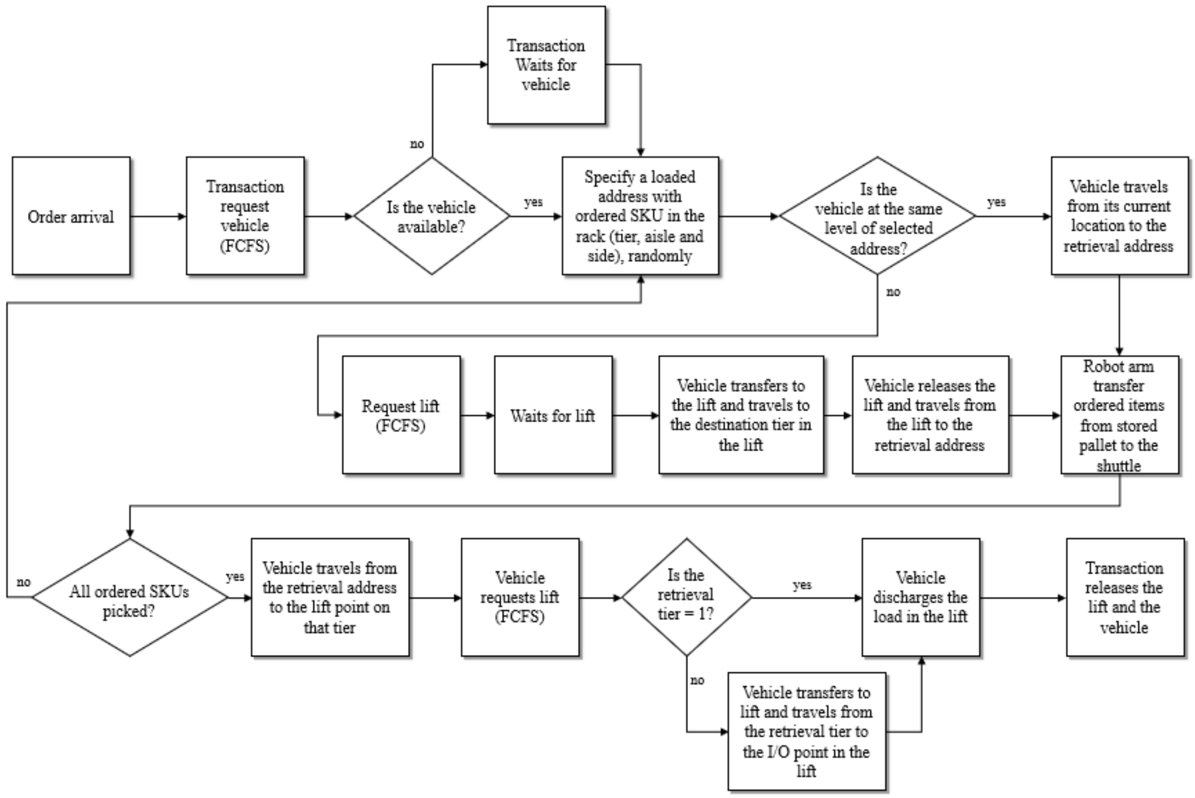
The system operation is similar to the retrieval transaction from the current system considered, but instead of loading the full pallet, the robotic arm picks only the quantity requested and puts it into the pallet being built in the vehicle. Furthermore, during the process the vehicle can go to more than one rack location, depending on the order. The shuttle travels inside the storage rack from selected rack location to another that contains the ordered SKUs, picking the items until the ordered pallet in the vehicle is finished. During the travels from one place to another, if the products to be picked are in different tiers, the vehicle requests and uses the lift to move upward or downward.

5.2.2. Model

The flow chart for simulation of the new system is showed in Figure 28. As explained before, the system is still being developed and it was not verified in the literature a study about a system that presents the same concept. Considering this and the similarities between the Eurofork's system operation and the retrieval transaction process in the current system, the simulation flow chart for this case was also developed by adapting the simulation flow chart from Ekren and Heragu (2011).

The model developed in Simpy is composed by only two main functions that are related to the processes of the simulation: Arrival and Picking. Besides that, the five auxiliary functions used in the current system model are also presented in the model for new system, they are: Resources, Storage, Order, Travel Time Vehicle, Travel Time Lift, Record Data. Additionally, 2 resources are defined in this model: vehicles and lift. The code for the simulation model can be verified in the Appendix.

Figure 28: Flowchart of the new system



Source: the author

The beginning of the simulation model for New system is the same as the one in the current system model. It begins with the creation of the storage rack, then the Arrival function is called, the orders are created, and the rack locations with the pallet from which the items are going to be picked are selected. However, in the end of the arrival function a new function is called, the Picking one.

The Picking function starts by requesting the resource “vehicle” and the cycle will begin only when it is available. This cycle is related to vehicle and lift movements inside the storage, besides the process of transferring items at each selected location. Moreover, the cycle occurs until all selected rack locations were visited by the vehicle, which means that all items were collected, and then resource “vehicle” is released.

After the simulation is completed, the function Record Data is also called and records the same type of information in an Excel file it occurs after the current system simulations. The same number of replications were set for the simulation of this system and likewise the current system simulation, six performance indicators (the cycle time from the moment the order arrives, the cycle time from when the vehicle started the process, the waiting time for the vehicle, lift and vehicle utilization and the total of orders completed) were collected in order to enable the comparison between both systems.

5.3. Models verification and validation

In the literature analyzed, usually the authors validate their model with the results from other studies that also developed a model for the similar system or validate the analytical model by comparing the outputs with the results from a simulation model developed in software, such as Arena (MARCHET et al., 2012; MANZINI et al., 2016). Due to the lack of similar systems in the literature, it was not possible to compare the results with another model. However, a series of tests were executed in order to verify the model operation.

Since the system initializes with the storage creation, the first test executed was to verify with all the rack locations were being considered and if the assignment of SKUs to the positions was following the probability distribution. To check if all rack locations were considered, it was tested if the length of the list with the storage positions was equal to the number of rack locations expected. The quantity of racks should be equal to the multiplication of the quantity of aisles, tiers, columns and sides. In relation to the distribution of SKUs, it was computed how many pallets of each SKU was stored and then the correspondent percentage of this value in relation to the total of pallets in the storage to certify that it was according to the probability defined. Since the SKUs were randomly assigned, the values in each replication was not exactly the expected but the average considering all replications was. Besides these tests, some small storages were created and list with storage locations was analyzed by the author in order to verify if there was any problem.

After the storage creation, orders are created. In order to verify if orders were created as expected, the quantity of each SKU ordered is exported to the Excel file, as well as other simulation data. Then, it was computed the average number of times each product was requested and then this value was compared to the total so it could be checked if the percentage was close to the probability defined.

Subsequentially, resources are requested in each system. In order to verify the vehicles and lift usage, data from all process that used them were exported and recorded in an Excel file. Using this data, first it was verified if the time the process started was equal or higher than the moment when the previous processes finished. An example of this verification considering the simulation outputs for all lift processes in New system with 1 vehicle is illustrated in Table 3.

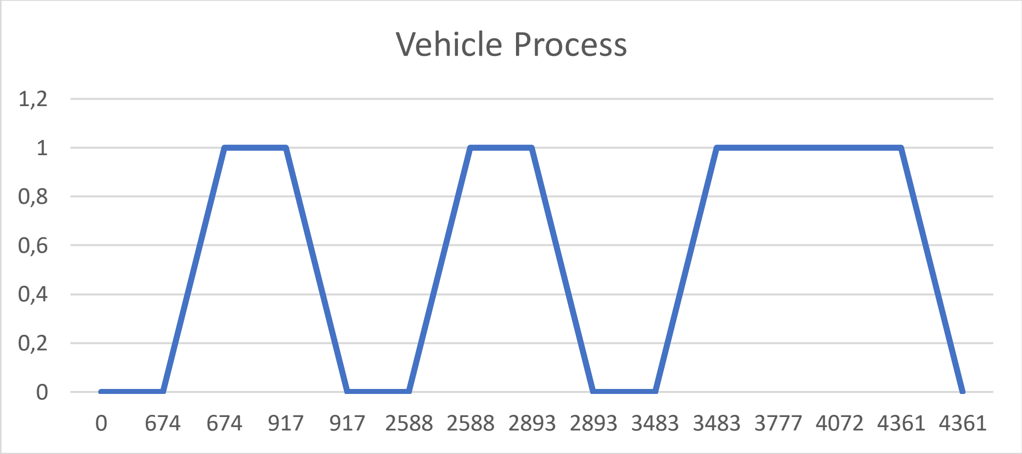
Table 3: Example of a verification of processes sequence

				#Falses -	0
Order	Resource	Time_I	Time_F	Duration	Time I >= Time F(-1)?
1	Vehicle	674.4284	916.8774	242.449	
2	Vehicle	2587.862	2893.046	305.1841	True
3	Vehicle	3483.382	3777.356	293.9745	True
4	Vehicle	3777.356	4072.222	294.866	True
5	Vehicle	4072.222	4361.433	289.2112	True
6	Vehicle	5101.859	5374.602	272.7427	True
7	Vehicle	5535.636	5778.489	242.8529	True
8	Vehicle	7173.942	7472.302	298.36	True

Source: the author

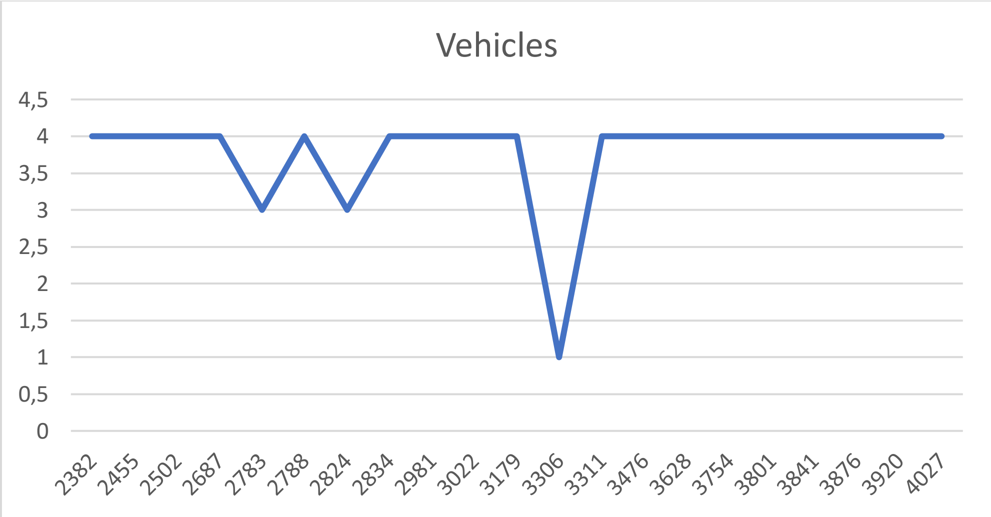
After this initial teste, the number of resources executing a process during intervals of the simulation was computed and compared if the value varied through time from 0 up to the value defined for the scenario. The charts displayed in Figure 29 and Figure 30 show examples of the verification made, for the scenario with one vehicle the values are equal to 0 when the vehicle is idle or to 1 when it is executing the process and for the scenario with 4 vehicles the values fluctuates between 0 and 4 .

Figure 29: Quantity of vehicles through time, scenario with 1 vehicle



Source: the author

Figure 30: Quantity of vehicles through time, scenario with 4 vehicles



Source: the author

One final teste was executed with respect to the interval required by vehicle and lift to move from one place to another. The travel time function was verified by exporting the initial and final resource's position besides the duration of the process in the simulation. Then, travel time computation was verified by calculating manually the expected travel time for some processes chosen randomly and comparing the both values.

All performed tests indicated that the models were working as expected. Therefore, they were used to simulate and evaluate both systems' performances, as described in the following sections.

6. DESING OF EXPERIMENTS - RESULTS AND ANALYSIS

After the simulation model for both systems were developed, the experimental design was proceeded. In order to evaluate the systems performance and the impact of different factor on it, a full factorial experiment was executed. The experiment considered three different factors, three levels for each factor, totalizing 27 different scenarios, and 10 replications for each configuration, resulting in 270 simulations for each system.

The scenarios as well as the data used as fixed input for all simulations are presented in section 6.1. Then, in section 6.2 the simulation results obtained for each scenario are displayed and discussed.

6.1. Scenarios

In order to compare both systems evaluated, the simulation models described in chapter 4 were run for the scenarios defined for the full factorial experiment, considering only one pallet being built per time. The scenarios were defined changing three different factors with three levels each, they are: the number of SKUs available in the warehouse (5, 10, 20), the orders arrival rate (2, 7, 15), the robotic arm picking time (2, 7, 12). The scenarios considered are presented in Table 4.

Table 4: Scenarios data for comparison between the systems

Scenario	Quantity of SKUs	Order Arrival Rate Per Hour	Robotic arm picking time [s]	Scenario	Quantity of SKUs	Order Arrival Rate per Hour	Robotic arm picking time [s]
1	5	2	2	15	10	7	12
2	5	2	7	16	10	15	2
3	5	2	12	17	10	15	7
4	5	7	2	18	10	15	12
5	5	7	7	19	20	2	2
6	5	7	12	20	20	2	7
7	5	15	2	21	20	2	12
8	5	15	7	22	20	7	2
9	5	15	12	23	20	7	7
10	10	2	2	24	20	7	12
11	10	2	7	25	20	15	2
12	10	2	12	26	20	15	7
13	10	7	2	27	20	15	12
14	10	7	7				

Source: the author

In addition to these scenarios, it was also carried simulations varying the quantity of shuttles available in the system to evaluate the impact of this variable in both systems' performance. For these simulations, the value set for quantity of SKUs was 10, for orders arrival rate was 50 and robotic arm picking time, 5, and the number of vehicles varied from 1 up to 4. In the case of the current AVS/RS system, since the number of slots also impacts the system's performance, the quantity of slots for retrieved pallets is set to be equal to the number of vehicles, while the number of slots for mixed pallets remained equal to one. The scenarios described are showed in Table 5.

Table 5: Scenarios defined to evaluate the impact of changing the quantity of vehicles in the system

Scenario	Number of shuttles operating	#Slots for mixed pallets (current system)	Quantity of SKUs	Order Arrival Rate Per Hour	Robotic Arm Picking Time
28	1	2	10	50	5
29	2	2	10	50	5
30	3	3	10	50	5
31	4	4	10	50	5

Source: the author

Besides the values defined for each factor in the scenarios defined in Table 4 and Table 5, some variables related to the system configuration were fixed for all simulations of both systems. These fixed variables are:

- A, the number of aisles in the storage rack;
- L, number of levels in the storage rack;
- C, the number of channels in each side of the aisle;
- LA, the number of lifts per aisle;
- I, number of items per pallet;
- μ_w, the width of each channel in the storage rack;
- μ_h, the height of each level in the storage rack;
- v_v, a_v , the vehicle's velocity and acceleration;
- v_l, a_l, the lift's velocity and acceleration;
- T_{rs}, time to transfer the load from the rack to the shuttle;
- T_{sl}, time to shuttles go into/leave the lift.

The values set for each variable are showed in Table 6. All values were defined by Eurofork, apart from the time to transfer the load from the rack to the shuttle (T_{rs}) and time to shuttles go into/leave the lift (T_{sl}), that was taken from Ekren and Heragu (2011).

Table 6: Data for scenarios analysis

Variable	Unit of measure	Value
A	-	1
L	-	6
C	-	19
LA	-	1
I	-	20
μ_w	m	1.3
μ_h	m	2
v_v	m/min	120
v_l	m/min	14
a_v	m/s²	0.5
a_l	m/s²	0.3
T_{rs}	s	14
T_{sl}	s	0

Source: the author

6.2. Factorial experiment results

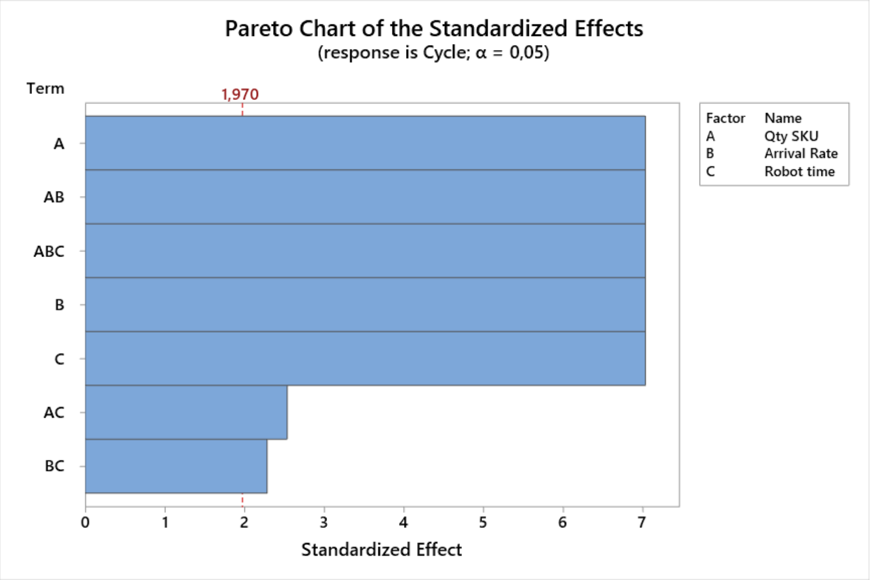
The tables with the results collected from the 270 simulations of each system executed for the factorial experiment are presented in the Appendix. As mentioned before, 4 different performance measures were collected, they are: the cycle time of the process (from the moment the order starts to be built until it is completed), quantity of orders completed, lift and vehicle utilization. The results were obtained by simulating each scenario for 51 hours, being 3 of them the warm-up period.

After the data collection, it was carried out an analysis of variance to identify the factors that impacts the performance, the facilitate the reading the effects of each factor are displayed in Pareto charts. Then, the results were plotted in charts to show the main effects of each factor in the systems performance. Finally, a boxplot chart was created with the data of completed orders in each scenario. In this section the results for each system are presented and discussed.

6.2.1. Current system

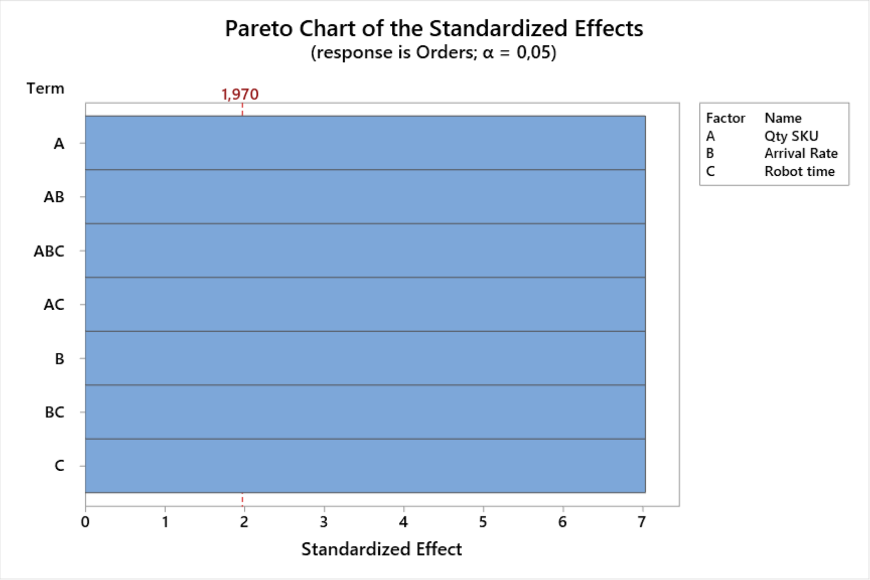
The standardized effects of each factor on the current system's performance is illustrated in the Paretos charts displayed in Figure 31, in relation to the cycle time, Figure 32, regarding the throughput capacity, Figure 33, in relation to the vehicle utilization and Figure 34, about the lift utilization. Considering a confidence interval of 95%, the current system's performance is significantly impacted by all three factors considered in the analysis and by their interaction too. Thus, these factors are important to evaluate the system's performance.

Figure 31:Standardized effects of the factors on the current system's cycle time



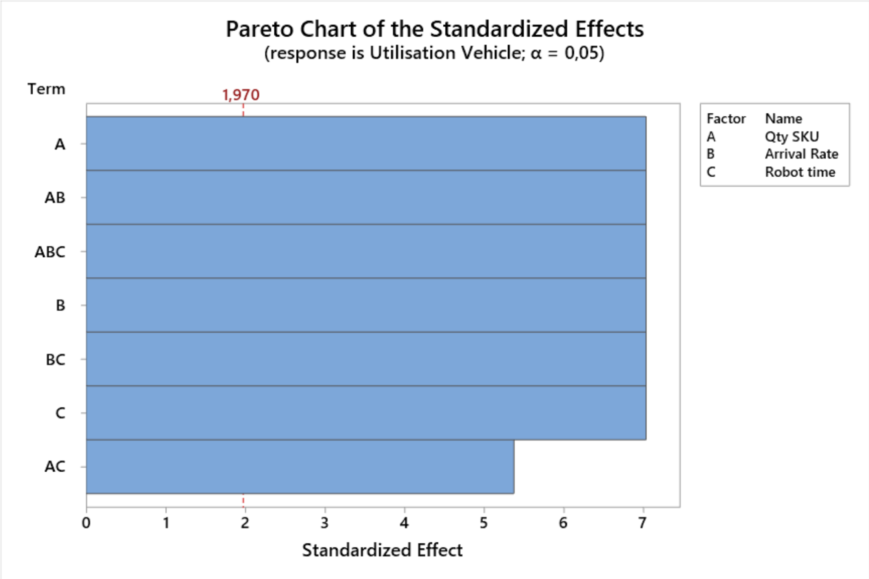
Source: the author

Figure 32: Standardized effects of the factors on the current system's throughput capacity



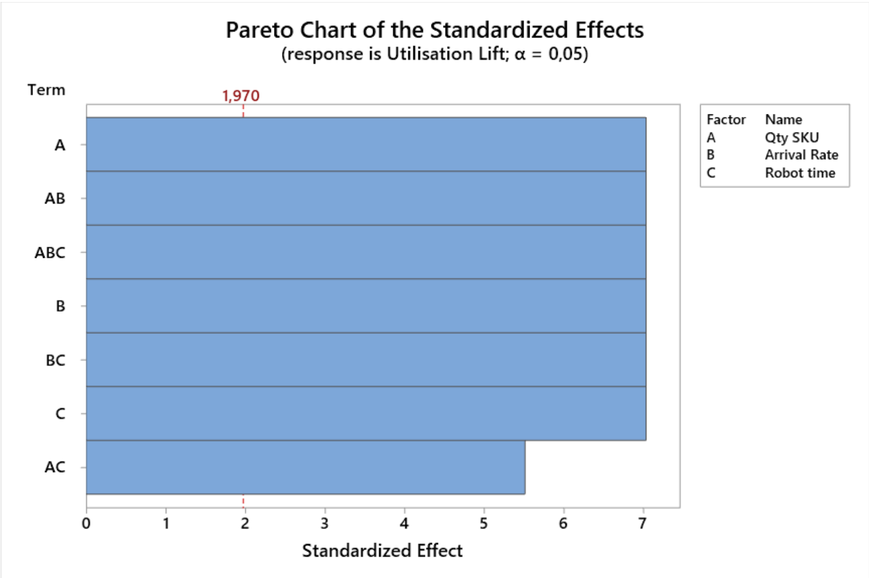
Source: the author

Figure 33: Standardized effects of the factors on the current system's vehicle utilization



Source: the author

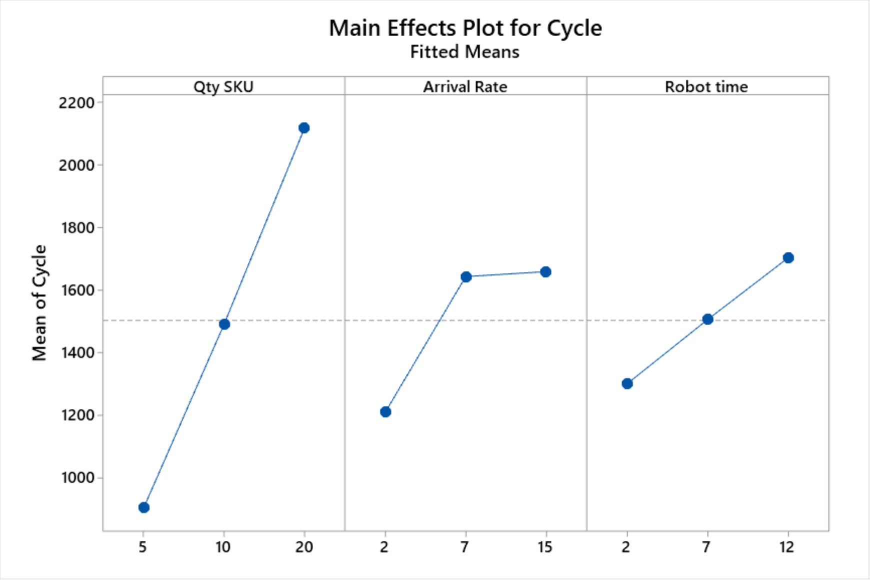
Figure 34: Standardized effects of the factors on the current system's lift utilization



Source: the author

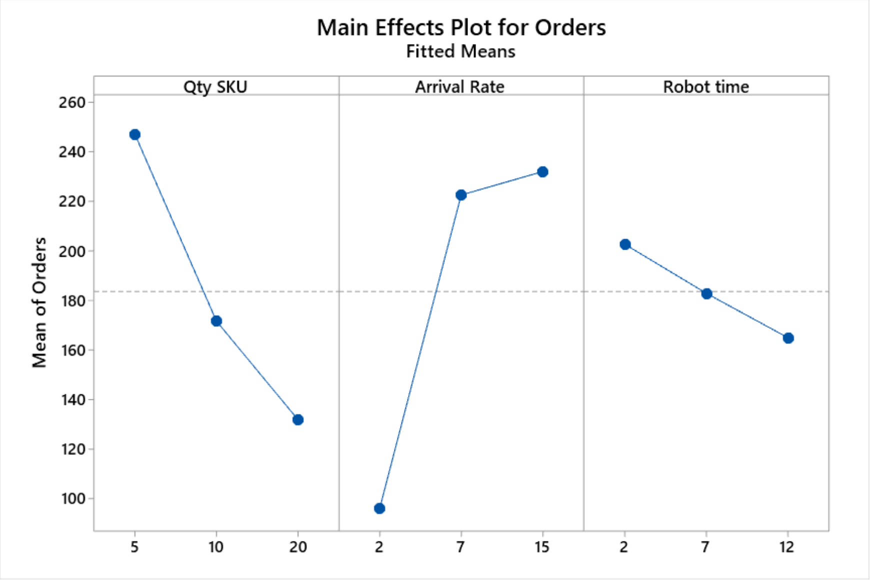
The effects of each factor on the system’s cycle time, quantity of orders completed, vehicle utilization and lift utilization can be analyzed by the charts illustrated in Figure 35, Figure 36, Figure 37 and Figure 38, respectively.

Figure 35: Current system's average cycle time for each factor level



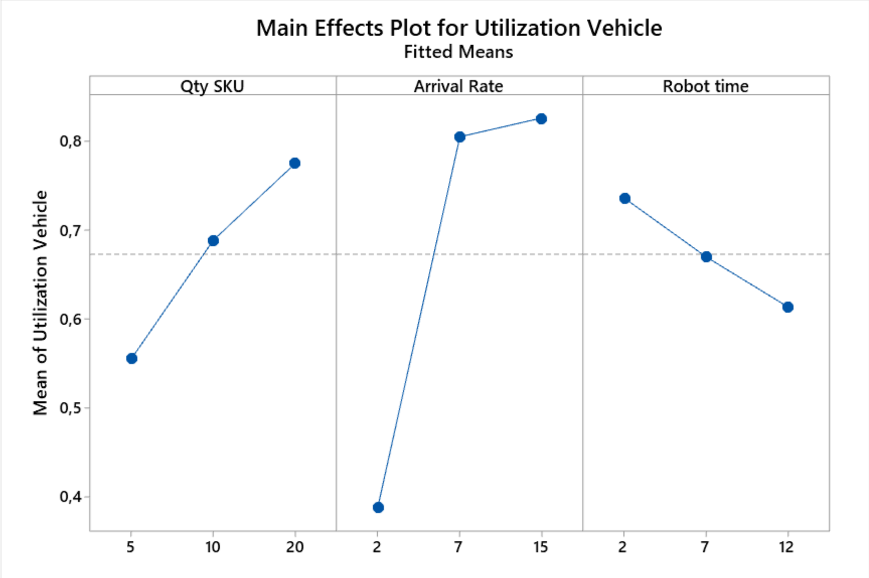
Source: the author

Figure 36: Current system's average number of completed orders for each factor level



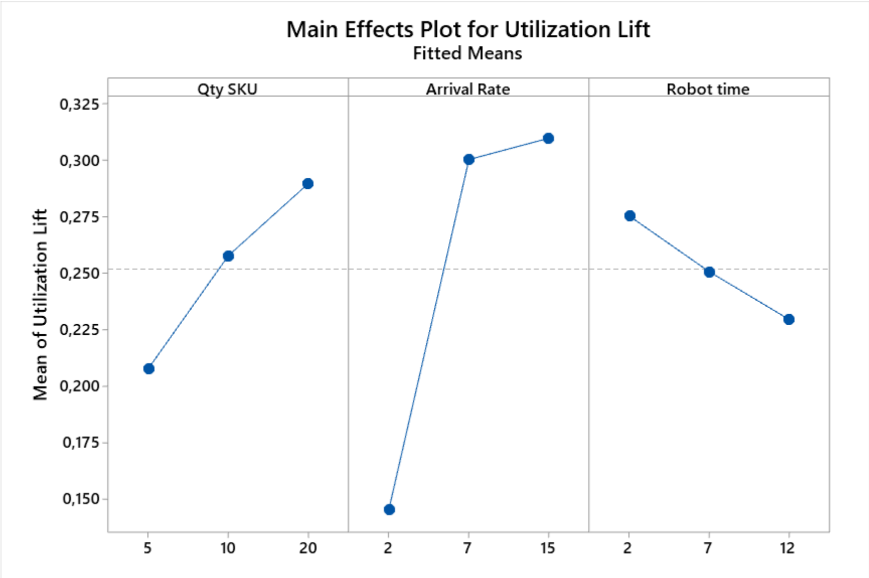
Source: the author

Figure 37: Current system's average vehicle utilization for each factor level



Source: the author

Figure 38: Current system's average lift utilization for each factor level



Source: the author

In relation to the effect that the number of SKUs has on the current system performance, it is observed that the systems performs better in conditions with lower quantity of SKUs. This can be explained as the number of retrieval is related to the number of different products ordered, thus if there are more variety the system needs to retrieve more pallets from the storage. Consequently, the cycle time increases, as well as the vehicle and lift utilization, while the number of orders completed decreases.

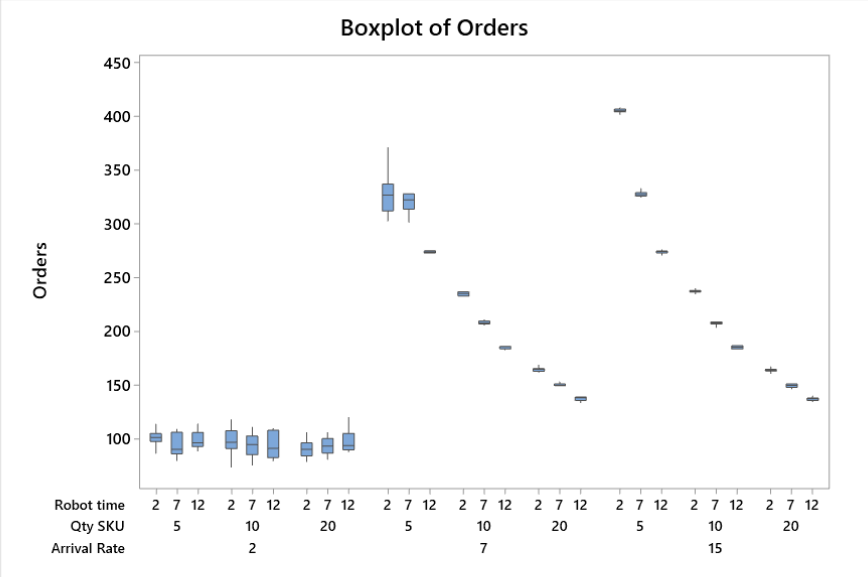
Regarding the impact of the orders arrival rate, it can be verified that the systems reaches its maximum capacity when the retrieval rate is approximately 7 and the number of completed orders stabilizes around 220, as the resources utilization stabilizes around this value. The vehicle utilization does not reach 100% because in the system's model it was considerated that the vehicle is going to be requested only when the slot at the palletizing area is available, consequently while the SKU ordered is being transferred to the mixed pallet the vehicle is idle, because the next retrieval process is not allowed to started yet. Moreover, it is noticed that the lift remains idle during most of the time, as its utilization is around 30% when the vehicle has reached its maximum capacity. This indicates that the system has capacity to operate with more than 1 vehicle at a time.

Finally, in relation to the robotic arm picking time, it is observed that it also impacts the system's throughput capacity, even though the vehicle and robotic arm processes are independent. This occurred also due the assumption the vehicle would be requested by to execute the next retrieval only when the slot at the palletizing area is free. Consequently, the higher the robotic arm picking time is, the longer is the interval between retrieval transactions processes and the lower is the thoughput capacity.

The chart displayed in Figure 39 presents an overview of the system's throughput capacity under the different scenarios analyzed. For the case the orders arrival rate is equal to 2 orders per hours, the performance measure does not change much among the scenarios. This occurs because the system has capacity to process all the orders that arrives. In the chart it can be observed that number of orders in each scenario with arrival rate equal to 2 is close to 96, as expected since 2 orders arrives per hour during the 48 hours simulated.

When the arrival rate increases, the chart shows that the system does not have capacity to complete all the orders, thus the throughput capacity changes depending on the scenario. As showed in the boxplot displayed in Figure 39, the system's throughput capacity decreases as the robotic arm picking time and number of SKUs increases.

Figure 39: Boxplot of the results for orders completed - Current System

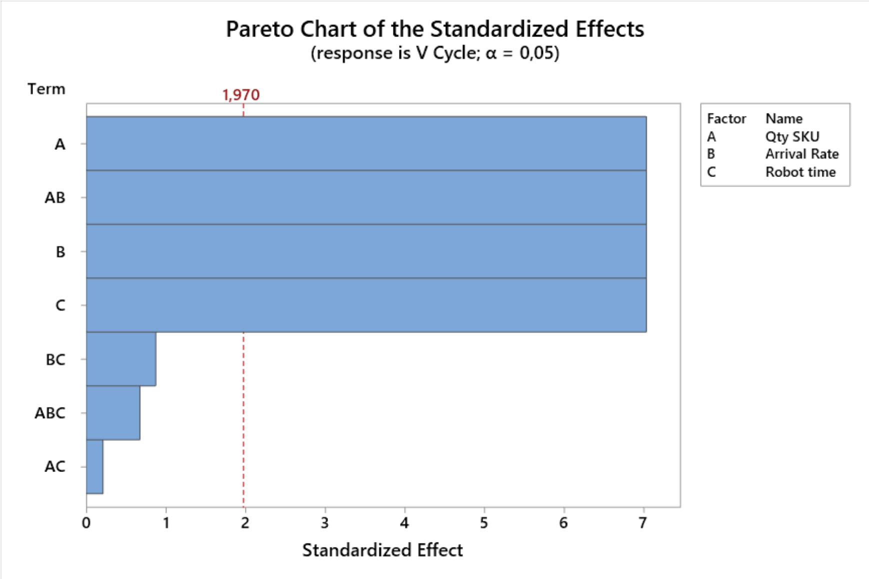


Source: the author

6.2.2. New system

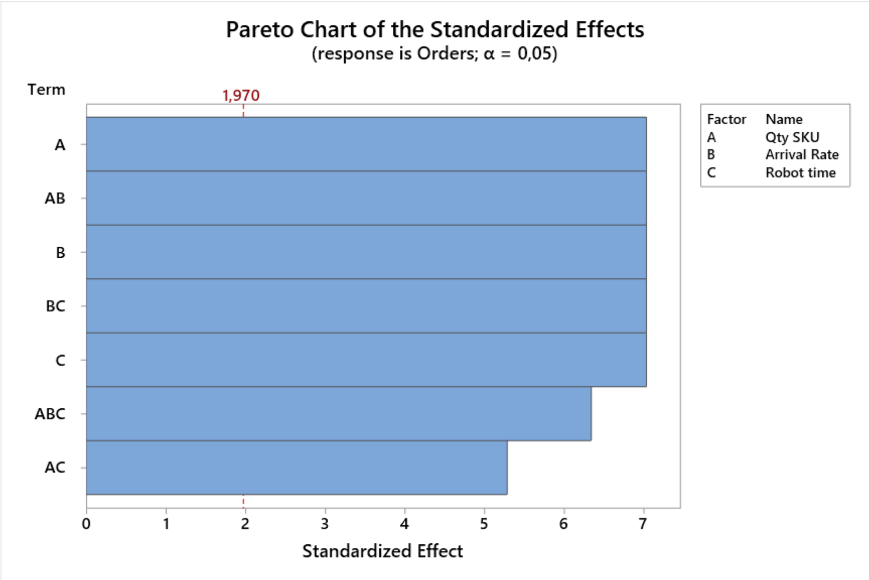
The Pareto Charts displayed in Figure 40, Figure 41, Figure 42 and Figure 43 shows the standardized effects of each factor on the new system's cycle time, throughput capacity, vehicle and lift utilization, respectively. The three factors considered individually impacted significantly all performance measures evaluated, considering a confidence interval of 95%. Besides that, their interaction also impacted the performance indicators, with exception of the cycle that that was only significantly impacted by the interaction between the number of SKUs and the arrival rate. Therefore, when considering the application of the new system in a warehouse, all these factors are important and need to be considered, as they impact on how well the new system is going to perform.

Figure 40: Standardized effects of the factors on new system's cycle time



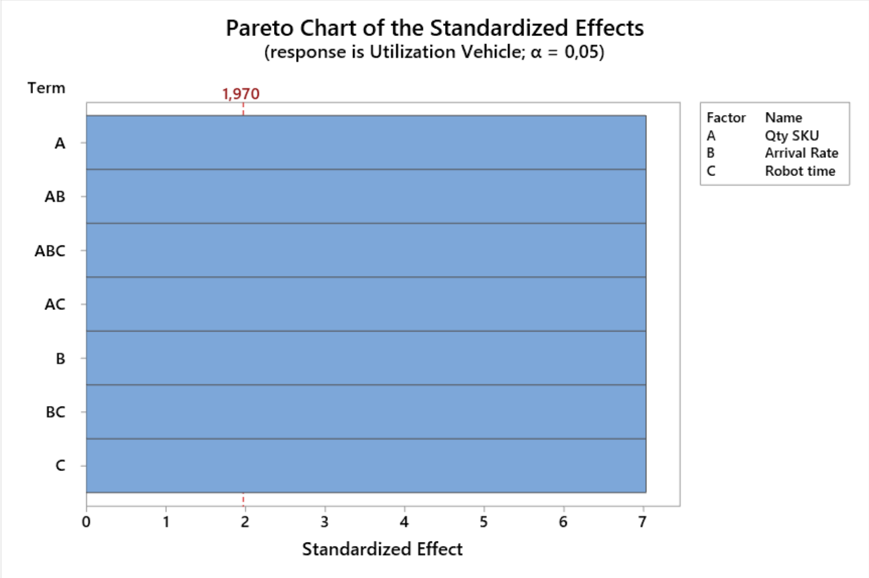
Source: the author

Figure 41: Standardized effects of the factors on new system's throughput capacity



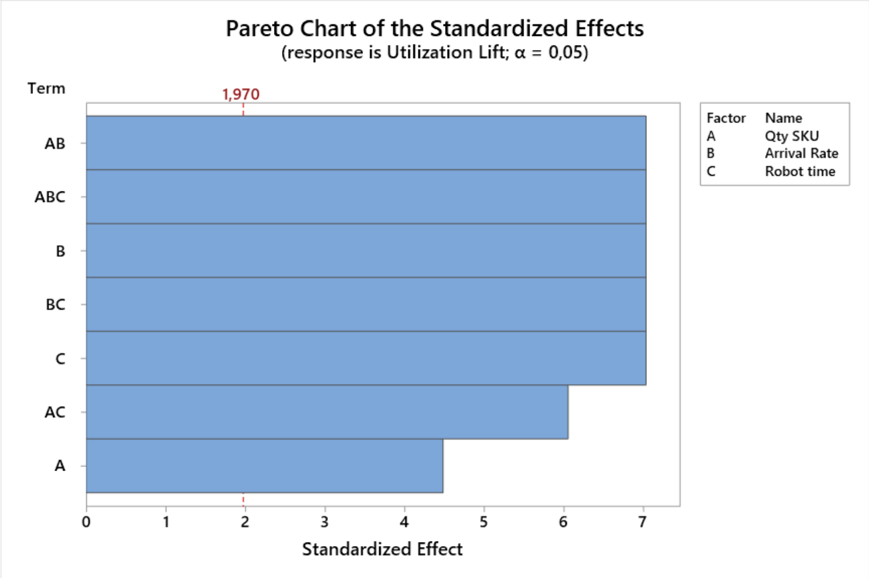
Source: the author

Figure 42: Standardized effects of the factors on new system's vehicle utilization



Source: the author

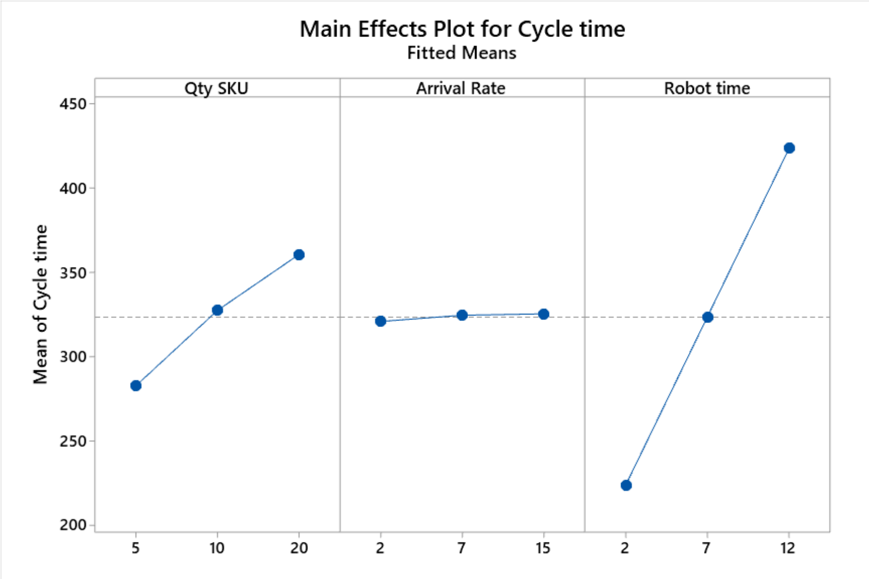
Figure 43:Standardized effects of the factors on new system's lift utilization



Source: the author

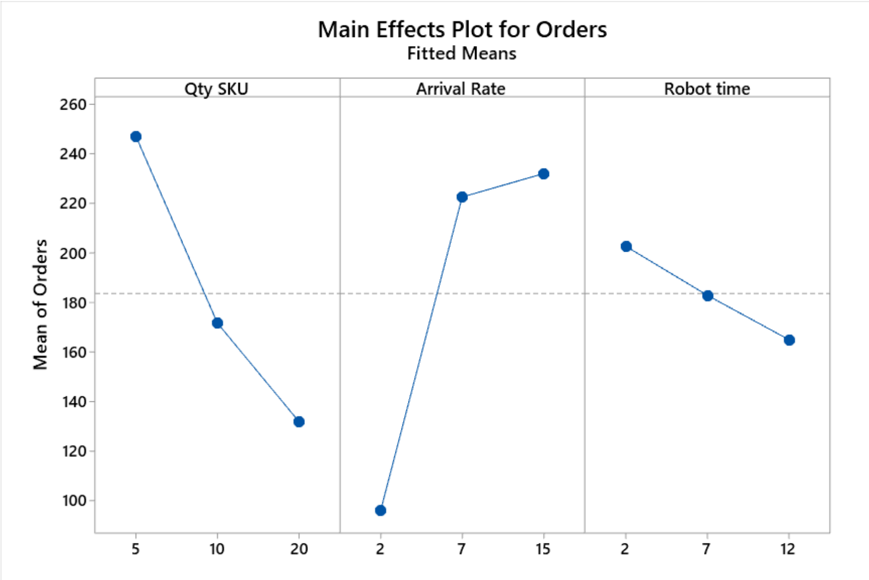
The impact of the factors on the systems performance can be observed in the charts illustrated in Figure 44 (cycle time), Figure 45 (quantity of orders completed), Figure 46 (vehicle utilization) and Figure 47 (lift usage).

Figure 44: New system's average cycle time for each factor level



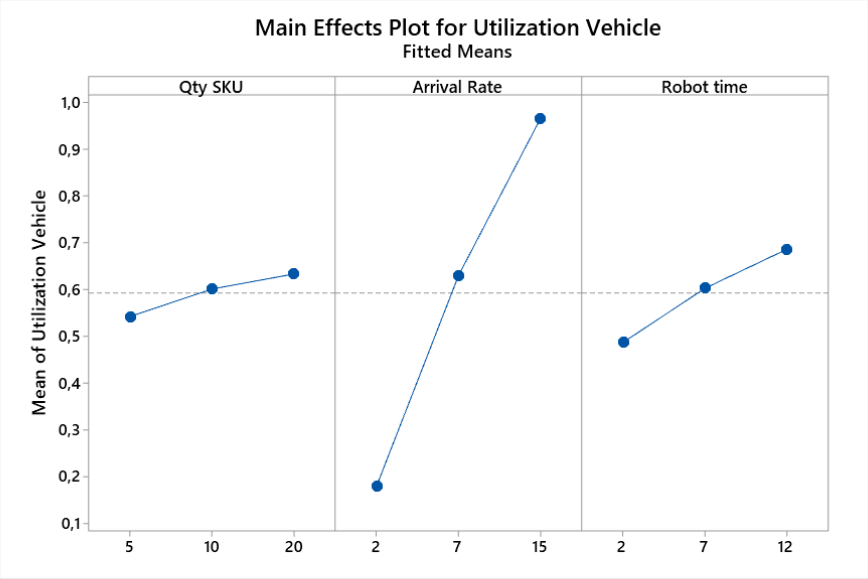
Source: the author

Figure 45: New system's average number of completed orders for each factor level



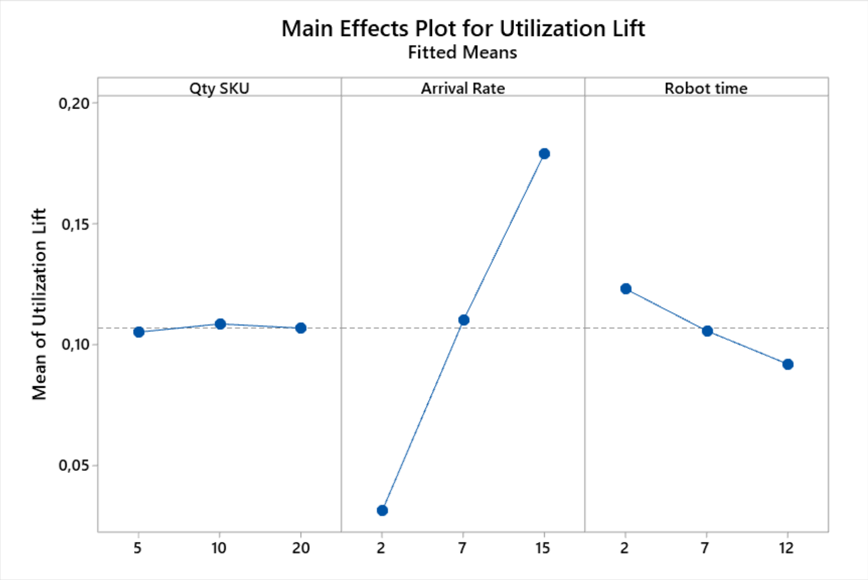
Source: the author

Figure 46: New system's average vehicle utilization for each factor level



Source: the author

Figure 47: New system's average lift utilization for each factor level



Source: the author

In relation to the effect of the number of SKUs in the storage, it is observed that the higher it is the higher is the cycle time and, consequently, the lower is the number of orders completed. Besides that, both resources utilization also increases when there are more SKUs stored. It is expected, as the variety of the mixed pallets is going to increase and therefore the vehicle must travel to more rack locations in order to collect all the items ordered. In this way, not only the vehicle movements more, but it also requires more the lift in order to move from one tier to another.

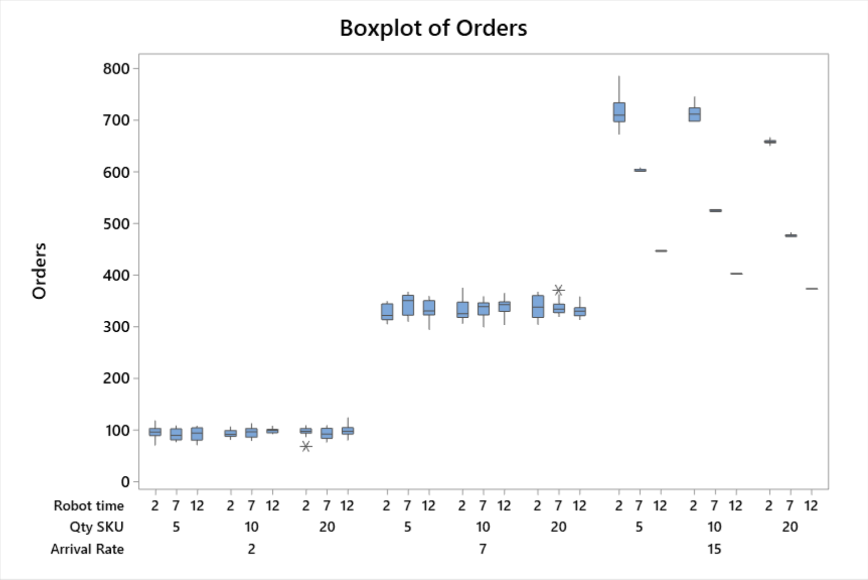
Regarding the effect of the orders arrival rate, it is verified that it does not impact the cycle time, as the cycle time is computed from the moment the vehicle starts processing the order, not since the order arrives. However, it does impact on the other performance measures, increasing all of them when the arrival rate is higher. The number of completed orders presents a significant increase when the arrival rate changes from 2 to 7, but from 7 to 15 the increment is smaller. This occurs because the vehicle utilization reaches its maximum when the arrival rate is around 15 orders per hour, consequently it does not matter if the arrival rate is higher since the system does not have capacity to build more mixed pallets that it already builds. The lift utilization increases with more orders arriving, but as the vehicle reaches its maximum capacity the lift utilization tends to stabilize, remaining around 20%.

Finally, with respect to the effect of robotic arm picking time, the higher it is the worse it the system's performance. The cycle time is strongly impacted by this factor, as showed in Figure 44 the cycle time is directly related to the robot picking time. This happens because in this system the vehicle travels and the picking process are dependents, so the vehicle can not move to another rack position while all the items were transferred from the stored pallet to the mixed one. Besides that, the total time to transfer all the ordered items is equal to picking time multiplied by the number of boxes in the mixed pallet. Since the cycle time increases with the robot picking time in the new system, the number of orders completed decreases. Besides that, the vehicle spends more time in front of the rack positions to transfer the items from the stored pallets to the mixed one, consequently it spends less time traveling from one tier to another, so the lift utilization become more idle.

The boxplot chart showed in Figure 48 summarizes the simulation results considering the number of orders completed in each scenario. It is possible to verify that the system operates under its maximum capacity when the arrival rate is equal to 2 and 7. The quantity of completed orders remains approximately the same for all scenarios and the number of completed orders is around 96 (2*48) and 336 (7*48), which means the system is attended all the orders that arrived during the simulation period.

Regarding the results when the arrival rate is 15, the boxplot chart shows that the robot picking time has a strong impact on results, as the number of completed orders in the best scenario (2 seconds) is about 40% lower in the worst scenario (12 seconds). In relation to the number of SKUs, the system's performance is impacted by this factor, but it is not that strong.

Figure 48: Boxplot of orders - New System



Source: the author

6.3. Number of vehicles

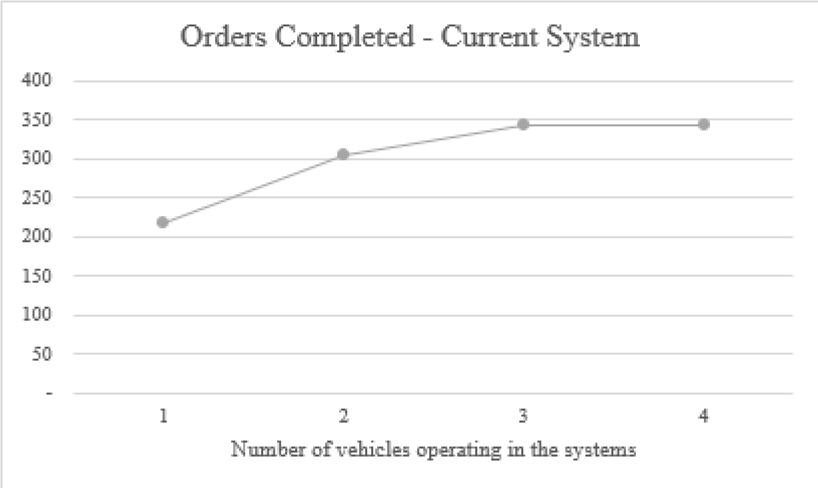
In the factorial experiment described before there was only one vehicle operating and in the current system only one slot for retrieved pallets. In this way for both systems a single ordered pallet was built per time. In this section it is described the impact of adding vehicles in each system.

In relation to the current system considered, when added a vehicle it was also included a need slot for retrieved pallets. Consequently, in order to improve the system performance by adding vehicles, the palletizing area also must get bigger. Moreover, it was considered that the system would build an order per time, so there is only one slot for mixed pallet. In this way, the vehicles are going to retrieve all pallets required to attend only this order.

In relation to the improvement of the systems output, as showed in the chart displayed in

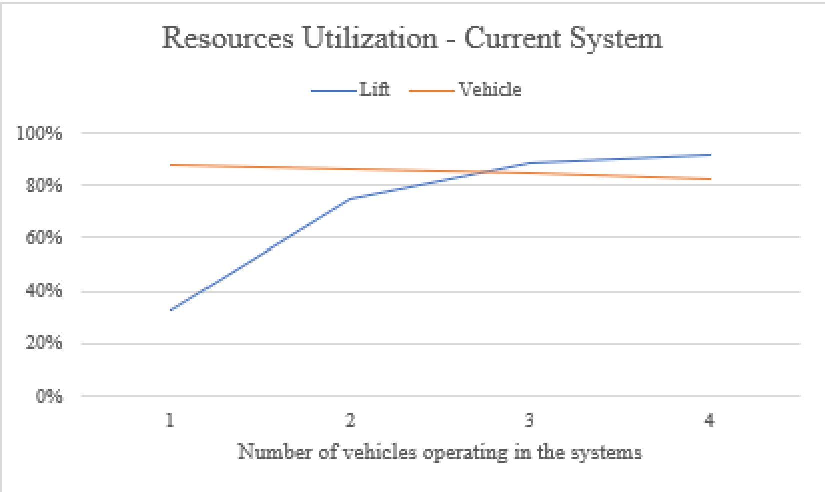
Figure **49**, the quantity of orders completed increases as the number of vehicles increases. The system with 4 vehicles, the throughput capacity increased 57% with respect to the scenario with 2 vehicles. However, the throughput capacity of the system tends to stabilize when the number of vehicles is around 3, it occurs because the lift utilization increases, and this resource becomes the bottleneck of the system. Besides that, the vehicles' utilization rate decreases as the lift becomes the bottleneck, as illustrated in the graph in Figure 50.

Figure 49: Throughput capacity for current AVS/RS with different number of vehicles



Source: the author

Figure 50: Resources utilization for Eurofork's system with different number of vehicles

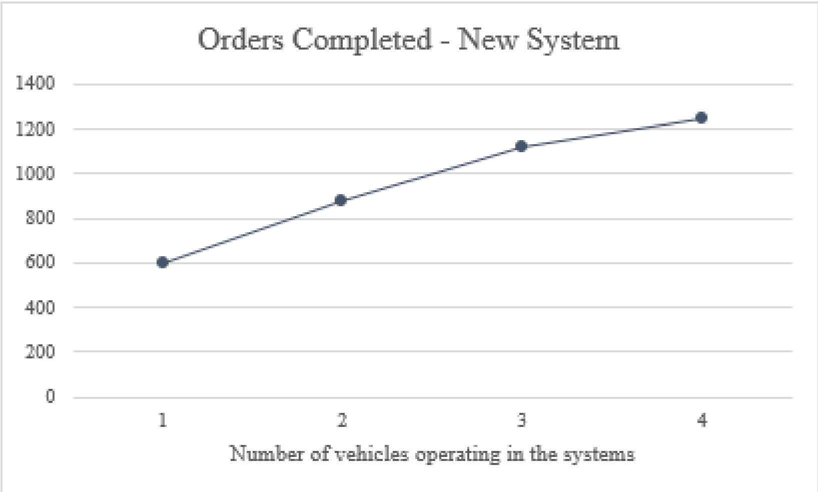


Source: the author

Regarding the New system, the results obtained from the simulations with multiple vehicles is presented in Figure 51, throughput capacity, and Figure 52, resources utilization. As expected, the quantity of orders completed is proportional to the number of vehicles in the systems, since each vehicle builds one pallet per time. Besides that, it is observed that the lift utilization increases, and it reaches almost 100% when there are 4 vehicles, so it is possible to assume that the system with one single lift almost reached its maximum throughput capacity with 4 vehicles. Then, the more vehicles are added the more the cycle time is going to increase due the waiting time for the lift.

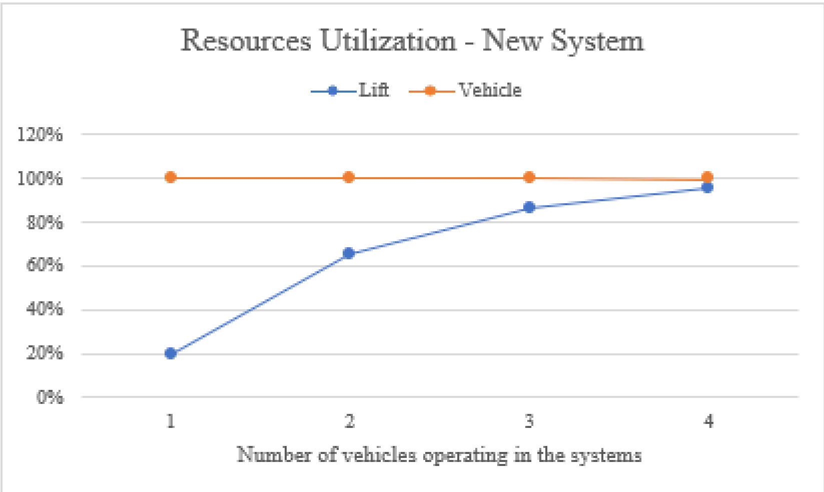
Although the results from the simulation give an idea of the impact of adding a vehicle in the system, in the reality the impact might be smaller, since in the simulation it was assumed that collisions between the vehicles would not occur. Instead, the vehicle path should be defined in order to avoid collisions and it could consequently increase the distance travelled by vehicles and the waiting times, impacting the system's performance.

Figure 51: Througtput capacity for New system with different number of vehicles



Source: the author

Figure 52: Resources utilization for New system with different number of vehicles



Source: the author

6.4. Comparison between the systems

Comparing the systems results when changing the number of vehicles, it was verified that the Eurofork's new system presents a better performance than the current AVS/RS in all scenarios analyzed. Moreover, even with 4 vehicles, the throughput capacity of the current system, which is approximately 350 orders completed, is still lower than the New system capacity for the same scenario but with only one vehicle, that is around 600 orders finished.

In addition to it, the new system is also less impacted by an increase in the number of SKUs stored. This indicates that this system works better than the current one in warehouses that deals with a high number of SKUs. Especially when the warehouse ships pallets with a big variety of items.

Due to the shorter cycle time observed in the New system, it also shows a higher performance than the current one when the orders arrival rate increases. For a small arrival rate the systems does not present big performance discrepancies in relation to quantity of orders completed. This occurs because both systems are able to build the mixed pallets during the interval between arrivals. Therefore, the throughput capacity for both systems are similar, but resources usages are not as the resources in Eurofork's systems remains idle for longer periods. Taking into account, the higher the arrival rate the more Eurofork's system is indicated.

Regarding the robot picking time, as verified by the results, it is a critical variable for the new system performance. Since the travelling processes are connected to the picking/palletizing one in Eurofork's system, the cycle time is directly related to the robot picking time. In the current system, the retrieval process for one product can happen at the same time of the picking process of another item, the system can execute both process for different products at the same time. Therefore, it can be concluded that Eurofork needs to focus on developing a robotic arm that transfers items from one pallet to another in the fastest way as possible.

Furthermore, in the simulation model developed for the current system, it is considered only one mixed pallet being built at the same time. In fact, there could be many pallets being built, with the retrieved pallets providing the items for more than one pallet in the palletizing area. Due the complexity to simulate this improvement, this configuration for the palletizing area was not modelled. However, it is expected that the current system presents a higher throughput capacity with this modification in its operation, but this improvement would also depend on a series of factor, for instance the number of slots for mixed and retrieved pallets and the distance between the robotic arm and the pallets.

In relation to the new system, the model was simplified by assuming that the vehicles would never collide. As a matter of fact, a routing optimization would be necessary in order to avoid collisions and consequently the cycle time might be impacted. In addition to it, the application of different storage policies, such as class-based, could improve the systems throughput capacity and vehicles usage, as it may reduce travels distance for instance.

As a conclusion, even though the models developed in this study assumed some simplifications, the results show that the new system tends to work better than the current one, especially when the operation involves a high quantity of SKUs. Besides that, since the new system also presents the advantage of not requiring a separated zone for the palletizing process, it is more flexible. The limited space in a warehouse can limit the expansion of the current system, while it does not interfere in the new system expansion. However, the robot picking time is very critical and depending of its values the system can become worse than the current one. Therefore, further comparisons are required when Eurofork defines the technology features and the warehouse where it is going to be applied, in order to represent better reality and consequently obtain the results of the real systems' operation.

7. CONCLUSIONS

This study main objective was the performance evaluation and comparison of two different autonomous vehicle storage and retrieval systems (AVS/RS). The two systems analyzed are: a tier-to-tier AVS/RS with end-of-aisle picking and a new type of AVS/RS system that is still being developed by an Italian company, Eurofork. This new system was designed considering the increasing demand for mixed pallets by the warehouses' clients. The developer company aims to integrate the picking and palletizing processes in a better way and consequently improve the whole warehouse operation.

The systems differ in relation to the moment and location that the picking activity occurs. Both picking activities are executed by robotic arm, but in the first system the operation occurs in a palletizing area dedicated to it and after the homogenous pallets are retrieved to this zone. While in the second one it happens in front of the channels in the storage rack and there is no need to retrieve the stored pallets. The difference on the picking activity impacts the whole operation as the new system does not require the pallets with ordered items to be retrieved to a different zone outside the storage rack.

The systems' performances were measured through simulation of their operation in a virtual warehouse, enabling the evaluation and comparison between both. Therefore, the simulation results were used to verify if the new system will perform better than the current one, as expected by the company. Besides the comparison between the two systems, since the system was simulated under different scenarios, it was possible to observe how the some of the system's variables impacts on its performance. In this way, this study also supports Eurofork in the development of the new system by pointing out important factors to improve the system's efficiency. Moreover, it also indicates under which conditions the system might be ideal because it performs much better than the other system analyzed.

Before the simulation model development, first it was carried out a literature review about topics related to the problem. In this manner, studies about the different autonomous storage and retrieval systems were analyzed, as well as the ones that describes simulations models for this kind of system. Besides that, the mixed pallet topic was also studied, since Eurofork is developing the system considering that it would be ideal for warehouses that receive homogeneous pallets and ship mixed ones. Then, the simulation model was developed taking into account what was observed in the literature and the information provided by the company.

The model developed presents limitations due to the lack of data of a real case scenario, since the new system is still being developed. Besides that, the performance was evaluated considering only one mixed pallet being built per time. As mentioned before, in the palletizing area more than one pallet can be built at the same time, saving time as the items from the retrieved pallet can be distributed in many pallets instead of requesting one retrieval transaction per client's order. However, the increase in the throughput capacity is not unlimited, since there are other resources with limited capacity (lift and vehicle) and there is the space constraint for the expansion.

In relation to the model developed to simulate the new system, it presents the limitation of not considering the possible collisions between vehicles. The system's performance might be impacted when considering the collision, since the travel might be deviated and consequently the distance traveled increases as well as the cycle time. might reduce the system's performance. Yet, the model also does not optimize the route, the rack locations are chosen randomly, so there is still opportunity to increase the system's performance.

Finally, a financial analysis of both systems is important in order to define which one performs better. Since the technology has not been developed, the financial data was not available, thus it was not possible to do this evaluation. Further, Eurofork needs to consider this factor when developing the final technology, as it might be crucial when choosing which system to apply in a warehouse. Yet, the analysis done by this study indicates that the new system has a great potential of application, especially when there are many SKUs available in the storage, since the client's orders tends to be more varied and consequently more retrievals are requested in the current system.

8. REFERENCES

AZADEH, K.; DEKOSTER, M. B. M.; ROY, D. Robotized Warehouse Systems: Developments and Research Opportunities. **ERIM Report Series Research in Management**, n. ERS-2017-009-LIS, 2017.

AZADEH, Kaveh; KOSTER, René de; ROY, Debjit. Robotized and Automated Warehouse Systems: Review and Recent Developments. Transportation Science, [s.l.], v. 53, n. 4, p.917-945, jul. 2019. Institute for Operations Research and the Management Sciences (INFORMS). http://dx.doi.org/10.1287/trsc.2018.0873.

BANKS, Jerry et al. Discrete-event system simulation. Upper Saddle River, NJ: Prentice hall, 1996.

BASTIAN, II William A.; CALLOWAY, Britt. Robotic automated storage and retrieval system mixed pallet build system. U.S. Patent n. 8,882,432, 11 nov. 2014.

BASTIAN SOLUTIONS. Robotic Mixed Load Palletizer. Disponível em: <https://www.bastiansolutions.com/solutions/service/industrial-robotics/industrial-robotic-solutions/robotic-palletizing/robotic-mixed-load-palletizer/>. Access on: May 15, 2019.

BASTIAN SOLUTIONS. Robotic Order Batch Picking. Disponível em: <https://www.bastiansolutions.com/solutions/service/industrial-robotics/industrial-robotic-solutions/mobile-robotics/order-batch-picking/>. Access on: May 15, 2019.

BRUNO, Giulia; D'ANTONIO, Gianluca. Flexible reconfiguration of AVS/RS operations for improved integration with manufacturing processes. **Procedia Cirp**, [s.l.], v. 78, p.196-201, 2018. Elsevier BV. http://dx.doi.org/10.1016/j.procir.2018.09.052.

EKREN, Banu Y. et al. Simulation based experimental design to identify factors affecting performance of AVS/RS. **Computers & Industrial Engineering**, [s.l.], v. 58, n. 1, p.175-185, fev. 2010. Elsevier BV. http://dx.doi.org/10.1016/j.cie.2009.10.004.

EKREN, Banu Y.; HERAGU, Sunderesh S.. Simulation based performance analysis of an autonomous vehicle storage and retrieval system. **Simulation Modelling Practice And Theory**, [s.l.], v. 19, n. 7, p.1640-1650, ago. 2011. Elsevier BV. http://dx.doi.org/10.1016/j.simpat.2011.02.008.

EKREN, Banu Yetkin; HERAGU, Sunderesh S.. Simulation-based regression analysis for the rack configuration of an autonomous vehicle storage and retrieval system. **International Journal Of Production Research**, [s.l.], v. 48, n. 21, p.6257-6274, 7 dez. 2009. Informa UK Limited. http://dx.doi.org/10.1080/00207540903321665.

EXOTEC. **Solutions.** Disponível em: <https://www.exotec.com/solutions/?lang=en>. Access on: May 26, 2019.

EPP, Martin; WIEDEMANN, Simon; FURMANS, Kai. A discrete-time queueing network approach to performance evaluation of autonomous vehicle storage and retrieval systems. **International Journal Of Production Research**, [s.l.], v. 55, n. 4, p.960-978, 14 jul. 2016. Informa UK Limited. http://dx.doi.org/10.1080/00207543.2016.1208371.

GU, Jinxiang; GOETSCHALCKX, Marc; MCGINNIS, Leon F.. Research on warehouse operation: A comprehensive review. **European Journal Of Operational Research**, [s.l.], v. 177, n. 1, p.1-21, fev. 2007. Elsevier BV. http://dx.doi.org/10.1016/j.ejor.2006.02.025.

GÜLLER, Mustafa; HEGMANNS, Tobias. Simulation-based Performance Analysis of a Miniload Multishuttle Order Picking System. **Procedia Cirp**, [s.l.], v. 17, p.475-480, 2014. Elsevier BV. http://dx.doi.org/10.1016/j.procir.2014.01.041.

KOSTER, René de; LE-DUC, Tho; ROODBERGEN, Kees Jan. Design and control of warehouse order picking: A literature review. **European Journal Of Operational Research**, [s.l.], v. 182, n. 2, p.481-501, out. 2007. Elsevier BV. http://dx.doi.org/10.1016/j.ejor.2006.07.009.

LERHER, T.; EKREN, Y. B.; SARI, Z.. Simulation Analysis of Shuttle Based Storage and Retrieval Systems. **International Journal Of Simulation Modelling**, [s.l.], p.48-59, 15 mar. 2015. DAAAM International. http://dx.doi.org/10.2507/ijsimm14(1)5.281.

LERHER, Tone. Design of Experiments for Identifying the Throughput Performance of Shuttle-Based Storage and Retrieval Systems. **Procedia Engineering**, [s.l.], v. 187, p.324-334, 2017. Elsevier BV. http://dx.doi.org/10.1016/j.proeng.2017.04.382.

MAGAZINO. Toru. Disponível em: <https://www.magazino.eu/products/toru/?lang=en>. Access on: Jun. 10, 2019.

MAGENS, Ernst-Peter; BECKHAUS, Kai; SEEMÜLLER, Stefan. Autonomous order picker. U.S. Patent Application n. 15/293,778, 20 abr. 2017.

MANZINI, Riccardo; BOZER, Yavuz; HERAGU, Sunderesh. Decision models for the design, optimization and management of warehousing and material handling systems. **International Journal Of Production Economics**, [s.l.], v. 170, p.711-716, dez. 2015. Elsevier BV. http://dx.doi.org/10.1016/j.ijpe.2015.08.007.

MANZINI, Riccardo et al. Travel time models for deep-lane unit-load autonomous vehicle storage and retrieval system (AVS/RS). International Journal Of Production Research, [s.l.], v. 54, n. 14, p.4286-4304, 9 fev. 2016. Informa UK Limited. http://dx.doi.org/10.1080/00207543.2016.1144241.

MARKETS AND MARKETS. Automated Storage and Retrieval System (ASRS) Market by Type (Unit Load, Mini Load, VLM, Carousel, Mid Load, Autostore), Function (Assembly, Kitting, Order Picking, Distribution, Storage), Industry, and Geography - Global Forecast to 2024. Available on: <https://www.marketsandmarkets.com/Market-Reports/automated-storage-retrieval-system-market-195267987.html>. Access on: Oct. 20, 2019.

MATERIAL HANDLING INDUSTRY OF AMERICA. ASRS Group Releases Infographic. Available on: <http://www.mhi.org/downloads/industrygroups/as-rs/as-rs-infographic.pdf>. Access on: Sept. 21, 2019.

MICHEL, Roberto. 2016 Warehouse/DC Operations Survey: Ready to Confront Complexity, *Supply Chain Management Review*, pp. S52-S59, Nov. 2016.

MHD Supply Chain Solutions, Vol. 48, No. 4, Aug 2018: 18-19. Automation is the star [online]. Availability:<https://search.informit.com.au/documentSummary;dn=730889688143954;res=IELENG> ISSN: 1445-0801. (Last access: May 2019).

MITAL, Anil. Comparison of manual and automated palletizing of mixed size and weight box containers. **International Journal of Industrial Ergonomics**, v. 9, n. 1, p. 65-74, 1992.

MONGOLD, Michael B.; JOHNSON, Alan W. Pure pallets: effectiveness and efficiency impacts on the defense transportation system. **Journal of Transportation Management**, v. 17, n. 1, p. 18-36, 2006.

MORENCY, Sylvain-Paul et al. Order picking system. U.S. Patent Application n. 12/149,177, 30 out. 2008.

MITCHELL, Pat R. Automated build-by-layer rainbow pallet system. U.S. Patent Application n. 12/584,257, 18 mar. 2010.

NING, Zhao et al. An efficient simulation model for rack design in multi-elevator shuttle-based storage and retrieval system. **Simulation Modelling Practice And Theory**, [s.l.], v. 67, p.100-116, set. 2016. Elsevier BV. http://dx.doi.org/10.1016/j.simpat.2016.03.007.

ROODBERGEN, Kees Jan; VIS, Iris F.a.. A survey of literature on automated storage and retrieval systems. **European Journal Of Operational Research**, [s.l.], v. 194, n. 2, p.343-362, abr. 2009. Elsevier BV. http://dx.doi.org/10.1016/j.ejor.2008.01.038.

SWISSLOG. CarryStar: A fully automated layer and stack picking solution. Disponível em: <https://www.swisslog.com/en-au/logistics-automation/products-systems-solutions/picking-palletizing/goods-to-person/carrystar>. Access on: June 05, 2019.

TAPPIA, Elena et al. Integrated storage-order picking systems: Technology, performance models, and design insights. European Journal Of Operational Research, [s.l.], v. 274, n. 3, p.947-965, maio 2019. Elsevier BV. http://dx.doi.org/10.1016/j.ejor.2018.10.048.

TROTTMANN, Marco; ZHANG, Sam. The trend towards warehouse automation, *Westernacher Knowledge Series, 2017*.

YAMAN, Hande; ŞEN, Alper. Manufacturer's mixed pallet design problem. **European Journal Of Operational Research**, [s.l.], v. 186, n. 2, p.826-840, abr. 2008. Elsevier BV. http://dx.doi.org/10.1016/j.ejor.2007.02.007.

ZHOU, Guiliang; MAO, Lina. Design and Simulation of Storage Location Optimization Module in AS/RS Based on FLEXSIM. **International Journal Of Intelligent Systems And Applications**, [s.l.], v. 2, n. 2, p.33-40, 16 dez. 2010. MECS Publisher. http://dx.doi.org/10.5815/ijisa.2010.02.05.

APPENDIX

1. Current AVS/RS simulation model

```python
# -*- coding: utf-8 -*-
"""
New system - Eurofork's system
"""
import simpy
import math
import random
import numpy
import xlwings as xw

" -------- Fixed Variables for the scenario -------- "
CHANNELS = 19
LEVELS = 6
AISLES = 1
SIDES = 2
QTY_LIFTS = 1
VEL_V = 2.0 #velocity vehicle
ACC_V = 0.5 #acceleration vehicle
TIME_PEAK_V = (2* VEL_V) / ACC_V #interval to vehicle reaches the peak velocity
VEL_L = 14.0/60 #velocity lift
ACC_L = 0.3 #acceleration lift
TIME_PEAK_L = (2* VEL_L) / ACC_L #interval to lift reaches the peak velocity
WIDTH = 1.3
HEIGHT = 2.0
TRANSFER_S = 14.0 #time to transfer load to shuttle, from Ekhren and Heragu (2011)
TRANSFER_L = 0.0 #time to transfer shuttle to lift,  from Ekhren and Heragu (2011)
WARM_UP = 10800

" --------- Variables for each scenario ---------- "
QTY_REPLICATION = 10
BOXES = 20
TRANSFER_PA = 0
QTY_VEHICLES = 1.0
SLOTS_PA_MIXED = 2 #slots in the packaging area destinated for mixed pallets to be built
SLOTS_PA_RETRIEVED = 2 #solts in the packaging area destinated for retrieved pallets

"""according to Ekhren and Heragu (2011),
assignment of 3 vehicles to 1 lift will ensure
one resource does not hae to wait too much for the other
Verify the average lift and vehicle transaction times to
compute this value"""



" -------------- Class Processes (steps)--------------------"
class Processes:
    def __init__(self, env, num_order, retrieval, resource, position_I, position_F, time_I, time_F, duration):
        self.num_order = num_order
        self.retrieval = retrieval
        self.resource = resource
        self.position_I = position_I
        self.position_F = position_F
        self.time_I = time_I
        self.time_F = time_F
        self.duration = duration

" -------------- Class Processes Total Vehicle ---------------------"
class Processes_Vehicle:
```

```
    def __init__(self, env, num_order, retrieval, resource, time_I, time_F, duration):
        self.num_order = num_order
        self.retrieval = retrieval
        self.resource = resource
        self.time_I = time_I
        self.time_F = time_F
        self.duration = duration

" -------------- Class Processes Total Lift ---------------------"
class Processes_Lift:
    def __init__(self, env, num_order, retrieval, resource, time_I, time_F, duration):
        self.num_order = num_order
        self.retrieval = retrieval
        self.resource = resource
        self.time_I = time_I
        self.time_F = time_F
        self.duration = duration


" -------------- Class Data ---------------------"
class Data:
    def __init__(self, env, retrieval, order, time_I, vehicle_I, time_F, duration, duration_V):
        self.retrieval = retrieval
        self.order = order
        self.time_I = time_I
        self.vehicle_I = vehicle_I
        self.time_F = time_F
        self.duration = duration
        self.duration_V = duration_V


" -------- Resources ----------- "
def Resources(env, QTY_LIFTS, QTY_VEHICLES, SLOTS_PA_MIXED, SLOTS_PA_RETRIEVED):
    lift = simpy.Resource(env, QTY_LIFTS)
    vehicles = simpy.Resource(env, QTY_VEHICLES)
    slots_mixed = simpy.Resource(env, SLOTS_PA_MIXED)
    slots_retrieved = simpy.Resource(env, SLOTS_PA_RETRIEVED)
    robot = simpy.Resource(env, 1)
    return lift, vehicles, slots_mixed, slots_retrieved, robot


       " -------- Storage creation --------- "
       """ Function to create the storage, allocating to each rack position
           a sku considering the probabilities and also contains the info
           about the quantity of boxes available in the storage"""

       def Storage(QTY_SKUS, PROB_SKUS):
           storage = []
           for aisle in range (AISLES):
               for level in range(LEVELS):
                   for channel in range(CHANNELS):
                       for side in range (SIDES):
                           sku_stored = numpy.random.choice(numpy.arange(0, QTY_SKUS), p=PROB_SKUS)
                           storage.append([aisle + 1, level + 1, channel + 1 , side + 1, sku_stored, BOXES]) #+1 because it starts on zero
           for n in range (20):
               sku_locations = filter( lambda storage:storage[4] == n, storage)
           return storage



       " ----------- Order -----------"
```

```
    """ Function to create the client order considering the probabilities
        of each sku be ordered and the number of boxes that can go into
        a pallet. Returns a list with the quantity ordered for each sku. """

    def Order(BOXES, QTY_SKUS, PROB_SKUS):
        order = []
        skus =[]
        for sku in range (QTY_SKUS):
            order.append(0) #list of quantities ordered for each SKU - initial value of zero for all SKUs availables in the storage
            skus.append(sku)
        for box in range (BOXES):
            sku_ordered = numpy.random.choice(skus) #for each box that goes into the pallet it is chosen the SKU
            order[sku_ordered] = order[sku_ordered] + 1 #update the list of quantities ordered for the sku ordered
        return order


    " ---------- Vehicle travel time -------------"
    """ Compute travel distance based on the initial and final channel the vehicle is going to be,
        with the distance computes the travel time for the vehicle considering the velocity profile
        and return the travel time for vehicle"""

    def Travel_T_Vehicle (initial_C, final_C):
        V_dist = WIDTH * abs(final_C - initial_C)
        V_travel_time_1 = ((2*VEL_V)/ACC_V + (V_dist - (2*VEL_V**2)/(2*ACC_V))/VEL_V)
        V_travel_time_2 = (2* math.sqrt(V_dist/ACC_V))
        if V_travel_time_2 <= TIME_PEAK_V:
            V_travel_time = V_travel_time_2
        else:
            V_travel_time = V_travel_time_1
        return V_travel_time


    " ---------- Lift travel time -------------"
    """ Compute travel distance based on the initial and final level the lift is going to be,
        with the distance computes the travel time for the lift considering the velocity profile
        and return the travel time for lift"""

    def Travel_T_Lift (initial_L, final_L):
        L_dist = WIDTH * abs(final_L - initial_L)
        L_travel_time_1 = ((2*VEL_L)/ACC_L + (L_dist - (2*VEL_L**2)/(2*ACC_L))/VEL_L)
        L_travel_time_2 = (2* math.sqrt(L_dist/ACC_L))
        if L_travel_time_2 <= TIME_PEAK_L:
            L_travel_time = L_travel_time_2
        else:
            L_travel_time = L_travel_time_1
        return L_travel_time

    " ------------ Order Arrival -------------"
    """ Function creates the client's order and also defines the tier where this order will be created
        and the rack locations from where the products are going to be picked"""

    def Arrival(env, L_level):
        num_order = 1
        count_retrieval = 1
        while True:
```

```

            yield env.timeout(random.expovariate(RETRIEVAL_RATE))

            order = Order(BOXES, QTY_SKUS, PROB_SKUS) #function return the quantity of each product available that was ordered
            storage_list = [] #list that constains the rack locations from where the ordered products are going to be picked
            picking_locations = []
            for sku in range (len(order)):
                qty_ordered = order[sku]
                if qty_ordered > 0: #the list order contains quantity for all SKUs available, even when the value is zero, so first is verified which items were in fact ordered

                    sku_locations = filter( lambda storage:storage[4] == sku and storage[4] == sku, storage) #all rack locations that contains the sku
                    if sku_locations == []:
                        print'PRODUCT UNAVAILABLE'
                        sys.exit() #in case the storage does not contain all ordered items, then it is not going to be possible to continue the simulation
                        # assumption: it is not going to happen

                    else:
                        info_picking = [] #list that contains info for storage location and quantity picked
                        selected_location = random.choice(sku_locations) #randomly select on of the possible rack locations

                        for i in range (5):
                            info_picking.append(selected_location[i])
                        info_picking.append(qty_ordered)
                        picking_locations.append(info_picking) #insert the info for each item into a list
                        selected_location[5] = BOXES #uptdat number of products in pallet stored
            #picking_locations.sort(key = lambda x: (x[1], x[2])) #organize all the rack locations in order by level and channel
            data =  Data(env, num_order, order, env.now, 0, 0, 0, 0) #create the retrieval transaction order (the arrival)
            env.process(Packaging_Area(env, picking_locations, data, num_order, L_level)) #send to the picking process
            num_order = num_order + 1



    def Packaging_Area(env, picking_locations, data, num_order, L_level):
        """ First of all, system verifies the slots available in the packaging area"""

        """ To start build a mixed pallet, an slot needs to be available """
        SMReq = slots_mixed.request()
        yield SMReq
        data.vehicle_I = env.now #started the process to build the pallet

        """ Then, it is verified if the ordered SKU is avalilable in the packaging area, being retrieved or needs to be retrieved """
        count_retrieval = 1
        for location in range (len(picking_locations)):
            """ Is available in the packaging area """

            """ It is already being retrieved """

            """ Needs to be retrieved """

            """" Before execute the retrieval process, an slot for the retrieved pallet is requested"""
```

```
            SRReq = slots_retrieved.request()
            yield SRReq

            """ Then the retrieval process is executed """
            env.process(Retrieval(env, picking_locations[location], picking_locations, data, num_order, count_retrieval, L_level, SMReq, SRReq)) #send to the retrieval process
            count_retrieval = count_retrieval + 1

    def Retrieval(env, picking_locations, all_picking_locations, data, num_order, count_retrieval, L_level, SMReq, SRReq):

        """ First of all request vehicle """
        VReq = vehicles.request()
        yield VReq
        process_V = Processes_Vehicle(env, num_order, count_retrieval, 'Vehicle', env.now, 0, 0)
        V_level = 0 #vehicle is outside the storage area, where it delivers the load, close to the lift. When process starts it goes to the lift
        V_channel = 0

        """ Process occurs once per SKU retrieved, the lenght it is going to be 1 """
        level_picking = picking_locations[1]
        channel_picking = picking_locations[2]
        if level_picking != V_level: #level of the rack location is different the level vehicle is

            """ Vehicle travels from its position to channel 1 to get the lift"""
            initial_C = V_channel
            final_C = 0 #to get the lift, vehicle needs to be at channel 0
            vehicle_travel_time = Travel_T_Vehicle(initial_C, final_C)
            time_I = env.now
            yield env.timeout(vehicle_travel_time) #vehicle travel to channel close to lift
            process = Processes(env, num_order, count_retrieval, 'Vehicle', initial_C, final_C, time_I, env.now, vehicle_travel_time)
            all_processes.append(process)
            initial_C = final_C #update vehicle's position

            """ V_level was defined as 0 to sinalize the operation just started, but in reality it's at level 1 """
            if V_level == 0:
                V_level = 1

            """ When vehicle arrives at channel 0, it requests the lift """
            LReq = lift.request()
            yield LReq
            process_lift = Processes_Lift(env, num_order, count_retrieval, 'Lift', env.now, 0, 0)

            """ Lift travels from its position to the level where vehicle is """
            initial_L = L_level[0]
            final_L = V_level
            lift_travel_time = Travel_T_Lift(initial_L, final_L)
            time_I = env.now
            yield env.timeout(lift_travel_time) #lift travels to level where vehicle is
            process = Processes(env, num_order, count_retrieval, 'Lift', initial_L, final_L, time_I, env.now, lift_travel_time)
            all_processes.append(process)
            L_level[0] = final_L #update lift's position

```

```

                """ When the lift arrives, the vehicle goes into the lift """
                yield env.timeout(TRANSFER_L)


                """ Then, it travels from its current level to the level of the rack location with the product to be picked is  """
                initial_L = L_level[0]
                final_L = level_picking
                lift_travel_time = Travel_T_Lift(initial_L, final_L)
                time_I = env.now
                yield env.timeout(lift_travel_time) #lift transports vehicle from one level to another
                process = Processes(env, num_order, count_retrieval, 'Lift', initial_L, final_L, time_I, env.now, lift_travel_time)
                all_processes.append(process)
                L_level[0] = level_picking #update lift's position

                """ When lift arrives at the picking level, vehicle leaves the lift and goes to channel 0 at that level"""
                yield env.timeout(TRANSFER_L) #interval during which vehicle leaves the lift
                lift.release(LReq) #vehicle is released
                V_channel = 0 #goes to channel 0 when leaves the lift
                V_level = level_picking #update vehicles' position
                process_lift.time_F = env.now
                process_lift.duration = env.now - process_lift.time_I
                all_lift.append(process_lift)



            """ When vehicle is at the level of the rack location with the product to be picked is, it travels from its location to the rack location  """
            initial_C = V_channel
            final_C = channel_picking
            vehicle_travel_time = Travel_T_Vehicle(initial_C, final_C)
            time_I = env.now
            yield env.timeout(vehicle_travel_time) #vehicle travel to channel
            process = Processes(env, num_order, count_retrieval, 'Vehicle', initial_C, final_C, time_I, env.now, vehicle_travel_time)
            all_processes.append(process)
            V_channel = final_C #update vehicle's position


            """ Finally, when vehicle is at the picking location, the pallet is transfered from the storage location to the vehicle """
            yield env.timeout(TRANSFER_S)
            picking_locations[5] = 20



            """ The pallet is picked, vehicle travels back to level 1 and leaves the lift to deliver the ordered pallet"""

            """ Vehicle travels from its position to channel 0 to get the lift"""
            initial_C = V_channel
            final_C = 0 #to get the lift, vehicle needs to be at channel 0
            vehicle_travel_time = Travel_T_Vehicle(initial_C, final_C)
            time_I = env.now
            yield env.timeout(vehicle_travel_time) #vehicle travel to channel
            process = Processes(env, num_order, count_retrieval, 'Vehicle', initial_C, final_C, time_I, env.now, vehicle_travel_time)
            all_processes.append(process)
            V_channel = final_C #update vehicle's position


```

```
        """ When lift arrives at channel 0, it requests the lift """
        LReq = lift.request()
        yield LReq
        process_lift = Processes_Lift(env, num_order, count_retrieval, 'Lift', env.now, 0, 0)


        """ Lift travels from its position to the level where vehicle is """
        initial_L = L_level[0]
        final_L = V_level
        lift_travel_time = Travel_T_Lift(initial_L, final_L)
        time_I = env.now
        yield env.timeout(lift_travel_time) #lift travels to level where vehicle is
        process = Processes(env, num_order, count_retrieval, 'Lift', initial_L, final_L, time_I, env.now, lift_travel_time)
        all_processes.append(process)
        L_level[0] = V_level #update lift's position
        V_level = V_level #update vehicles' position


        """ When the lift arrives, the vehicle goes into the lift """
        yield env.timeout(TRANSFER_L)

        """ Then, it travels from its current level to the level 1 (one)  """
        initial_L = L_level[0]
        final_L = 1
        lift_travel_time = Travel_T_Lift(initial_L, final_L)
        time_I = env.now
        yield env.timeout(lift_travel_time) #lift transports vehicle to level 1
        process = Processes(env, num_order, count_retrieval, 'Lift', initial_L, final_L, time_I, env.now, lift_travel_time)
        all_processes.append(process)
        L_level[0] = final_L #update lift's position



        """ When lift arrives at the picking level, vehicle leaves the lift """
        yield env.timeout(TRANSFER_L) #interval during which vehicle leaves the lift
        lift.release(LReq) #vehicle is released
        process_lift.time_F = env.now
        process_lift.duration = env.now - process_lift.time_I
        all_lift.append(process_lift)

        """ Finally, pallet is transfered to the slot in the packaging area """
        yield env.timeout(TRANSFER_PA) #interval during which vehicle discharges the load in the packaging area and returns to dwell point
        yield env.timeout(TRANSFER_S) #interval during which vehicle discharges the load in the packaging area and returns to dwell point
        vehicles.release(VReq) #vehicle is released
        process_V.time_F = env.now
        process_V.duration = env.now - process_V.time_I
        all_vehicle.append(process_V)

        V_level = final_L #update vehicles' position


        env.process(Build_Pallet(env, picking_locations, all_picking_locations, data, num_order, count_retrieval, L_level, SMReq, SRReq))


    def Build_Pallet(env, picking_locations, all_picking_locations, data, num_order, count_retrieval, L_level, SMReq, SRReq):
        """ Item arrived, it is transferred to the mixed pallet, then pallet is stored again and slot is released """
```

```
            qty_items = picking_locations[5]
            RAReq = robot.request()
            for q in range (qty_items):
                yield env.timeout(ROBOTIC_ARM)
            robot.release(RAReq)
            slots_retrieved.release(SRReq)

            """ Verify if all SKU were picked, if they were pallet is ready to go to client and slot is released """
            total_retrieval = len(all_picking_locations)
            if count_retrieval == total_retrieval: #all retrieval processes were completed, mixed pallet is ready
                slots_mixed.release(SMReq)
                data.time_F = env.now
                data.duration = data.time_F - data.time_I
                data.duration_V = data.time_F - data.vehicle_I
                all_data.append(data)



        " ---------- Record data on Excel file ------------"
        """ Leitura das planilhas excel """
        def recordData(all_data,all_processes, replication):
            tab1 = []
            tab2 = []
            tab3 = []
            tab4 = []
            tab5 = []
            count_retrieval = 0
            WV_sum = 0.0
            LO_sum = 0.0
            VO_sum = 0.0
            V_cycle = 0.0
            total_cycle = 0.0

            for processes in all_processes:
                tab1.append([ processes.num_order, processes.retrieval, processes.resource, processes.position_I, processes.position_F,
                              processes.time_I, processes.time_F, processes.duration])

            for data_vehicle in all_vehicle:
                tab2.append([data_vehicle.num_order, data_vehicle.retrieval, data_vehicle.resource, data_vehicle.time_I, data_vehicle.time_F, data_vehicle.duration])
                if data_vehicle.time_I > WARM_UP:
                    VO_sum = VO_sum + data_vehicle.duration

            for data_lift in all_lift:
                tab3.append([data_lift.num_order, data_lift.retrieval, data_lift.resource, data_lift.time_I, data_lift.time_F, data_lift.duration])
                if data_lift.time_I > WARM_UP:
                    LO_sum = LO_sum + data_lift.duration


            for data in all_data:
                tab4.append([data.retrieval, data.time_I, data.vehicle_I, data.time_F, data.duration_V])
                tab5.append(data.order)
                if data.vehicle_I > WARM_UP:
                    WV_sum = WV_sum + (data.duration - data.duration_V)
                    V_cycle = V_cycle + data.duration_V
                    total_cycle = total_cycle + data.duration
                    count_retrieval = count_retrieval + 1
                    time_max = data.time_F

            V_cycle_mean = V_cycle / count_retrieval
```

```
    total_cycle_mean = total_cycle / count_retrieval
    WV_mean = WV_sum / count_retrieval
    LU = (LO_sum)/(time_max - WARM_UP)
    VU = (VO_sum)/(QTY_VEHICLES*(time_max - WARM_UP))

    i = wb2.sheets.count
    pl1 = wb2.sheets.add('Replication %d Processes' %(replication+1),after=i)

    pl1.range('A10').value=['Order','Retrieval', 'Resource', 'Position_I', 'Position_F', 'Time_I', 'Time_F', 'Duration']
    pl1.range('A11').value=tab1

    pl1.range('J10').value=['Order','Retrieval', 'Resource', 'Time_I', 'Time_F', 'Duration']
    pl1.range('J11').value=tab2

    pl1.range('S10').value=['Order','Retrieval', 'Resource', 'Time_I', 'Time_F', 'Duration']
    pl1.range('S11').value=tab3

    table_order = []
    for sku in range (QTY_SKUS):
        table_order.append('SKU %d' %sku)

    i = wb2.sheets.count
    pl2 = wb2.sheets.add('Replication %d Data' %(replication+1),after=i)

    pl2.range('A10').value = ['Order', 'Arrival', 'Starts', 'Finishes', 'Duration V']
    pl2.range('A11').value = tab4
    pl2.range('G10').value = table_order
    pl2.range('G11').value = tab5

    pl0.range('B%d'%(replication+11)).value=[V_cycle_mean, LU, VU, count_retrieval]


""" --- MAIN FUNCTION ----"""
for a in range (3):
    for b in range (3):
        for c in range (3):
            if a == 0:
                QTY_SKUS = 5
                PROB_SKUS = [0.2]*5
            elif a == 1:
                QTY_SKUS = 10
                PROB_SKUS = [0.1]*10
            elif a == 2:
                QTY_SKUS = 20
                PROB_SKUS = [0.05]*20

            if c == 0:
                ROBOTIC_ARM = 2.0
            elif c == 1:
                ROBOTIC_ARM = 7.0
            elif c == 2:
                ROBOTIC_ARM = 12.0

            if b == 0:
                RETRIEVAL_RATE = 2.0/3600 #retrievals per hour
            elif b == 1:
                RETRIEVAL_RATE = 7.0/3600 #retrievals per hour
            elif b == 2:
                RETRIEVAL_RATE = 15.0/3600 #retrievals per hour

```

```
                    wb2 = xw.Book()
                    pl0=wb2.sheets[0]
                    pl0.name='Summary'
                    pl0.range('A4').value=['Retrieval rate', RETRIEVAL_RATE*3600,' ', 'ROBOT', ROBOTIC_ARM, ' ', 'QTY SKUS',QTY_SKUS]
                    pl0.range('A5').value=['VEHICLES', QTY_VEHICLES]
                    pl0.range('B10').value=['V Cycle', 'Utilisation Lift', 'Utilisation Vehicle', 'Orders']

                    for replication in range (QTY_REPLICATION):
                        pl0.range('A%d'%(replication+11)).value=['Retrieval %d' %(replication+1)]
                        env = simpy.Environment()

                        #Lists to collect information
                        all_data = []
                        all_processes = []
                        all_vehicle = []
                        all_lift = []

                        # Creation of the storage
                        storage = Storage(QTY_SKUS, PROB_SKUS)

                        #Create resources
                        lift, vehicles, slots_mixed, slots_retrieved, robot = Resources(env, QTY_LIFTS, QTY_VEHICLES, SLOTS_PA_MIXED, SLOTS_PA_RETRIEVED )

                        #Every repication lift starts in the first level
                        L_level = [1]

                        # Run the simulation
                        env.process(Arrival(env, L_level))
                        env.run(until=183600)

                        recordData(all_data,all_processes, replication)
```

2. Eurfork's system simulation model

```
# -*- coding: utf-8 -*-
"""
New system - Eurofork's system
"""
import simpy
import math
import random
import numpy
import xlwings as xw

" -------- Fixed Variables for the scenario -------- "
CHANNELS = 19
LEVELS = 6
AISLES = 1
SIDES = 2
QTY_LIFTS = 1
VEL_V = 2.0 #velocity vehicle
ACC_V = 0.5 #acceleration vehicle
TIME_PEAK_V = (2* VEL_V) / ACC_V #interval to vehicle reaches the peak velocity
VEL_L = 14.0/60 #velocity lift
ACC_L = 0.3 #acceleration lift
TIME_PEAK_L = (2* VEL_L) / ACC_L #interval to lift reaches the peak velocity
WIDTH = 1.3
HEIGHT = 2.0
```

```
TRANSFER_S = 14.0 #time to transfer load to shuttle, from Ekhren and Heragu (2011)
TRANSFER_L = 0.0 #time to transfer shuttle to lift,  from Ekhren and Heragu (2011)
WARM_UP = 10800

" --------- Variables for each scenario ---------- "
QTY_REPLICATION = 10
BOXES = 20
QTY_VEHICLES = 1




" -------------- Class Processes ---------------------"
class Processes:
    def __init__(self, env, retrieval, resource, position_I, position_F, time_I, time_F, duration):
        self.retrieval = retrieval
        self.resource = resource
        self.position_I = position_I
        self.position_F = position_F
        self.time_I = time_I
        self.time_F = time_F
        self.duration = duration

" -------------- Class Processes Total Vehicle ---------------------"
class Processes_Vehicle:
    def __init__(self, env, retrieval, resource, time_I, time_F, duration):
        self.retrieval = retrieval
        self.resource = resource
        self.time_I = time_I
        self.time_F = time_F
        self.duration = duration

" -------------- Class Processes Total Lift ---------------------"
class Processes_Lift:
    def __init__(self, env, retrieval, resource, time_I, time_F, duration):
        self.retrieval = retrieval
        self.resource = resource
        self.time_I = time_I
        self.time_F = time_F
        self.duration = duration


" -------------- Class Data ---------------------"
class Data:
    def __init__(self, env, retrieval, order, time_I, vehicle_I, time_F, duration, duration_V):
        self.retrieval = retrieval
        self.order = order
        self.time_I = time_I
        self.vehicle_I = vehicle_I
        self.time_F = time_F
        self.duration = duration
        self.duration_V = duration_V


" -------- Resources ----------- "
def Resources(env, QTY_LIFTS, QTY_VEHICLES):
    lift = simpy.Resource(env, QTY_LIFTS)
    vehicles = simpy.Resource(env, QTY_VEHICLES)
    return lift, vehicles


" -------- Storage creation --------- "
""" Function to create the storage, allocating to
    each rack position a sku considering the probabilities
```

```
    and also contains the info about the quantity of boxes
    available in the storage"""

def Storage(QTY_SKUS, PROB_SKUS):
    storage = []
    #seed = 0
    for aisle in range (AISLES):
        for level in range(LEVELS):
            for channel in range(CHANNELS):
                    for side in range (SIDES):
                        sku_stored = numpy.random.choice(numpy.arange(0, QTY_SKUS), p=PROB_SKUS)
                        storage.append([aisle + 1, level + 1, channel + 1 , side + 1, sku_stored, BOXES]) #+1 because it starts on zero
        for n in range (20):
            sku_locations = filter( lambda storage:storage[4] == n, storage)
            print len(sku_locations)
        return storage


    " ----------- Order -----------"
    """ Function to create the client order considering the probabilities
        of each sku be ordered and the number of boxes that can go into
        a pallet. Returns a list with the quantity ordered for each sku. """

    def Order(BOXES, QTY_SKUS, PROB_SKUS):
        order = []
        for sku in range (QTY_SKUS):
            order.append(0) #list of quantities ordered for each SKU - initial value of zero for all SKUs availables in the storage
        for box in range (BOXES):
            sku_ordered = numpy.random.choice(numpy.arange(0, QTY_SKUS), p = PROB_SKUS) #for each box that goes into the pallet it is chosen the SKU
            order[sku_ordered] = order[sku_ordered] + 1 #update the list of quantities ordered for the sku ordered
        return order


    " ---------- Vehicle travel time -------------"
    """ Compute travel distance based on the initial and final channel the vehicle is going to be,
        with the distance computes the travel time for the vehicle considering the velocity profile
        and return the travel time for vehicle"""

    def Travel_T_Vehicle (initial_C, final_C):
        V_dist = WIDTH * abs(final_C - initial_C)
        V_travel_time_1 = ((2*VEL_V)/ACC_V + (V_dist - (2*VEL_V**2)/(2*ACC_V))/VEL_V)
        V_travel_time_2 = (2* math.sqrt(V_dist/ACC_V))
        if V_travel_time_2 <= TIME_PEAK_V:
            V_travel_time = V_travel_time_2
        else:
            V_travel_time = V_travel_time_1
        return V_travel_time


    " ---------- Lift travel time -------------"
    """ Compute travel distance based on the initial and final level the lift is going to be,
        with the distance computes the travel time for the lift considering the velocity profile
        and return the travel time for lift"""

    def Travel_T_Lift (initial_L, final_L):
```

```
        L_dist = WIDTH * abs(final_L - initial_L)
        L_travel_time_1 = ((2*VEL_L)/ACC_L + (L_dist - (2*VEL_L**2)/(2*ACC_L))/VEL_L)
        L_travel_time_2 = (2* math.sqrt(L_dist/ACC_L))
        if L_travel_time_2 <= TIME_PEAK_L:
            L_travel_time = L_travel_time_2
        else:
            L_travel_time = L_travel_time_1
        return L_travel_time

    " ------------ Order Arrival -------------"
    """ Function creates the client's order and also defines the tier where this order will be created
        and the rack locations from where the products are going to be picked"""

    def Arrival(env, L_level):
        count_retrieval = 1
        while env.now<183600:

            yield env.timeout(random.expovariate(RETRIEVAL_RATE))

            order = Order(BOXES, QTY_SKUS, PROB_SKUS) #function return the quantity of each product available that was ordered

            storage_list = [] #list that constains the rack locations from where the ordered products are going to be picked

            picking_locations = []
            for sku in range (len(order)):
                qty_ordered = order[sku]
                if qty_ordered > 0: #the list order contains quantity for all SKUs available, even when the value is zero, so first is verified which items were in fact ordered
                    missing_items = qty_ordered
                    while missing_items > 0:
                        sku_locations = filter( lambda storage:storage[4] == sku, storage) #all rack locations that contains the sku
                        if sku_locations == []:
                            print'PRODUCT UNAVAILABLE'
                            print count_retrieval
                            print sku
                            print storage
                            for n in range (20):
                                sku_locations = filter( lambda storage:storage[4] == n, storage)
                                print len(sku_locations)
                            sys.exit() #in case the storage does not contain all ordered items, then it is not going to be possible to continue the simulation
                            # assumption: it is not going to happen

                        else:
                            info_picking = [] #list that contains info for storage location and quantity picked
                            selected_location = random.choice(sku_locations) #randomly select on of the possible rack locations
                            qty_available = selected_location[5]
                            if qty_available <= missing_items:
                                """if the quantity available in the storage is less than the ordered,
                                first all items available in the pallet are picked,the pallet in the
                                second bay becomes available and vehicle travels to another rack
                                location to pick the missing proudcts"""
                                for i in range (6):
```

```
                                info_picking.append(selected_location[i])
                            picking_locations.append(info_picking) #insert the info for each item into a list
                            #selected_location[4] = random.choice([0,1, 2, 3, 4, 5, 6, 7, 8, 9]) #update the product available in the rack
                            selected_location[5] = BOXES #new pallet - it is full
                            missing_items = missing_items - qty_available

                        else:
                            for i in range (5):
                                info_picking.append(selected_location[i])
                            info_picking.append(missing_items)
                            picking_locations.append(info_picking) #insert the info for each item into a list

                            selected_location[5] = selected_location[5] - missing_items #update the quantity available in the pallet stored

                            missing_items = 0 #all items were picked


        picking_locations.sort(key = lambda x: (x[1], x[2])) #organize all the rack locations in order by level and channel
        data =  Data(env, count_retrieval, order, env.now, 0, 0, 0, 0) #create the retrieval transaction order (the arrival)

        env.process(Picking(env, picking_locations, data, count_retrieval, L_level)) #send to the picking process

        count_retrieval = count_retrieval + 1

def Picking(env, picking_locations, data, count_retrieval, L_level):

    """ First of all request vehicle """
    VReq = vehicles.request()
    yield VReq
    process_V = Processes_Vehicle(env, count_retrieval, 'Vehicle',  env.now, 0, 0)
    data.vehicle_I = env.now

    V_level = 0 #vehicle is outside the storage area, where it delivers the load, close to the lift. When process starts it goes to the lift
    V_channel = 0

    """ Process occurs until all items are collected """
    for location in range (len(picking_locations)):
        level_picking = picking_locations[location][1]
        channel_picking = picking_locations[location][2]
        if level_picking != V_level: #level of the rack location is different the level vehicle is

            """ Vehicle travels from its position to channel 1 to get the lift"""
            initial_C = V_channel
            final_C = 0 #to get the lift, vehicle needs to be at channel 0

            vehicle_travel_time = Travel_T_Vehicle(initial_C, final_C)
            time_I = env.now
            yield env.timeout(vehicle_travel_time) #vehicle travel to channel

            process = Processes(env, count_retrieval, 'Vehicle', initial_C, final_C, time_I, env.now, vehicle_travel_time)
            all_processes.append(process)
            initial_C = final_C #update vehicle's position

```

```python
                    """ V_level was defined as 0 to sinalize the operation just started, but in reality it's at level 1 """
                    if V_level == 0:
                        V_level = 1

                    """ When vehicle arrives at chanel 0, it requests the lift """
                    LReq = lift.request()
                    yield LReq
                    process_lift = Processes_Lift(env, count_retrieval, 'Lift', env.now, 0, 0)


                    """ Lift travels from its position to the level where vehicle is """
                    initial_L = L_level[0]
                    final_L = V_level
                    lift_travel_time = Travel_T_Lift(initial_L, final_L)
                    time_I = env.now
                    yield env.timeout(lift_travel_time) #lift travels to level where vehicle is
                    process = Processes(env, count_retrieval, 'Lift', initial_L, final_L, time_I, env.now, lift_travel_time)
                    all_processes.append(process)
                    L_level[0] = final_L #update lift's position


                    """ When the lift arrives, the vehicle goes into the lift """
                    yield env.timeout(TRANSFER_L)


                    """ Then, it travels from its current level to the level of the rack location with the product to be picked is  """
                    initial_L = L_level[0]
                    final_L = level_picking
                    lift_travel_time = Travel_T_Lift(initial_L, final_L)
                    time_I = env.now
                    yield env.timeout(lift_travel_time) #lift transports vehicle from one level to another
                    process = Processes(env, count_retrieval, 'Lift', initial_L, final_L, time_I, env.now, lift_travel_time)
                    all_processes.append(process)
                    L_level[0] = level_picking #update lift's position

                    """ When lift arrives at the picking level, vehicle leaves the lift and goes to channel 0 at that level"""
                    yield env.timeout(TRANSFER_L) #interval during which vehicle leaves the lift
                    lift.release(LReq) #vehicle is released
                    V_channel = 0 #goes to channel 0 when leaves the lift
                    V_level = level_picking #update vehicles' position        process_lift.time_F = env.now
                    process_lift.time_F = env.now
                    process_lift.duration = env.now - process_lift.time_I
                    all_lift.append(process_lift)


                """ When vehicle is at the level of the rack location with the product to be picked is, it travels from its location to the rack location  """
                initial_C = V_channel
                final_C = channel_picking
                vehicle_travel_time = Travel_T_Vehicle(initial_C, final_C)
                time_I = env.now
                yield env.timeout(vehicle_travel_time) #vehicle travel to channel
                process = Processes(env, count_retrieval, 'Vehicle', initial_C, final_C, time_I, env.now, vehicle_travel_time)
                all_processes.append(process)
```

```
            V_channel = final_C #update vehicle's position


            """ Finally, when vehicle is at the picking location, the robotic arm transfer the items from the stored pallet to the vehicle's pallet """
            quantity = picking_locations[location][5] #verify how many boxes are going to be picked
            transfer_items_time = ROBOTIC_ARM * quantity
            yield env.timeout(transfer_items_time)


        """ When all items are collected, vehicle travels back to level 1 and leaves the lift to deliver the ordered pallet"""

        """ Vehicle travels from its position to channel 1 to get the lift"""
        initial_C = V_channel
        final_C = 0 #to get the lift, vehicle needs to be at channel 0
        vehicle_travel_time = Travel_T_Vehicle(initial_C, final_C)
        time_I = env.now
        yield env.timeout(vehicle_travel_time) #vehicle travel to channel
        process = Processes(env, count_retrieval, 'Vehicle', initial_C, final_C, time_I, env.now, vehicle_travel_time)
        all_processes.append(process)
        V_channel = final_C #update vehicle's position


        """ When lift arrives at chanel 0, it requests the lift """
        LReq = lift.request()
        yield LReq
        process_lift = Processes_Lift(env, count_retrieval, 'Lift', env.now, 0, 0)


        """ Lift travels from its position to the level where vehicle is """
        initial_L = L_level[0]
        final_L = V_level
        lift_travel_time = Travel_T_Lift(initial_L, final_L)
        time_I = env.now
        yield env.timeout(lift_travel_time) #lift travels to level where vehicle is
        process = Processes(env, count_retrieval, 'Lift', initial_L, final_L, time_I, env.now, lift_travel_time)
        all_processes.append(process)
        L_level[0] = V_level #update lift's position
        V_level = V_level #update vehicles' position


        """ When the lift arrives, the vehicle goes into the lift """
        yield env.timeout(TRANSFER_L)

        """ Then, it travels from its current level to the level 1 (one)  """
        initial_L = L_level[0]
        final_L = 1
        lift_travel_time = Travel_T_Lift(initial_L, final_L)
        time_I = env.now
        yield env.timeout(lift_travel_time) #lift transports vehicle to level 1
        process = Processes(env, count_retrieval, 'Lift', initial_L, final_L, time_I, env.now, lift_travel_time)
        all_processes.append(process)
        L_level[0] = final_L #update lift's position


        """ When lift arrives at the picking level, vehicle leaves the lift - ordered pallet is ready, vehicle is released"""
        yield env.timeout(TRANSFER_L) #interval during which vehicle leaves the lift
```

```
        lift.release(LReq) #vehicle is released
        vehicles.release(VReq) #vehicle is released
        V_level = final_L #update vehicles' position
        process_lift.time_F = env.now
        process_lift.duration = env.now - process_lift.time_I
        all_lift.append(process_lift)


        process_V.time_F = env.now
        process_V.duration = env.now - process_V.time_I
        all_vehicle.append(process_V)

        data.time_F = env.now
        data.duration = data.time_F - data.time_I
        data.duration_V = data.time_F - data.vehicle_I
        all_data.append(data)


    " ---------- Record data on Excel file ------------"
    """ Leitura das planilhas excel """
    def recordData(all_data,all_processes, replication):
        tab1 = []
        tab2 = []
        tab3 = []
        tab4 = []
        tab5 = []
        count_retrieval = 0
        WV_sum = 0
        LO_sum = 0
        VO_sum = 0
        V_cycle = 0
        total_cycle = 0

        for processes in all_processes:
            tab1.append([processes.retrieval, processes.resource, processes.position_I, processes.position_F,
                          processes.time_I, processes.time_F, processes.duration])

        for data_vehicle in all_vehicle:
            tab2.append([data_vehicle.retrieval, data_vehicle.resource, data_vehicle.time_I, data_vehicle.time_F, data_vehicle.duration])
            if data_vehicle.time_I > WARM_UP:
                VO_sum = VO_sum + data_vehicle.duration

        for data_lift in all_lift:
            tab3.append([data_lift.retrieval, data_lift.resource, data_lift.time_I, data_lift.time_F, data_lift.duration])
            if data_lift.time_I > WARM_UP:
                LO_sum = LO_sum + data_lift.duration

        for data in all_data:
            tab4.append([data.retrieval, data.time_I, data.vehicle_I, data.time_F, data.duration, data.duration_V])
            tab5.append(data.order)
            if data.vehicle_I > WARM_UP:
                WV_sum = WV_sum + (data.duration - data.duration_V)
                V_cycle = V_cycle + data.duration_V
                total_cycle = total_cycle + data.duration
                count_retrieval = count_retrieval + 1
                time_max = data.time_F


        V_cycle_mean = V_cycle / count_retrieval
        total_cycle_mean = total_cycle / count_retrieval
        WV_mean = WV_sum / count_retrieval
```

```
        LU = (LO_sum)/(time_max - WARM_UP)
        VU = (VO_sum)/(QTY_VEHICLES*(time_max - WARM_UP))

        i = wb2.sheets.count
        pl1 = wb2.sheets.add('Replication %d Processes' %(replication+1),after=i)

        pl1.range('A10').value=['Order', 'Resource', 'Position_I', 'Position_F', 'Time_I', 'Time_F', 'Duration']
        pl1.range('A11').value=tab1

        pl1.range('J10').value=['Retrieval', 'Resource', 'Time_I', 'Time_F', 'Duration']
        pl1.range('J11').value=tab2

        pl1.range('R10').value=['Retrieval', 'Resource', 'Time_I', 'Time_F', 'Duration']
        pl1.range('R11').value=tab3

        table_order = []
        for sku in range (QTY_SKUS):
            table_order.append('SKU %d' %sku)

        i = wb2.sheets.count
        pl2 = wb2.sheets.add('Replication %d Data' %(replication+1),after=i)

        pl2.range('A10').value = ['Order', 'Arrival', 'Starts', 'Finishes', 'Duration', 'Duration V']
        pl2.range('A11').value = tab4
        pl2.range('G10').value = table_order
        pl2.range('G11').value = tab5

        pl0.range('B%d'%(replication+11)).value=[V_cycle_mean, total_cycle_mean, WV_mean, LU, VU, count_retrieval]


    """ --- MAIN FUNCTION ----"""
    for a in range (3):
        for b in range (3):
            for c in range (3):
                if a == 0:
                    QTY_SKUS = 5
                    PROB_SKUS = [0.2]*5
                elif a == 1:
                    QTY_SKUS = 10
                    PROB_SKUS = [0.1]*10
                elif a == 2:
                    QTY_SKUS = 20
                    PROB_SKUS = [0.05]*20

                if c == 0:
                    ROBOTIC_ARM = 2.0
                elif c == 1:
                    ROBOTIC_ARM = 7.0
                elif c == 2:
                    ROBOTIC_ARM = 12.0

                if b == 0:
                    RETRIEVAL_RATE = 2.0/3600 #retrievals per hour
                elif b == 1:
                    RETRIEVAL_RATE = 7.0/3600 #retrievals per hour
                elif b == 2:
                    RETRIEVAL_RATE = 15.0/3600 #retrievals per hour


                wb2 = xw.Book()
```

```python
                pl0=wb2.sheets[0]
                pl0.name='Summary'
                pl0.range('A5').value=['Retrieval rate', RETRIEVAL_RATE*3600, 'VEHICLES', QTY_VEHICLES, 'ROBOT', ROBOTIC_ARM, 'QTY SKUS',QTY_SKUS,'PROB UNIFORM']
                pl0.range('B10').value=['V Cycle', 'Total Cycle', 'Waiting Vehicle', 'Utilisation Lift', 'Utilisation Vehicle', 'Orders']
                for replication in range (QTY_REPLICATION):
                    pl0.range('A%d'%(replication+11)).value=['Retrieval %d' %(replication+1)]
                    env = simpy.Environment()

                    #Lists to collect information
                    all_data = []
                    all_processes = []
                    al_lift = []
                    all_vehicle = []
                    all_lift = []

                    # Creation of the storage
                    storage = Storage(QTY_SKUS, PROB_SKUS)

                    #Create resources
                    lift, vehicles = Resources(env, QTY_LIFTS, QTY_VEHICLES)

                    #Every repication lift starts in the first level
                    L_level = [1]

                    # Run the simulation
                    env.process(Arrival(env, L_level))
                    env.run(until=183600)

                    recordData(all_data,all_processes, replication)
```

3. Design of Experiment Results

Scenario	Qty SKU	Arrival Rate	Robot time	CURRENT SYSTEM				NEW SYSTEM			
				Cycle	Util. Lift	Util Vehicle	Orders	Cycle	Util. Lift	Util Vehicle	Orders
1	5	2	2	513.07	0.09	0.24	106	175.48	0.03	0.10	100
2	5	2	2	512.43	0.08	0.22	99	181.52	0.02	0.08	71
3	5	2	2	500.53	0.09	0.23	102	178.29	0.03	0.10	94
4	5	2	2	503.13	0.07	0.20	87	176.03	0.03	0.10	94
5	5	2	2	477.54	0.08	0.23	104	178.39	0.04	0.12	118
6	5	2	2	537.94	0.10	0.26	114	179.53	0.03	0.11	101
7	5	2	2	516.87	0.08	0.23	100	176.87	0.03	0.09	90
8	5	2	2	501.61	0.09	0.24	105	178.28	0.03	0.09	88
9	5	2	2	510.28	0.09	0.23	101	176.97	0.03	0.11	109
10	5	2	2	497.14	0.08	0.21	94	176.80	0.03	0.10	99
11	5	2	7	659.71	0.09	0.24	108	279.38	0.02	0.13	78
12	5	2	7	672.90	0.08	0.20	88	277.10	0.03	0.14	87
13	5	2	7	643.40	0.08	0.21	91	272.68	0.03	0.14	85
14	5	2	7	641.09	0.08	0.24	105	281.34	0.03	0.18	108
15	5	2	7	689.64	0.08	0.20	87	276.52	0.02	0.13	81
16	5	2	7	706.82	0.07	0.19	84	275.32	0.03	0.16	102
17	5	2	7	646.89	0.07	0.20	90	275.66	0.03	0.15	95

18	5	2	7	575.12	0.07	0.18	80	280.96	0.03	0.13	81
19	5	2	7	624.64	0.09	0.23	106	276.87	0.03	0.17	104
20	5	2	7	688.77	0.09	0.25	109	274.52	0.03	0.15	93
21	5	2	12	820.00	0.08	0.21	89	374.41	0.03	0.23	108
22	5	2	12	852.67	0.10	0.26	114	381.10	0.02	0.18	79
23	5	2	12	760.74	0.08	0.21	95	377.49	0.03	0.20	91
24	5	2	12	761.77	0.10	0.26	113	382.95	0.03	0.22	94
25	5	2	12	738.98	0.08	0.21	98	380.61	0.02	0.16	72
26	5	2	12	871.91	0.08	0.21	90	375.71	0.03	0.18	81
27	5	2	12	805.27	0.08	0.22	94	379.63	0.03	0.22	95
28	5	2	12	768.01	0.08	0.23	104	380.84	0.03	0.23	99
29	5	2	12	774.30	0.08	0.22	98	377.80	0.03	0.23	106
30	5	2	12	785.31	0.08	0.21	95	378.95	0.03	0.23	104
31	5	7	2	720.04	0.26	0.70	313	182.20	0.10	0.34	320
32	5	7	2	737.65	0.27	0.73	327	183.84	0.10	0.33	313
33	5	7	2	760.79	0.30	0.78	350	184.85	0.10	0.34	314
34	5	7	2	779.45	0.30	0.82	371	183.91	0.11	0.37	344
35	5	7	2	759.77	0.28	0.74	327	184.79	0.10	0.34	317
36	5	7	2	695.44	0.25	0.68	303	186.40	0.09	0.33	306
37	5	7	2	745.74	0.28	0.75	333	186.07	0.11	0.37	346
38	5	7	2	753.25	0.26	0.70	310	182.02	0.11	0.37	349
39	5	7	2	744.55	0.28	0.72	324	182.69	0.11	0.36	338
40	5	7	2	738.77	0.27	0.74	328	183.83	0.10	0.35	324
41	5	7	7	1028.68	0.27	0.72	322	283.94	0.11	0.59	356
42	5	7	7	1030.77	0.26	0.70	307	286.24	0.10	0.52	315
43	5	7	7	1044.80	0.27	0.73	329	284.15	0.11	0.61	367
44	5	7	7	999.38	0.25	0.68	302	285.83	0.11	0.60	360
45	5	7	7	1048.40	0.27	0.73	328	282.87	0.11	0.57	346
46	5	7	7	1041.25	0.27	0.73	329	285.36	0.11	0.59	356
47	5	7	7	1048.35	0.28	0.73	323	282.07	0.10	0.51	311
48	5	7	7	1024.73	0.27	0.71	316	284.67	0.11	0.60	364
49	5	7	7	1023.06	0.26	0.70	316	282.96	0.10	0.53	325
50	5	7	7	1061.86	0.28	0.74	323	286.25	0.10	0.54	326
51	5	7	12	1233.62	0.23	0.61	274	386.11	0.10	0.72	323
52	5	7	12	1240.76	0.23	0.62	276	384.85	0.11	0.80	357
53	5	7	12	1257.46	0.24	0.62	273	383.37	0.10	0.73	330
54	5	7	12	1259.17	0.23	0.62	273	384.05	0.11	0.76	342
55	5	7	12	1257.22	0.23	0.62	273	383.81	0.11	0.80	359
56	5	7	12	1245.43	0.23	0.62	276	382.42	0.10	0.72	323
57	5	7	12	1251.79	0.23	0.62	274	385.38	0.11	0.74	332
58	5	7	12	1249.44	0.23	0.62	274	385.46	0.10	0.73	326
59	5	7	12	1244.61	0.23	0.62	274	384.79	0.11	0.78	349
60	5	7	12	1242.91	0.23	0.61	275	384.05	0.09	0.66	295
61	5	15	2	843.03	0.34	0.90	408	186.18	0.25	0.85	785
62	5	15	2	843.73	0.33	0.91	407	185.55	0.22	0.78	722

63	5	15	2	846.79	0.34	0.91	406	186.36	0.22	0.77	711
64	5	15	2	845.62	0.34	0.91	407	186.87	0.23	0.81	749
65	5	15	2	847.58	0.34	0.91	405	185.88	0.21	0.72	673
66	5	15	2	848.10	0.34	0.91	405	185.96	0.22	0.75	699
67	5	15	2	847.29	0.34	0.91	406	186.35	0.21	0.74	690
68	5	15	2	854.69	0.34	0.91	402	184.00	0.22	0.75	703
69	5	15	2	851.13	0.34	0.91	405	186.99	0.22	0.77	708
70	5	15	2	852.61	0.35	0.91	403	185.67	0.22	0.78	728
71	5	15	7	1050.07	0.28	0.73	327	285.29	0.19	1.00	605
72	5	15	7	1057.16	0.28	0.74	325	285.18	0.19	1.00	605
73	5	15	7	1034.56	0.27	0.73	333	286.71	0.19	1.00	602
74	5	15	7	1053.29	0.28	0.73	327	285.34	0.19	1.00	604
75	5	15	7	1052.66	0.28	0.74	326	286.77	0.19	1.00	602
76	5	15	7	1048.05	0.27	0.73	328	286.28	0.19	1.00	601
77	5	15	7	1053.15	0.27	0.73	326	287.07	0.19	1.00	601
78	5	15	7	1043.14	0.27	0.73	329	287.38	0.19	1.00	600
79	5	15	7	1042.77	0.27	0.73	329	287.54	0.19	1.00	599
80	5	15	7	1040.13	0.27	0.73	330	284.44	0.19	1.00	607
81	5	15	12	1256.60	0.24	0.62	274	384.57	0.14	1.00	448
82	5	15	12	1244.94	0.23	0.62	275	384.53	0.14	1.00	448
83	5	15	12	1262.75	0.24	0.62	271	385.13	0.14	1.00	447
84	5	15	12	1245.53	0.23	0.62	275	386.08	0.14	1.00	446
85	5	15	12	1262.16	0.24	0.62	273	385.87	0.14	1.00	446
86	5	15	12	1244.72	0.23	0.61	276	386.35	0.14	1.00	446
87	5	15	12	1250.39	0.23	0.62	274	385.97	0.14	1.00	446
88	5	15	12	1246.72	0.23	0.62	275	385.30	0.14	1.00	447
89	5	15	12	1255.34	0.23	0.62	273	384.75	0.14	1.00	448
90	5	15	12	1250.63	0.23	0.62	275	384.91	0.14	1.00	448
91	10	2	2	1065.71	0.18	0.47	118	225.84	0.04	0.14	106
92	10	2	2	1010.66	0.14	0.38	96	222.57	0.03	0.12	91
93	10	2	2	970.58	0.11	0.30	74	226.03	0.03	0.12	92
94	10	2	2	998.75	0.16	0.42	105	226.39	0.03	0.11	85
95	10	2	2	1036.16	0.14	0.39	98	227.16	0.03	0.14	103
96	10	2	2	981.70	0.15	0.39	97	220.83	0.03	0.12	93
97	10	2	2	967.50	0.14	0.37	92	225.58	0.03	0.11	82
98	10	2	2	975.13	0.14	0.38	97	226.13	0.03	0.13	98
99	10	2	2	951.82	0.14	0.37	89	224.37	0.03	0.12	92
100	10	2	2	1086.53	0.17	0.47	116	225.81	0.03	0.12	89
101	10	2	7	1076.39	0.15	0.39	94	327.89	0.03	0.19	99
102	10	2	7	1142.21	0.13	0.35	89	322.71	0.03	0.15	80
103	10	2	7	1196.75	0.15	0.40	96	327.19	0.03	0.16	85
104	10	2	7	1213.15	0.16	0.43	106	326.45	0.03	0.17	87
105	10	2	7	1104.78	0.13	0.34	84	325.43	0.03	0.17	88
106	10	2	7	1257.33	0.17	0.45	111	327.08	0.03	0.20	104
107	10	2	7	1170.77	0.15	0.41	102	320.28	0.03	0.19	103

108	10	2	7	1159.11	0.13	0.35	86	321.42	0.03	0.19	100
109	10	2	7	1018.48	0.12	0.31	76	323.14	0.04	0.22	113
110	10	2	7	1280.61	0.14	0.39	98	324.48	0.03	0.18	95
111	10	2	12	1376.92	0.16	0.44	109	424.05	0.03	0.24	99
112	10	2	12	1308.51	0.12	0.32	83	424.41	0.03	0.24	99
113	10	2	12	1321.74	0.14	0.38	93	424.67	0.03	0.26	103
114	10	2	12	1408.92	0.14	0.38	91	425.54	0.03	0.25	101
115	10	2	12	1337.91	0.16	0.43	108	425.28	0.03	0.25	101
116	10	2	12	1340.93	0.12	0.33	80	427.20	0.03	0.24	95
117	10	2	12	1399.83	0.14	0.37	92	424.41	0.03	0.23	93
118	10	2	12	1342.90	0.13	0.35	82	425.90	0.03	0.23	95
119	10	2	12	1339.63	0.13	0.35	88	423.38	0.03	0.25	100
120	10	2	12	1440.07	0.16	0.44	110	425.78	0.04	0.27	108
121	10	7	2	1461.50	0.36	0.95	235	228.37	0.10	0.41	307
122	10	7	2	1468.47	0.36	0.95	233	227.78	0.11	0.46	348
123	10	7	2	1444.64	0.35	0.95	237	229.00	0.11	0.43	324
124	10	7	2	1444.83	0.35	0.95	237	228.64	0.11	0.43	322
125	10	7	2	1447.40	0.35	0.95	236	226.86	0.11	0.45	339
126	10	7	2	1468.97	0.36	0.95	233	227.26	0.10	0.40	307
127	10	7	2	1450.62	0.35	0.95	236	225.49	0.11	0.43	326
128	10	7	2	1473.25	0.36	0.95	233	227.24	0.11	0.43	325
129	10	7	2	1474.60	0.35	0.95	233	229.37	0.12	0.46	348
130	10	7	2	1444.58	0.35	0.94	237	228.77	0.12	0.50	375
131	10	7	7	1641.13	0.31	0.83	209	327.67	0.12	0.66	345
132	10	7	7	1648.77	0.31	0.83	208	328.51	0.12	0.68	358
133	10	7	7	1632.00	0.31	0.83	210	329.41	0.12	0.67	350
134	10	7	7	1643.38	0.31	0.83	208	328.75	0.10	0.58	302
135	10	7	7	1659.25	0.31	0.83	207	328.51	0.10	0.57	300
136	10	7	7	1623.35	0.31	0.83	211	329.49	0.11	0.65	338
137	10	7	7	1637.41	0.31	0.83	210	330.98	0.11	0.64	336
138	10	7	7	1645.95	0.30	0.83	207	326.81	0.11	0.63	330
139	10	7	7	1652.88	0.31	0.84	206	327.15	0.11	0.64	340
140	10	7	7	1659.08	0.32	0.83	207	331.10	0.11	0.66	345
141	10	7	12	1836.25	0.27	0.74	186	429.97	0.11	0.83	333
142	10	7	12	1822.87	0.27	0.74	187	426.33	0.12	0.90	365
143	10	7	12	1862.97	0.28	0.74	184	429.29	0.12	0.86	347
144	10	7	12	1844.32	0.28	0.74	186	427.88	0.11	0.80	321
145	10	7	12	1826.74	0.28	0.74	187	426.86	0.11	0.85	345
146	10	7	12	1860.55	0.28	0.74	184	428.36	0.12	0.86	346
147	10	7	12	1834.26	0.28	0.74	186	430.49	0.10	0.76	304
148	10	7	12	1847.19	0.27	0.75	184	429.16	0.12	0.88	353
149	10	7	12	1865.96	0.28	0.74	183	429.87	0.11	0.85	341
150	10	7	12	1842.98	0.27	0.74	186	427.16	0.11	0.83	335
151	10	15	2	1438.10	0.36	0.94	238	228.67	0.24	0.93	705
152	10	15	2	1445.43	0.35	0.94	237	228.14	0.24	0.97	735

153	10	15	2	1448.31	0.36	0.95	237	230.11	0.23	0.93	697
154	10	15	2	1437.30	0.35	0.95	238	229.88	0.24	0.95	711
155	10	15	2	1460.66	0.36	0.95	235	229.82	0.23	0.93	698
156	10	15	2	1444.84	0.35	0.94	238	229.70	0.23	0.93	696
157	10	15	2	1428.67	0.35	0.95	240	229.07	0.24	0.95	712
158	10	15	2	1449.51	0.35	0.95	237	229.06	0.24	0.95	720
159	10	15	2	1447.95	0.36	0.95	236	229.39	0.24	0.95	718
160	10	15	2	1438.63	0.36	0.94	239	229.09	0.25	0.99	745
161	10	15	7	1680.78	0.32	0.84	204	327.38	0.17	1.00	527
162	10	15	7	1641.68	0.31	0.83	209	330.76	0.17	1.00	522
163	10	15	7	1645.59	0.31	0.83	208	326.95	0.17	1.00	528
164	10	15	7	1659.12	0.31	0.83	207	327.15	0.18	1.00	527
165	10	15	7	1648.77	0.31	0.83	207	330.56	0.17	1.00	521
166	10	15	7	1642.44	0.31	0.84	208	328.11	0.18	1.00	526
167	10	15	7	1633.35	0.31	0.83	209	329.57	0.17	1.00	523
168	10	15	7	1636.78	0.31	0.83	209	327.55	0.18	1.00	527
169	10	15	7	1644.79	0.31	0.83	209	328.42	0.18	1.00	525
170	10	15	7	1647.90	0.31	0.83	208	329.65	0.17	1.00	523
171	10	15	12	1852.58	0.28	0.74	185	427.13	0.13	1.00	404
172	10	15	12	1838.21	0.28	0.74	187	427.22	0.13	1.00	403
173	10	15	12	1852.99	0.28	0.74	184	428.39	0.13	1.00	402
174	10	15	12	1840.43	0.28	0.74	186	427.34	0.13	1.00	404
175	10	15	12	1853.76	0.28	0.75	184	428.85	0.13	1.00	402
176	10	15	12	1849.35	0.28	0.75	184	427.96	0.13	1.00	402
177	10	15	12	1842.29	0.28	0.74	186	429.67	0.13	1.00	401
178	10	15	12	1839.99	0.28	0.74	187	429.50	0.13	1.00	402
179	10	15	12	1843.61	0.28	0.75	185	428.21	0.13	1.00	403
180	10	15	12	1830.21	0.28	0.74	187	428.82	0.13	1.00	402
181	20	2	2	1465.07	0.21	0.58	96	255.34	0.04	0.16	107
182	20	2	2	1594.12	0.20	0.53	88	264.84	0.04	0.16	102
183	20	2	2	1611.87	0.18	0.49	85	259.80	0.03	0.15	97
184	20	2	2	1537.35	0.20	0.53	89	256.51	0.03	0.13	87
185	20	2	2	1608.50	0.23	0.62	106	258.70	0.03	0.15	97
186	20	2	2	1399.66	0.18	0.48	83	259.99	0.04	0.17	109
187	20	2	2	1561.64	0.21	0.56	98	258.29	0.03	0.15	97
188	20	2	2	1671.76	0.21	0.55	96	254.86	0.04	0.15	102
189	20	2	2	1473.69	0.18	0.48	79	267.57	0.02	0.11	68
190	20	2	2	1521.03	0.20	0.54	92	261.55	0.03	0.15	99
191	20	2	7	1785.15	0.21	0.55	95	363.45	0.03	0.17	82
192	20	2	7	1792.19	0.20	0.54	92	355.08	0.03	0.21	101
193	20	2	7	1651.07	0.20	0.52	87	357.65	0.03	0.20	94
194	20	2	7	1706.74	0.18	0.48	81	359.82	0.03	0.19	91
195	20	2	7	1914.79	0.22	0.58	97	360.60	0.03	0.16	77
196	20	2	7	1908.63	0.24	0.62	105	359.11	0.03	0.18	85
197	20	2	7	1785.39	0.19	0.50	87	357.19	0.04	0.23	109

198	20	2	7	1838.19	0.19	0.53	90	358.65	0.04	0.22	106
199	20	2	7	1845.28	0.22	0.59	99	362.02	0.04	0.22	103
200	20	2	7	1953.21	0.24	0.64	106	362.46	0.03	0.19	89
201	20	2	12	2084.47	0.23	0.62	105	456.92	0.03	0.21	81
202	20	2	12	1916.38	0.20	0.53	90	463.04	0.04	0.30	111
203	20	2	12	1964.55	0.21	0.54	88	460.64	0.03	0.24	90
204	20	2	12	2074.01	0.21	0.55	90	459.57	0.04	0.28	103
205	20	2	12	2299.04	0.26	0.71	120	458.60	0.03	0.27	101
206	20	2	12	2084.22	0.22	0.59	101	462.07	0.03	0.27	100
207	20	2	12	2080.97	0.21	0.54	90	457.31	0.03	0.25	93
208	20	2	12	1973.53	0.23	0.63	106	458.36	0.04	0.33	124
209	20	2	12	1881.45	0.20	0.53	91	456.11	0.03	0.26	95
210	20	2	12	2014.51	0.21	0.56	97	458.56	0.03	0.26	96
211	20	7	2	2099.00	0.36	0.96	163	262.48	0.11	0.50	332
212	20	7	2	2070.25	0.36	0.96	165	260.63	0.11	0.48	320
213	20	7	2	2071.00	0.35	0.96	165	260.97	0.11	0.51	335
214	20	7	2	2060.78	0.36	0.96	165	261.79	0.10	0.46	305
215	20	7	2	2069.99	0.36	0.97	165	259.28	0.12	0.51	341
216	20	7	2	2035.11	0.35	0.96	169	260.20	0.12	0.54	356
217	20	7	2	2104.30	0.37	0.97	162	260.34	0.12	0.54	360
218	20	7	2	2069.53	0.36	0.97	164	260.59	0.13	0.55	367
219	20	7	2	2097.03	0.36	0.96	163	262.42	0.12	0.55	362
220	20	7	2	2037.88	0.36	0.96	168	260.90	0.11	0.47	313
221	20	7	7	2257.63	0.33	0.88	152	362.76	0.12	0.76	362
222	20	7	7	2264.51	0.33	0.88	151	359.93	0.11	0.68	328
223	20	7	7	2272.24	0.32	0.88	150	360.92	0.12	0.70	337
224	20	7	7	2268.83	0.33	0.88	149	360.80	0.11	0.67	320
225	20	7	7	2281.70	0.33	0.88	150	361.75	0.11	0.68	325
226	20	7	7	2274.31	0.34	0.88	149	357.69	0.11	0.68	329
227	20	7	7	2268.67	0.32	0.88	151	359.86	0.12	0.70	338
228	20	7	7	2273.26	0.33	0.88	150	360.82	0.11	0.70	333
229	20	7	7	2270.91	0.33	0.88	150	359.88	0.11	0.70	335
230	20	7	7	2231.70	0.32	0.88	153	361.28	0.13	0.77	370
231	20	7	12	2494.49	0.31	0.81	136	461.17	0.11	0.89	333
232	20	7	12	2460.31	0.30	0.81	139	459.78	0.11	0.88	330
233	20	7	12	2513.61	0.30	0.81	134	460.63	0.11	0.84	314
234	20	7	12	2457.29	0.30	0.81	138	460.84	0.12	0.92	343
235	20	7	12	2421.72	0.30	0.81	140	460.64	0.12	0.96	358
236	20	7	12	2459.14	0.30	0.81	139	460.83	0.11	0.85	317
237	20	7	12	2488.58	0.30	0.81	137	461.68	0.11	0.88	331
238	20	7	12	2472.59	0.30	0.81	139	459.42	0.11	0.86	323
239	20	7	12	2442.17	0.30	0.80	140	459.86	0.11	0.86	325
240	20	7	12	2493.37	0.30	0.81	136	459.50	0.11	0.89	335
241	20	15	2	2069.28	0.36	0.96	165	260.50	0.22	0.99	656
242	20	15	2	2071.47	0.36	0.96	165	259.75	0.23	1.00	664

243	20	15	2	2043.75	0.36	0.97	167	262.76	0.22	0.99	651
244	20	15	2	2080.76	0.35	0.96	164	262.09	0.23	1.00	658
245	20	15	2	2090.76	0.37	0.96	164	261.83	0.23	1.00	659
246	20	15	2	2104.73	0.37	0.97	161	258.97	0.23	1.00	666
247	20	15	2	2077.06	0.35	0.97	164	261.77	0.22	1.00	658
248	20	15	2	2124.72	0.36	0.96	161	260.45	0.22	0.99	656
249	20	15	2	2071.28	0.37	0.96	165	261.26	0.22	1.00	658
250	20	15	2	2081.07	0.36	0.97	164	260.91	0.23	1.00	658
251	20	15	7	2253.92	0.33	0.88	150	357.73	0.17	1.00	482
252	20	15	7	2325.07	0.34	0.88	147	362.27	0.16	1.00	476
253	20	15	7	2295.79	0.34	0.89	148	362.48	0.16	1.00	475
254	20	15	7	2269.08	0.33	0.88	150	360.62	0.16	1.00	478
255	20	15	7	2257.84	0.33	0.88	152	363.10	0.16	1.00	474
256	20	15	7	2282.67	0.33	0.89	149	361.69	0.16	1.00	477
257	20	15	7	2276.40	0.33	0.88	150	364.45	0.16	1.00	473
258	20	15	7	2304.28	0.33	0.88	148	361.41	0.16	1.00	477
259	20	15	7	2258.84	0.33	0.88	151	361.80	0.16	1.00	477
260	20	15	7	2247.01	0.33	0.88	152	361.04	0.16	1.00	478
261	20	15	12	2484.35	0.30	0.81	138	461.94	0.13	1.00	373
262	20	15	12	2472.63	0.31	0.81	137	461.33	0.13	1.00	373
263	20	15	12	2498.21	0.31	0.82	136	459.49	0.13	1.00	375
264	20	15	12	2443.10	0.30	0.80	140	461.98	0.13	1.00	373
265	20	15	12	2485.40	0.31	0.81	137	460.53	0.13	1.00	374
266	20	15	12	2469.88	0.30	0.81	138	461.45	0.13	1.00	374
267	20	15	12	2480.44	0.31	0.81	138	461.41	0.13	1.00	373
268	20	15	12	2515.35	0.30	0.81	136	462.77	0.13	1.00	372
269	20	15	12	2510.59	0.31	0.81	135	460.42	0.13	1.00	375
270	20	15	12	2459.07	0.30	0.81	139	460.69	0.13	1.00	374

4. Example of the table created in Excel file with all resources travels data exported

Order	Resource	Position_I	Position_F	Time_I	Time_F	Duration
1	Vehicle	0	0	14.83497454	14.83497454	0
1	Lift	1	1	14.83497454	14.83497454	0
1	Lift	1	1	14.83497454	14.83497454	0
1	Vehicle	0	1	14.83497454	18.05987764	3.224903099
1	Vehicle	1	15	43.05987764	56.15987764	13.1
1	Vehicle	15	0	81.15987764	94.90987764	13.75
1	Lift	1	1	94.90987764	94.90987764	0
1	Lift	1	2	94.90987764	101.259084	6.349206349
1	Vehicle	0	4	101.259084	107.7088902	6.449806199

5. Examples of the table created in Excel file with all vehicle complete processes data (from when the it is requested until it is released)

Order	Resource	Time_I	Time_F	Duration
1	Vehicle	14.83497454	259.6344993	244.7995247
2	Vehicle	259.6344993	484.2215226	224.5870233
3	Vehicle	484.2215226	730.4024683	246.1809457
4	Vehicle	730.4024683	946.7171149	216.3146467
5	Vehicle	946.7171149	1202.267221	255.5501061
6	Vehicle	1202.267221	1459.549903	257.2826821
7	Vehicle	1459.549903	1702.246678	242.6967752
8	Vehicle	1702.246678	1944.98992	242.7432414
9	Vehicle	1944.98992	2194.622405	249.6324849

6. **Example of the table created with order information (arrival, processes and SKUs ordered)**

Order	Arrival	Starts	Finishes	Duration	Duration V	SKU 0	SKU 1	SKU 2	SKU 3	SKU 4
1	14.83497454	14.83497454	259.6344993	244.7995247	244.7995247	3	5	2	5	5
2	31.48928275	259.6344993	484.2215226	452.7322399	224.5870233	6	3	2	4	5
3	32.0749799	484.2215226	730.4024683	698.3274884	246.1809457	2	3	2	9	4
4	53.10206873	730.4024683	946.7171149	893.6150462	216.3146467	5	7	1	3	4
5	75.54006447	946.7171149	1202.267221	1126.727157	255.5501061	4	4	4	4	4
6	124.3810102	1202.267221	1459.549903	1335.168893	257.2826821	1	4	5	8	2
7	216.872449	1459.549903	1702.246678	1485.374229	242.6967752	8	5	3	1	3
8	279.6168505	1702.246678	1944.98992	1665.373069	242.7432414	1	7	3	3	6
9	384.1632888	1944.98992	2194.622405	1810.459116	249.6324849	5	3	5	3	4
10	538.9419792	2194.622405	2426.288252	1887.346273	231.6658473	4	3	4	4	5

7. **Example of the table created with a summary of all replications results**

	V Cycle	Total Cycle	Waiting Vehicle	Utilization Lift	Utilization Vehicle	Orders
Retrieval 1	245.673	69655.913	69410.239	0.218	0.999	702
Retrieval 2	245.390	67612.487	67367.097	0.221	1.000	703
Retrieval 3	245.921	69515.451	69269.530	0.217	0.999	701
Retrieval 4	244.945	66772.951	66528.007	0.222	0.999	705
Retrieval 5	245.606	66292.677	66047.071	0.218	0.999	703
Retrieval 6	245.770	67689.855	67444.085	0.217	0.999	702
Retrieval 7	244.331	66461.834	66217.503	0.220	0.999	706
Retrieval 8	246.651	69736.508	69489.857	0.217	0.999	699
Retrieval 9	245.260	70192.202	69946.943	0.218	1.000	704
Retrieval 10	246.687	70748.869	70502.182	0.217	0.999	699
Average	245.623	68467.875	68222.251	0.219	0.999	702