ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

Corso di laurea in ingegneria informatica magistrale

Mission scheduler for a rail guided vehicle system

Relatore:
Chiar.mo prof.
Silvano Martello

Presentata da:
Roberto Giuliani

Correlatore:
Prof. Michele Monaci
Ing. Claudio Gambetti

Sessione II
Anno Accademico 2018/2019

To my family

Abstract

A transport system with AGVs (automatic guided vehicles) is a fully automatic system that provides logistics services in industrial environments such as warehouses and production plants. These systems have reached such a degree of maturity as to allow, in their daily use, the application of heuristic algorithms for the optimisation of the various operations they perform. For instance find the shortest paths between working stations and storage area, assign movements and strategic positions for idle vehicles, operate efficient and long-life battery management and more.
A relevant interesting algorithm, presented and developed in this thesis, concerns the sorting of products in the shipping phase, which affects the scheduling tasks assigned to the autonomous vehicles. The scheduler has the aims of determining which operations have more strict constraints and more priority over others.
Studies and practice have shown that the adoption of a valid scheduler implies considerable improvements in the system performance, consequently it is advisable to dedicate time and effort to the research for the right one.
The following algorithms obtained a successful outcome and they have been implemented for the production of a modern automated warehouse located in the city of Cesena, Italy. The paper is divided into four chapters, with a further one dedicated to conclusions.

Introduction

Nowadays logistics, concerning the industrial business, constitute a fundamental component of the company operating system, together with production, marketing and finance.
In recent years, the common awareness with regard to logistical issues has grown significantly and the manufacturers are more and more aware of the need to improve certain aspects, minimising the stock-taking and maximising the products rotation. In this context, we find issues related to the handling flow, which, in order to keep pace with the needs of the market, requires state-of-the-art structures and machines.
Automation meets these demands by offering products and techniques that can increase corporate value, reduce costs and increase revenues. An example of this is found in AGVs systems, capable of handling hundreds of products per hour in relation to the plant layout and the number of vehicles, reducing maintenance costs and making the warehouse continuously operational.
This innovative internal transport system has been experiencing considerable interest in recent years thanks to its flexibility and efficiency.
The first part of this thesis is aimed to clarify the present literature concerning logistics and automated warehouses along with their respective operations, such as acceptance controls, storage, order preparation and shipment. After providing a theoretical overview, the design and implementation of an autonomous guided vehicles system is examined, considering its components and formulating algorithms for their efficient control and administration.
The paper then concludes with the formulation of a scheduler, as a support to decision-making process in the optimal management of limited resources: the vehicles; their usage and assignment of certain activities. The approach adopted is based on breaking down the problem into sub-problems: the first part involves the implementation of an algorithm to define the priority of the activities in relation to defined criteria coming from external systems; the

second keeps track of the current state of the system through a scheduling matrix and decides which operation to perform when one or more resources become idle; and, in conclusion, the third part has the task of assigning activities to the vehicle fleet.
The heuristics of this thesis constitute an attempt to propose feasible solutions that have good quality, even if not optimal, calculated on real applications and data; and through models that avoid certain simplifications occurring in theoretical studies and academic researches.
The final part of the paper concludes with the model results as a whole and as each individual algorithms, in different instances of the problem, from situations of low product movement to peaks of high movement. Which are then analysed and discussed.
The project was developed with the support of the *automation business unit* from Onit group and was able to test its application in a real business scenario thanks to the construction of the new Orogel plant based in Cesena. Orogel is an agricultural cooperative company, leader in the production of frozen foods, oriented towards innovation and in continuous development.
The year 2020 will see the launch of a new automated refrigeration plant, the largest product storage warehouse in Europe. The warehouse is managed in a completely automatic way, it has a surface area of 7,800 square metres, approximately 230 thousand cubic metres of volume, and it is made to store 35 thousand tons of frozen product at an operating temperature of minus 25 degrees.

Contents

Chapter 1

Logistics

The need to orient company production policies to the market has made the traditional logistics and production-planning model inadequate and obsolete.
Today the role of logistics is to bring production closer to sales, creating the assumption for a perfect harmony between the two operations, to ensure delivery of the products on time and in the quantities requested and minimising stock, transport and information management costs, regarding production flows.
Companies work to develop logistics projects not only with the aim of reducing costs, but to increase revenues and, in general, their business: it consequently becomes a competitive point and can represent a strength in the market.

1.1 Logistic

Adopting a widely shared notion, business logistics is defined as the process of planning, implementing and controlling the flow and storage of materials and, additionally, of the associated information, from the point of origin to the point of consumption, in order to respond to customer requests in the most efficient and effective way for the company.
More precisely, the logistic process aims to harmonise flow - both of physical products and information - through the various functions and company processes, in order to create greater value for the company and for the customer. Logistics cannot overlook the dynamics of market demand which, at different levels, guide and dictate times and procedures of company operations.
Following this line of reasoning, the planning is distinguished according to the initiators of the business logistic process, in which the strategy defined as "pull" is the driving force, where logistic processes are activated by impulse of the request, in response to a customer order.
Although this type of planning is common in engineered production processes, more and more companies are adopting these productive and logistic aspects. The result is a customer-oriented and guided business logistics, able to respond reactively to the volatility of demand, classic of current markets. The environment of accelerated rhythms and greater scale, in which logistics operations must act, have led, and are still leading, to a change in the business logistic planning model. To maximise the value in a logistics system, there are a large number of planning decisions to consider: from the choice of the storage system, the elements to be taken in succession to respond to the orders received from customers and so on, until high-level business decisions to invest in the construction of a new plant.
There is much literature and many software support tools that focus on every single area of logistics systems. From this premise, we can understand the importance of the supply chain in today's world and how companies are interested in possible improvements in any of its many aspects.

1.2 Logistics integration and automation

The natural evolution of the logistic system leads to the development of integrated logistics, with the aim of seeking the best coordination between the

area that are part of it and the information system that guide its progress. Over time, we have witnessed an ever-stronger integration of logistics flows, via automation applied to both physical aspects and information aspects. In fact, robotics and automation are now key success factors and are revolutionising the world of industry. Applied to autonomous robots and automatic storage systems, to detection and tracking technologies or even advanced software for the supply chain, they represent a turning point that allows an increasingly rapid, safe and error-free distribution, shorter time to market, with estimated efficiency increases of 20-30% and, ultimately, lower costs for businesses and families. It should be noted that 10-15% of the cost of the product is influenced by the logistic aspect, therefore reducing these values ensure an increased strength in the market.
Today, logistics automation projects have reached an average annual increase of 7% of total turnover and a long growth outlook sees this figure being close to or above 10%. [Source ANIE Federation]
The integration of all operational activities is the main tool to achieve the objectives set, as it reduces inefficiencies and downtime. Innovation and business acceleration look to a rapidly changing industry. Notwithstanding the importance of automation in the modern industrial vision, it is necessary to maintain a critical approach: even for automatic devices, the guiding spirit must be the technical-economic justification of each solution, thus avoiding extreme implementations, driven by an unreasonable desire for automation.

1.3 Supply Chain Management

The term supply chain management (SCM) was introduced in 1982 by two consultants, Oliver and Weber, to describe the connections between logistics and other business functions. Although the principles remain the same, the definition undergoes various transformations in relation to the area of interest. Among the various possibilities, a new interpretation is preferred in view of integration between different processes and entities.
Supply chain management is a complex system that optimises the coordination between the processes of suppliers, producers and distributors - in other words the totality of the organisations that are involved - through upstream and downstream connections, in the various developments that generate value for the end customer, in terms of product or service.
SCM also includes the logistic system that starts from the raw materials,

transforms them into products and ends with their distribution to the final consumers. Below is a possible outline given by the logistic chain of a generic company, where the raw materials are delivered by the suppliers to the manufacturing plants, which then produce components or semi-finished parts. The finished products are then assembled in different plants, called assembly plants. Distribution is also subdivided into several steps: two central distribution centres (CDC) supply multiple regional distribution centres (RDC).
The chain is created according to the type of product and request, removing or adding plants and connections between the various phases. Moreover, the latter can be more or less complex, composed by a single and simple line of transport or by a more articulated network that involves different systems and companies, such as automotive and rail.

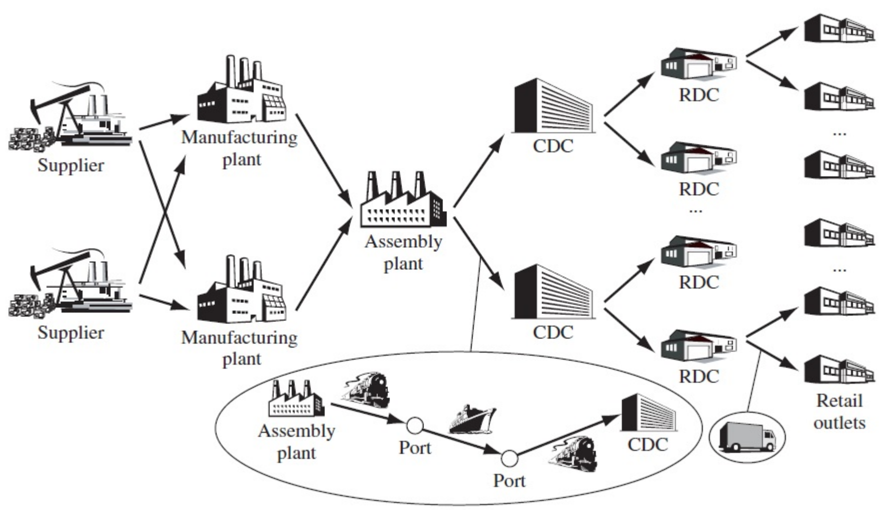

Figure 1.1: A supply chain example

The two main flows have opposite directions: the physical one, from the supplier to the customer and the informative one in the opposite direction. The products follow the sequence of the supply chain, except for those that are obsolete, damaged or do not work, while, on the contrary, information follows the reverse path, from consumers to suppliers.
In the pull system, sellers take charge of customer requests and transmit them to the producer who reformulates them as orders for suppliers. Similarly, in pull type systems, historic sales are used to make future economic

forecasts on product demand and the related demand for materials. Almost all the authors of the literature, dealing with supply chain management in terms of process integration, agree that it:

1. allows for the opening up of a consideration of process integration, from an intra-organizational perspective to an inter-organizational one;

2. potentially involves a very large number of autonomous organisations. This link is inter-company in nature and can take place both in terms of information sharing and in terms of coordinating operational processes;

3. the purpose is to supply a product/service of high quality for the consumer with an appropriate use of resources and to build competitive advantages for all the players in the chain. In keeping with such an approach, the concept of competition passes from being that of business against business, to one that searches for forms of collaborative competition that increase the value for all the sides involved in the chain.

Although some authors use the term SCM as a synonym for integrated logistics, there are significant differences between the two approaches. Logistics, unlike SCM, deals with the operations related to the movement and the flow about goods and information. It therefore represents a function of vital importance for supporting SCM, as it is directly responsible for integrating the needs of the market with the production process.

1.4 Industrial warehouses

The warehouse must be perfectly integrated in the logistic cycle of the company. Latterly the evolution of the market and the high labour cost have extremely complicated the role of warehouses and many companies, in order to contain costs, have shifted production to emerging countries, making production scheduling - and therefore the correct determination of warehouse storage capacity - particularly complex.
The warehouse represents a key point for the supply chain, as it supplies the main logistic service: making the right product available, in the correct location, at the right time and at minimum cost.
The distribution system consists of two distinct structures:

1. the logistics channel represents the distribution network for the grouping, selection, sorting and transport of goods. Of these functions, the first three are carried out within the warehouse;

2. the commercial channel is made up of the structures suitable for selling the product.

The warehouse is important both from a temporal point of view, as it enables economies linked to supplies and to optimise production, and from a commercial perspective, making it possible to consolidate loads, with a consequent reduction in transport costs. Warehouses are mainly divided according to the position in which they are located: production warehouses handle large quantities of materials and are subdivided into warehouses of raw materials, inter-operational, finished products and combinations of them; distribution warehouses, on the other hand, contain a wide variety of goods and are subdivided into central or peripheral, depending on the size of their catchment area.

Central warehouses allow for the storage of large quantities of materials, with the aim of maximising efficiency of reception, storage, extraction and shipping. Regional warehouses are centres located near the point of final distribution. They contain large quantities of goods received and stored; on the contrary, shipments are reduced and heterogeneous.

The nature of a warehouse deeply influences the processes that manage it. In this regard, to increase efficiency, the management of flows is totally delegated to automated systems. The scheduling of storage and retrieval missions is usually managed by electronic computers, equipped with dedicated software. Minimising routes, maximising volumes within the storage locations, and minimising internal movements are just some of the logistical problems that must be faced in order to boast an optimised and flexible system. Many of these problems can be traced back to classic examples of Operational Research, such as the Vehicle Routing Problem for transportation or the Knapsack for volume management, although structural or organisational constraints often make the study of the optimal solution very complicated. In these cases, a heuristic approach enables the achievement of the right compromise between calculation times and quality of the solution.

1.5 The operational cycle of warehouses

The operational cycle of a warehouse can be subdivided into six phases, of which the first, third and sixth are of primary importance:

1. arrival or reception;
2. acceptance testing, performed in the raw material warehouses after the arrival of the goods;
3. storage;
4. preparation, possibly carried out after storage to better adapt the materials to production or transport requirements;
5. awaiting shipment, necessary in some finished product warehouses;
6. shipping.

The arrival area must be sized in anticipation of the acceptance of potentially notable quantitative peaks of goods, due to a discontinuous entry flow; this occurs especially in the case of goods coming from external suppliers, in the presence of a strongly seasonal trend, or when materials coming from internal departments or work-centres with non-regular production must be brought-in to the warehouse.
The storage phase includes the period in which the product is released into the warehouse whilst waiting to be sold. Its area represents the largest area as it is prepared for storing unsold goods. It is characterised by a structure specially equipped for proper storage and preservation.
The company is not willing to maintain a high number of excess stocks, because it would mean an immobilisation of capital in unused resources. Raw materials, semi-finished products and finished products should remain immobilised in the warehouse as short a time as possible pending their use in the transformation or sales processes. It is therefore necessary to plan the requirements of materials and sub-components of the finished products so that they are always where they are needed, when necessary and in the quantities requested.
Shipment is the last stage of the cycle and includes the exit of the product from the warehouse and its subsequent transport. It represents the smallest

area of the warehouse due to the effect of order planning within the company, which has greater room for manoeuvre in relation to the needs of the area in question.
In the cost of the warehouse, receipt, shipment and storage account for 50%, in equal parts.
Today in large complexes, shipping is completely automated, robotic vehicles move on their own on physical or magnetic tracks, pick up products, load them and position themselves for exit according to how the customer's orders are scheduled. The correct performance of tasks is facilitated by the use of information support and the application of IT procedures for inventory management.

1.6 Internal movement

Internal movement is subdivided into the picking phase, the storage phase and the displacement between internal positions within the warehouse. During the picking phase it is required to know the real location of the item to carry out the picking operation, while, in the stocking phase, the product position is initially logically assigned and later the actual allocation is carried out. Generally, the goods entering the warehouse are more homogeneous than the outgoing ones. Indeed, orders from end customers often involve a set of products that are more heterogeneous than those coming from suppliers or from production chains.
Picking is more costly than storing, in terms of locations to be reached and therefore of paths to be taken since it requires access to different classes of products.
Each sales and shipping order results in the following set of operations:

1. the formation of batches, or the set of withdrawal missions;
2. the scheduling of the machines;
3. picking;
4. loading onto vehicles.

In the first point the orders are formed by lots, called batches. Optimal batch size is estimated by balancing the picking times with the subsequent sorting. The computation can be done using more or less complex algorithms,

depending on the system requirements and the specific customer's requests.
The second stage is carried out by electronic calculators which, through models and algorithms, have the task of identifying the best solution to promptly release the products with higher priority, minimise the displacements and guarantee an admissible batch ordering within the same shipment.
The third stage involves the physical picking of the products in relation to each batch. While the previous phase has the requirement of ensuring feasibility in the organisation of the batches, this phase has the true task of sorting the items in order to allow a simple and correct picking process once the final customer is reached.
Finally, the loading onto the vehicles takes place by packing the products in pallets or containers, ordered according to the last come first served policy and aggregating consecutive orders. The intention, to minimise costs, is to reduce the number of bins needed to contain the objects: this translates into problems of a geometric nature, with more or less constraints, depending on whether the objects are stackable, and with various degrees of complexity, based on the number of dimensions to be taken into account.
The third point, overall, is considered the most expensive, requiring about 70% of the total time. The picking phase is dealt with greater depth in this thesis, thanks to the introduction of interesting and innovative flexible transport systems that use autonomous guided trolleys, called *automated guided vehicles AGV*.
The AGV machine, equipped with a calculator, moves along extremely complex and flexible paths defined by the guide system, automatically exchanging materials, pallets or other items, with the machining centres. The master system provides real-time detection of fleet positions, management of transport requests, according to certain priority rules, and scheduling the mission orders for individual trolley.
The introduction of these devices increases efficiency, reduces operating costs and optimises picking and stocking operations. In this regard, the following chapter deals specifically with the design and control of AGV vehicles, introducing algorithms and models for their correct implementation.

1.7 Logistic performance measures

Logistics, warehouse management, internal handling, order management and customer service represent processes that, if optimised and coordinated with the actions of other agents, establish or at least influence some performances that are critical for the success of the company.

In recent years, much of the management studies has focused on identifying a set of metrics that can be used in logistics. These indicators integrate economic-financial measures and technical-operational measures. Some are general in nature and can be found in all processes, such as performance in terms of cost, time, quality and value; others are typical of specific processes, such as productivity, versatility and flexibility in the logistics chain.

Key Performance Indicators KPIs, are a number of qualitative and quantitative indicators that assess the business results achieved, with reference to fundamental aspects of business management such as the level of service guaranteed to the customer, the level of stock in the warehouse and the level of costs for transport. The better the choice and measurement of the KPI is, the better the control on the progress and the regulation of the objectives can be. The main indicators aim to evaluate the efficiency of storage, studying the surface saturation of the warehouse, and the materials handling, measuring speed and accuracy of operations.

The main factor to be analysed within the warehouse area concerns handling, i.e. the operation by which a material is stored in a location or picked from the same, considering time and position domains. The related KPIs are called lead-time and potential handling.

The lead-time is measured in terms of response. The lower the wait, the greater the competitiveness. The lead-time or reaction time has a particular importance in the logistics sector. Focal point of outgoing logistics, the lead-time makes response times flexible between an order validation, its shipping and delivery. The competitiveness of a company lies not only in its ability to satisfy customer needs, but also in being able to reduce the time between the product purchase and its delivery.

The formula is very simple, it represents the time difference between the end and the beginning of the operation. Inside the warehouses, it is mainly used to measure the timing between the product picking request and the conclusion of the operation. This indicator can have very high values in relation

to the number of operations and movements that the warehouse is currently managing. As far as possible it should be weighted with regard to the daily flow.
The potential handling is the dynamic capacity of the warehouse, expressed in quantity, as the maximum number of incoming, outgoing and in-transit units. This parameter takes into consideration the quantities and the vehicles used. The actual potential handling is given by the product of the potential of each individual vehicle, whether automatic or operator-guided, and the number of means of transport.

$$PH_{\text{vehicle}} = UF \cdot \frac{3600}{T_{\text{op}}} \qquad [operation/hour] \tag{1.1}$$

$$PH_{\text{tot}} = PH_{\text{vehicle}} \cdot N_{\text{vehicle}} \qquad [operation/hour] \tag{1.2}$$

Where:
PH_{vehicle} is the potential handling index for each vehicle while PH_{tot} is the total potential handling, UF is the utilization factor for each vehicle and in conclusion T_{op} is the time spent for usual operations.
In an automated warehouse, once designed and built, both the incoming potential and the handling potential are fixed and defined.
A further point is the financial aspect, some financial indicators are essential to control the operational costs related to the warehouse and appreciate the speed of circulation, including the inventory turnover. The inventory turnover index is the parameter used to assess the speed of regeneration of the goods, indicating the number of times a complete renewal of stock takes place in a given period of time. It is given by the formula:

$$IT = \frac{\sum_{i=1}^{t} U_{\text{i}}}{G_{\text{m}}} \qquad [product\ sold/time] \tag{1.3}$$

Where:
IT is the inventory turnover index, U_{i} is the total consumption in the period and G_{m} is the average stock in the period. Assuming that, during a defined period, 1,000 pieces were consumed against an average stock of 250, the inventory turnover index will be equal to 4, which means that the 250 units have left, for sale or use, and then reintegrated 4 times. The index shows that from the logical viewpoint the warehouse has been regenerated 4 times during the year, or that the materials have remained in storage on average 91 days (365/4 days).

In general, a high turnover rate translates into less capital invested in inventories and lower inventory management costs, while a low rotation index leads to the characteristic drawbacks of over-large stocks such market discounts and internal inventory obsolescence. It is intuitive that the rotation index should tend to be high.
The above listed are just some of the most common indicators used in logistics. From the warehouse point of view, they become fundamental factors to put right the flows that can slow down the entire logistic process. KPIs have the aim of making a significant contribution to improving the process itself in terms of value creation, in line with the implementation of short, medium- and long-term corporate strategies.

Chapter 2

RGV system: design and control

The most flexible and innovative automatic transport system inside the warehouse is the one that uses automated guided vehicles AGV along a complex set of routes. The exchange of material with working stations and assembly centre takes place automatically and under the control of the central calculator. The plant is completely monitored by the supervision system, which administers in real time every operational situations, manages the fleet and automatically process data relating to production statistics. In this way, it is possible to reduce human involvement to minimal responsibilities and optimise production flows, creating a safe and structured working environment in a simple and linear way.
Indicatively in 2012, Amazon, the largest internet company in the world, purchased 15,000 AGVs for its warehouses, with the aim of reducing lead times and increasing the level of customer satisfaction.
Currently the use of AGV vehicles mainly includes the automotive, electronics and food sectors, but in the coming years an interesting increase for small/medium companies is also predicted.

2.1 AGV vehicles

Barrett Electronics built the first AGV vehicle in 1950. In the 60s and 70s the on-board controllers were first transistorised then replaced by integrated circuits, thanks to this, in the following decades there was a strong increase in deployments.
Thanks to their flexibility and efficiency, nowadays AGV systems are used not only in classic production lines, but also in automated warehouses where high quantities of goods are constantly in motion. Companies that produce consumer goods, such as in the food and beverage field, have invested and are still investing in automated logistics.
The possibility of covering 24 hours of work using a single AGV greatly increases the market in countries where labour cost is high and where the return on investment is rapid. Indeed, the purchase of an AGV system implies a significant initial capital cost, but nevertheless it guarantees high economic potential due to low maintenance costs, compared with conventional vehicles, and to the possibility to operate without interruptions with reduced human assistance.
In Europe, the countries that are most increasing the development of AGV systems are Italy, Germany, the United Kingdom and the Scandinavian countries.
The online orders received by customer are managed through the company system that prepares the regular flow of operations, including the internal movement of the goods and the subsequent shipment, to achieve the sale of the product. It assigns the designated products to the warehouse locations and defines the route to reach them. The AGV vehicles will follow this route so that the order is completed correctly and with the priorities needed.
The movement is defined by algorithms that are based on objective criteria such as minimising travel time or waiting time caused by traffic congestion. Although the layout is static, choosing the best possible route is a dynamic decision and depends on current conditions and system forecasts.
The central calculator, in this context, has the task of monitoring the position of the vehicles, controlling flows and guaranteeing productive maintenance of the battery. Inappropriate direction policies can lead to congestion problems, deadlocks and collisions.
The sequence of products and orders, on the other hand, depends on complex indicators such as priority, product type, quantity of goods involved, waiting time, etc. The need to carry out multiple orders simultaneously, due

to simultaneous sales flows, involves a high level of organisation and management. This task is performed by the warehouse supervision system thanks to heuristic models and algorithms for the correct scheduling of the missions assigned to the AGV vehicles.
An AGV system must deal with variable and dynamic executive conditions. The operating flow entering the system is typically irregular and unpredictable, certain areas of the warehouse can be temporarily closed and the AGVs themselves can enter and exit the system because of maintenance. Although these work situations may arise, the system must operate in a robust, efficient and safe mode.
In conclusion, the realisation of a material handling system using AGV vehicles offers advantages such as: operational flexibility, reduction of the overall handling time, better control of the process and the materials being worked on and balancing of production with better quality.
During the planning stage it is necessary to guarantee a correct labour distribution among the various systems that allow the overall management of the automatic warehouse.
The typical topology in automated warehouses with AGV vehicles consists of:

1. Warehouse management system WMS: it has the task of supporting the operational management of the physical flows passing through the warehouse, from the control of incoming goods during the acceptance phase to the preparation of shipments to customers;

2. Warehouse control system WCS, also called supervision system: it is a software solution with integrated control and real-time monitoring, that manages the flows of goods during their movements on different types of automatic equipment. It is able to exchange ongoing communications, process commands and signals and optimise the flow of materials via strategic decisions;

3. Master PLC or central calculator: ensures the real-time management of a set of devices and acts as an intermediary between the supervision system, typically located in servers away from the warehouse, and the on-site machines;

4. PLC inside every single device.

A proper implementation and coordination between the presented systems enables the pre-established objectives to be met, and the advantages that AGV vehicle technology makes available to be fully exploited.
Efficient and effective solutions should be sought by adopting heuristic approaches in order to obtain the right compromise between calculation time and quality.

2.2 RGV - Rail Guided Vehicle system

The routing systems are divided between those with a fixed layout and those with a variable layout. The first require the installation of a track on the floor - for example a rail or magnetic belt - or underneath the surface; the second require a map of the area in which the vehicles operate and a series of fixed reference points that can be tagged. Certainly, the second category allows greater flexibility, but requires superior computation capacity on the part of the central calculator; it can be said that with the routing systems of the first type the vehicles follow a path, while with the systems of the second type they choose a path.
In AGV systems, the most common vehicles are those with a mechanical guide: the electrically-driven transport vehicles with on-board calculator move along specific paths, guaranteeing high degree of precision both when driving and approaching the stations.
The RGV system is a special transport system on an electrified monorail, very common in all industrial sectors where logistic efficiency is a key factor; in many production processes and in the connection between different areas of the warehouse, it is in fact particularly suitable and convenient for routes that cover long distances.
It is a relatively simple transport system, where, in the traditional version, the circuit is composed of an electrified aluminium belt with graphite and copper brushes that run over it. These brushes guarantee the power required for the power supply and the independent movement of each machine.
The rail is hung from the ceiling to reduce clutter on the floor, and is configured in relation to the space and the characteristics of the building, which consists of straight lines, curves, turnouts and parallel lines. The result is a versatile and flexible transport system that allows future modifications and expansions. The circuit, if necessary, can be redesigned and new vehicles can be inserted at any time. Typically, the number of vehicles is chosen in

Figure 2.1: RGV vehicle

relation to the quantity of pallets per hour that they wish to move. A correct decision requires an in-depth analysis and the use of specific indicators such as the lead time and the potential handling presented in the previous chapter.
The RGV system ensures an increase in productivity and a reduction in the space required. Transportation is guaranteed at high speed and performance levels. The maximum speed of an empty vehicles is 2000 mm/sec, which is reduced in relation to the load they are delivering or on particular sections of the route. This system is considerably faster than the most common conveyor systems, where the speed is around 500 mm/sec. Furthermore, the space and time required to move the pallets between the different stations is minimal.
The vehicles, which include chains or rollers, can move products of different sizes, formats and weights. Usually intended for the transport of pallets, they are suitable for types such as euro-pallets (1200 x 800 mm) and american pallets (1220mm x 1020mm). They are able to transport goods with a maximum weight of two tons, whether individual or composite pallets.
The RGV system has advantages and disadvantages compared to other types of AGV vehicles; the choice is a decision that is up to the system designer.
It guarantees a very high level of safety. The operations are completed with completely automatic procedures and under the supervision of the central system. The products are handled in a silent, clean and efficient manner.

2.3 Guide-path design: single loop type

The efficiency of an RGV system is strongly influenced by the path designation algorithm and it means that it needs a careful planning; in this regard, it is considered the first problem to be evaluated.
The path depends mainly on the space and layout of the plant, which establishes physical and fixed constraints on the problem. Large spaces with high manoeuvring possibilities increase the number of admissible routes, reduce travel times and guarantee greater decision-making capacity for the central system in relation to the current handling flow.
The route has a significant effect on the time it takes for vehicles to perform tasks. The more alternatives and intersections there are, the more complicated it becomes to control the traffic in an efficient way.
Each layout can be represented by a graph $G(N, A)$, where N is the set of nodes that evidences the operating stations and A is the set of arcs that defines the connecting sections between adjacent nodes. Each node has at least two incident arcs, for incoming and outgoing movement respectively. The operating stations are locations in which it is possible to carry out picking and depositing operations and provide for a temporary release of the product pending the progress of sales or storage operations.
The arc direction indicates the vehicles direction flow and in each arc can be assigned a cost representing the distance between the designated nodes or the time required to travel itself.
In the literature various models have been presented, among which the most commonly used one has as objective function the minimisation of the distance travelled considering both transfers, when the vehicle is loaded and when it is empty (Kaspi et al, 2002).
Among the EMS rail systems, there are three main configurations when determining the route, presented by the literature. The literature does not explain where and when it is appropriate to use one specific design over another, the decision must be taken in relation to the requirements and characteristics of the system.
The conventional driving system is composed of a network of routes that connects all the stations, through straight lines, curves, intersections and shortcuts. The conventional system can be unidirectional or bidirectional in relation to needs and the number of rails. They involve a high level of

complexity, both computationally and in terms of decision-making, with considerable control and efficient difficulties.

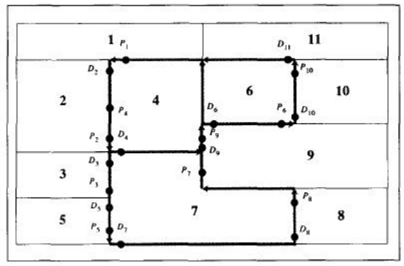

Figure 2.2: Conventional guide path system

The tandem configuration is based on the strategy of splitting frequently used routes into zones. It basically divides the entire route into non-overlapping circuits, where each loop is served by a single vehicle. The main advantage is the reduction of congestion, blocks and interference due to high traffic, but it restricts the number of vehicles to one per zone and involves the creation of innumerable restricted areas to satisfy large flows of movement.
The main disadvantage is the need to declare a zone inactive when its respective vehicle is inoperable due to malfunction.

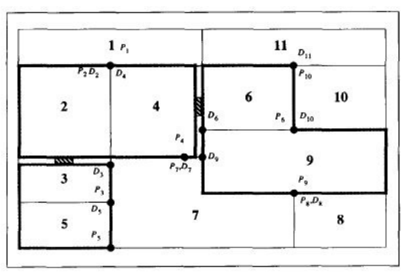

Figure 2.3: Tandem guide path system

In conclusion, the single-loop system is composed of a single path without any branches, in one direction and served by one or more vehicles. Bidirectionality in these cases is possible but involves possible interference between vehicles. This system ensures simplicity of implementation and operation with respect to the previous types, while considerably reducing the flexibility of the system in the face of problems such as vehicle malfunctions, which involve the shutdown of the entire system while awaiting the action of an operator who allows the vehicle to exit the circuit, typically transporting it to a branch set up specifically for maintenance.

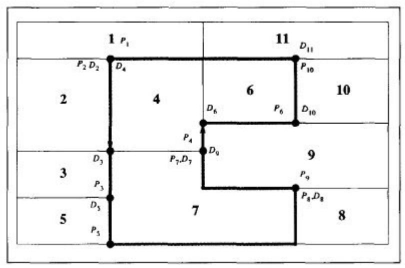

Figure 2.4: Single loop system

The production capacity of a single loop system is considerably reduced when compared with conventional route systems; in this regard a higher number of vehicles is required to obtain acceptable results. The characteristics of the single loop make it possible to avoid the problems of intersections on the path and considerably reduce the computational complexity, and consequently the final cost.
Within a single loop a barcode string, a subject discussed in greater depth in the following sections, determines the position of the vehicles. The barcode value is unique and incremental throughout the path, but, being a circuit, there is a junction point between the end and the start that involves a jump in the reading value. In this regard, to allow correct policies for determining position, for calculating the distance to be travelled and for process the vehicle's journey, it is convenient split the circuit into sub-sections.
A section is defined as a range of contiguous barcode values with start and end values, such that the constraint start value ¡ end value is always respected. Partitioning the route into sections implies the following advantages:

1. an increase in the safety distance between vehicles can be considered in curvilinear sections, where the vehicle encumbrance is greater;

2. the avoidance of queuing in determined zones, such as handling points for unloading and loading of the products, points of exchange with the maintenance branch etc;

3. the maximum speed is increased or reduced in relation to the type of section - rectilinear zones allow higher speeds than others - thus adapting the speed of the vehicles to the physical characteristics of the path.

The route structure is then composed by a set of sections, modelled as follows:

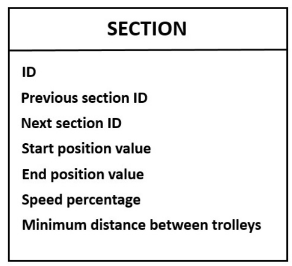

Figure 2.5: Section data type

The top three fields allow to generate a unique chain of sections, while the rest concern the topics discussed above.
In single loop systems, the journey - starting from a specific node to reach another one - is unique and it is not necessary to determine the path to take. The cost of the arc is calculated as the difference in position between end node and initial node, measurable using the *Compute distance to target position* algorithm, which takes into consideration the sections model.
During communication with the vehicle, the central calculator dynamically updates the distance that the vehicle must perform to reach the final position, which can be a node or the maximum distance that can be travelled due to the vehicles physical presence along the route. Since there are no crossings or other viable routes, the vehicle continues its movement in a straight line. Until the designated node is reached, where the missing distance will be zero.

2.3.1 Compute distance to target position algorithm

The following algorithm determines the distance between two different positions. The algorithm is performed by the vehicle's internal calculator and used in various functions such as speed calculation.

Algorithm 1: Compute distance to target position

```
Input  : sections, current position [mm], target position [mm];
Output: distance [mm];
current section = Get section from position (current position);
target section = Get section from position (target position);
if current section is equal to target section &&
  current position ≤ target position then
    result = target position - current position;
else
    result = end position value from current section - current
      position;
foreach section do
    next section = Get next section from ID(id from current section);
    if next section is equal to target section then
        result = result +
          (target position - start position value from next section);
        exit;
    else
        result = result +
          (end position value from next section - start position value
          from next section);
```

2.3.2 Get section from position algorithm

The following algorithm determines the section to which a given position belongs. To do this it uses the property of uniqueness of the position value within the route.

Algorithm 2: Get section from position

```
Input  : sections, position [mm];
Output: section;
1 foreach section do
    if start position value from section ≤ position ≤ end position
      value from section then
        result = section
```

2.4 Communication between entities

The need to be able to transmit data remotely, in real time, occurs in many situations and in areas that are very different from each other; in fact, often, the place where data is processed and managed is far away from the system that must execute them.
Communication in automated plants plays a fundamental role given the high quantity of information. An example of this is represented by the data relating to the mapping of the warehouse or the handling area, the information on the products and the fleet management in EMS systems.
In EMS systems, communication is bidirectional: according to the configuration of the vehicles, the vehicle contains one or more fixed antennas, which, connected to the internal calculator, guarantee the reception of information; and a transmitter, through which it sends the data to the system concerning its identity and status.
The central calculator controls and coordinates the fleet of vehicles and the remaining system components, such as working stations, conveyors and elevators, via an ethernet-type wired network.
The architecture most used for WLAN communication in EMS systems is called "access point based", in which, through one or more access points, it is possible to connect wireless equipment to a LAN network, conveying information on a wired support. It allows you to connect wireless mobile equipment to an existing fixed network, extending its connectivity. The main WLAN technologies available are infrared and radio frequency. Radio frequency transmission is much more used because, unlike with infrared, sending and receiving is possible even if the transmitter and the receiver are not visible, which is a typical circumstance in many operating conditions.
The type of frequency signal used is the broadband radio signal, so that, thanks to the European legislation ETSI 300-220, transmitting equipment can be used without the need for any license from the Ministry of Telecommunications. These so-called low power devices must transmit 0.01 W at low power, which also implies a reduction, practically zero, of the risks for the health of the operators; just think that a mobile phone transmits with a power of 2 W at 900 MHz. However, the low power limits the range of action, which goes from a minimum of a few tens of metres indoors to a maximum of around a hundred metres outdoors, in practice covering a thousand

square meters. However, in many cases, if the area to operate is greater, it is possible to position the various repeaters extending the coverage as desired with the only limit set by the management capacity of the software used.

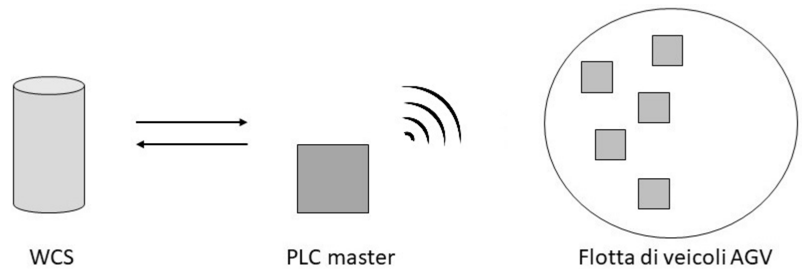

Figure 2.6: Communication topology

The main data transmitted, unavoidable for correct system and vehicle management, are:

1. the enabling: signal that enables the vehicle to move and whose usefulness comes into play when anomalous or emergency situations arise, and an immediate stop is required;

2. the route mapping: represents the sequence of the sections of the route. Saved and configured in the central calculator, it is sent to the vehicle upon specific request or when the model has been modified;

3. the destination value: indicates the position that the vehicle must reach - it can be a destination node or a momentary value due to the physical presence of other vehicles on the path;

4. the current position: the diffusion of the signal allows the central system to know in real time the position of each vehicle, avoiding collisions and congestion;

5. the present mission: represents the knowledge, by the vehicle, of having a mission assigned or not, since a movement can also be a request for removal from the current path;

6. the state of the sensors: identification of the state of the motor, of the barcode reader and of the photocells that detect physical presence. In order to find mechanical anomalies and damage of the components;

7. the loading and unloading permit: the central system, coordinating with the vehicles, allows the synchronisation of operations with the workstations, assigning and removing execution permits;

8. the keep alive bit: the parameter defined "keep alive" is of fundamental importance in determining network problems. It allows you to identify a non-explicit disconnection by the vehicle, making sure that the connection with the system is still active and working. A vehicle could be considered stationary in a given position since no updates are sent on its displacement, but on the contrary, physically the vehicle is moving. To recognise these emergency situations, where the logical representation differs from the physical one, the keep alive parameter makes it possible to determine, through a continuous exchange of signals, if there is a fault on the network, which causes delays in transmitting or receiving data. A delay in position reception is not allowed and can lead to collisions and subsequently requiring massive maintenance interventions.

The mission is a specific material transport task, including operations of retrieval and deposit from defined sources and to prescribed destinations. It does not represent a parameter to be sent, because it is composed of the destination value and the loading and unloading permit. Only the central calculator and the supervisor system need information about the status of a mission, while the operating vehicle is not responsible for it.
In the case of single-loop EMS systems, the route to reach a destination node is also a piece of information whose transmission is not required.
The same communication principles seen previously are reproduced for data transmission between the central calculator and the warehouse supervision system.
It is a duty of the supervisor to determine the priority of the missions to be carried out, the designation of the missions to the vehicles and the timing of entry and exit of goods in relation to the actual and future forecasts and then to communicate them to the central calculator. The information can come in real time from the plant, from others systems like the warehouse management system or from its own knowledge of the current progress of customer orders. It requires the information from various systems to make decisions that are best suited to optimising the flow.
Communication is therefore fundamental for the fleet control and the entire

system supervision, so as to instantly identify the correct arrangement in the succession of events and communicate it to the appropriate entity.

2.5 Material handling

Material handling is the process of transferring products from a source location to a destination. RGV vehicles have the task of positioning themselves in front of the working station, coordinating with the central system so that the picking operation takes place correctly and then moving towards the destination, carrying out the dual process.
The systems involved are the central calculator, which manages and controls the stations and the plant points, and the individual vehicles, which are responsible for moving the goods. Coordination and synchronisation are required during picking and delivering, both to ensure the right conclusion of the operation and to prevent alerts or anomalies.
The calculator must in fact recognise failed conclusion whether on the station side and on the vehicle side. The procedure takes place via two handshake signals, the handling permission and the handling in progress.
The respective entities recognise an anomalous situation in relation to the received handshake signals, which differ from the expected behaviour.
If either of the two entities behaves differently than expected, an alert is signalled during the transfer cycle and a maintenance phase is requested for the operators.
An approximate time is also indicated for the operation conclusion to avoid deadlock situations; if the times are not respected, an error message is sent to the supervisor system. So that the steps of each operation are executed and tracked, a finite state machine is used, which makes it possible to keep track of both, the steps and the errors, in which they occurred.
Each operation is preceded by three preconditions that must be respected in order to start the requested operation, material loading or unloading.

2.5.1 Loading and unloading preconditions

The preconditions differ in relation to the entities involved; they are a form of communication between two separately managed systems, so that control criteria are required both on the central calculator side and on the vehicle side. The conditions shown represent the goods loading operations; the unloading phase has similar constraints, but with a dual procedure.

The physical conditions of the central calculator determine the physical presence of products in a load bay and it is waiting to be picked up by a vehicle.

```
physical condition =
    physical presence in working station &&
    obstruction not intercepted &&
    vehicle is empty &&
    vehicle has reached the target position;
```

The logical conditions of the central calculator determine the logical presence of the products in the loading bays. This value keep track of the progress made and guarantee the constant control of the movement.

```
logical condition =
    mission type is equal to load operation &&
    vehicle is not in error mode &&
    permitted signal from vehicle = 0 &&
    rolling in progress signal from vehicle = 0 &&
    vehicle has not logical presence &&
    working station has logical presence;
```

The central calculator cycle conditions verify that there is no cycle in progress in adjacent system points and that no errors have occurred previously.

```
cycle condition =
    current cycle is not executing &&
    current cycle has not fault &&
    next cycle is not executing &&
    next cycle has not fault;
```

The vehicle's preconditions require fewer controls and are easier to calculate.

```
1 physical condition =
    vehicle position is equal to working station position &&
    vehicle is empty;

2 cycle condition =
    current cycle is not executing &&
    current cycle has not fault;
```

2.5.2 Handshake

As described above, in order to guarantee a correct communication between the systems involved it is required the usage of two signals: handling permission and handling in progress.
The first value represents a question to which corresponds a positive or negative answer, the second indicates the actual movement of the rollers by the corresponding entity.
A single signal does not allow for correct interpretation at the action end. If only the operation in progress signal is used, its conclusion could be due to either successful completion of the transfer or to shutdown due to anomalies. On the other hand, with the introduction of at least two signals it is possible to figure out two or more different irregularities.
The handshake phase always starts from the central calculator, which has: direct contact with the WCS, the complete plant management and a greater control of general operating; while the subject who completes the ending phase changes in relation to the action carried out.
Their behaviour is shown by the following images, first for the loading and then for the unloading procedure.
In the loading procedure the goods are moved from the plant point to the vehicle. The signals *permitted!* and *rolling in progress* are sent simultaneously from the vehicle to the calculator; ideally the one who receives the product moves its rollers before the source. This sequence means the goods will not get turned around and bent during the transfer.
To advise that the transfer has taken place correctly, the vehicle then resets

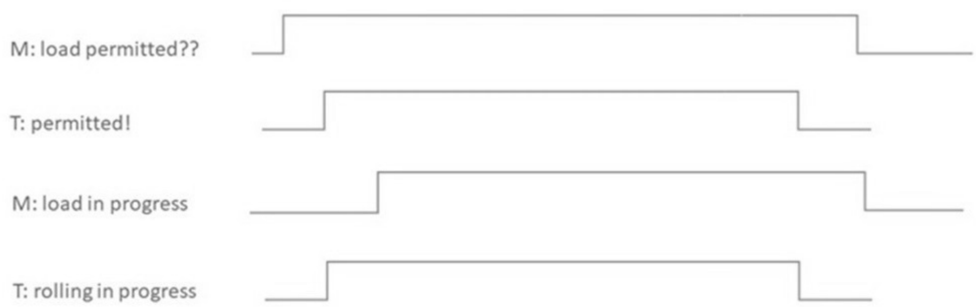

Figure 2.7: Handshake in load cycle

both signals and waits for the reply in order to consider the operation successfully completed. Upon receiving *permitted! = 0* and *rolling in progress = 0* in the same clock cycle, the calculator checks its own status and responds with the conclusion of the operation. At this point the vehicle is loaded and ready to move to a destination node.

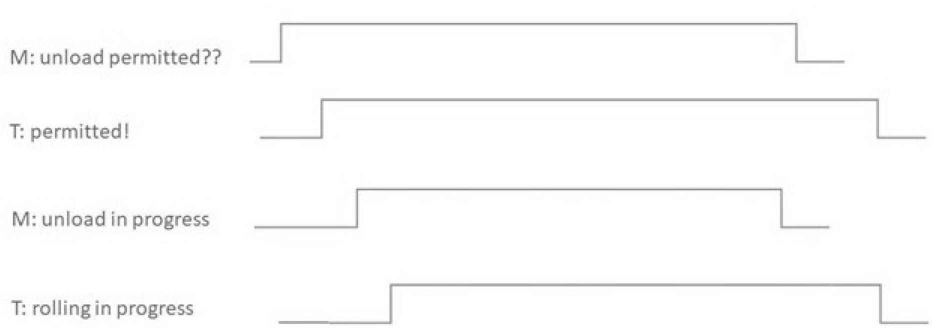

Figure 2.8: Handshake in unload cycle

The unloading procedure involves the transfer of the goods from the vehicle to the designated plant point. The handshake is very similar to the previous one; where the central system is the initiator of the operation and the vehicle the receiver.
In the current circumstances, the vehicle awaits the movement of the rollers on the part of the station, which appears as the product destination and,

only after receiving the signal *unload in progress = 1*, the transfer begins, signalled by *rolling in progress = 1.*
The central system advises the vehicle that the handling has come about correctly by sending signals *unload permitted?? = 0* and *unload in progress = 0.* The conclusion is subsequently brought about with a positive response from the vehicle.

2.5.3 Finite state machine

Loading and unloading operations do not only involve the affected systems, but also a set of sensors, inverter and motors, such as photocells and position control devices. To ensure supervision and management of the transfer, simple finite-state machines are built for the central calculator and the vehicle, where the chain of steps represents the correct sequence of actions up to the conclusion.
The states are represented by circles and named according to the step, while the transactions represent the occurrence of definite events. The starting states are named *idle*, while the conclusion states are *M6* and *T6*.
The finite state machines for loading procedures is shown below and the states of the central calculator are explained in-depth.

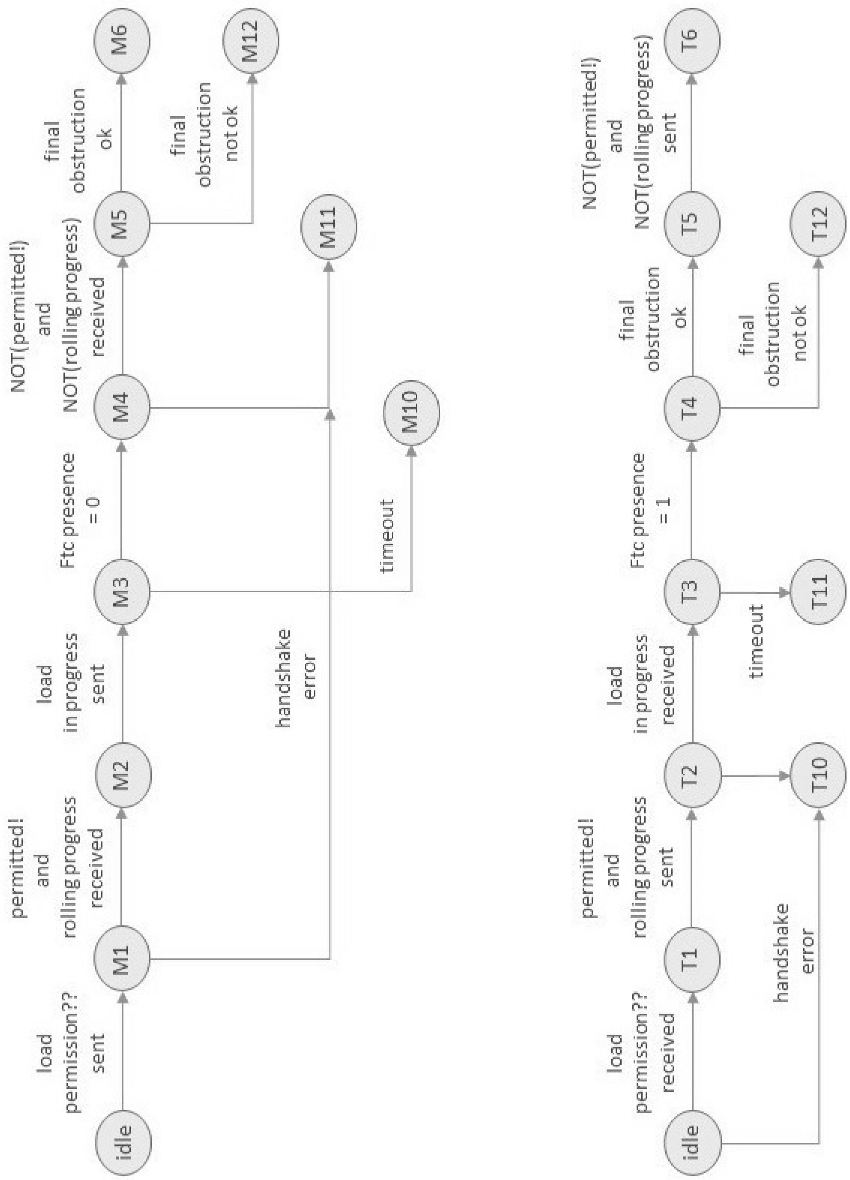

Figure 2.9: Finite state machines for loading procedures

In state *M1* the two systems begin the handshake phase, the product is physically in the loading area, indicated by the photocell that detects physical presence, and it is waiting to be transferred. There is also a second photocell, called obstruction or anti-fall, which checks that there are no obstructions before, during and after the movement between the station and the vehicle.

```
M: load permission?? = 1 & load = 0;
T: permitted! = 0 & rolling in progress = 0;
FTC presence in working station = 1;
FTC obstruction = 0;
Timer = 0;
```

In state *M2* the handshake phase continues, the vehicle has consented to the load request by the central calculator and has started moving the rollers whilst waiting to receive the product. The goods are still physically at the plant point, awaiting the activation of the rollers.

```
M: load permission?? = 1 & load = 0;
T: permitted! = 1 & rolling in progress = 1;
FTC presence in working station = 1;
FTC obstruction = 0;
Timer = 0;
```

In state *M3* the first handshake phase ends, the central calculator moves the rollers and the product transfer begins.

```
M: load permission?? = 1 & load = 1;
T: permitted! = 1 & rolling in progress = 1;
FTC presence in working station = 1;
FTC obstruction = 0;
Timer = start;
```

In state *M4* the physical movement is completed, the product is on the vehicle and the presence photocell of the station indicates the absence of the

```
M: load permission?? = 1 & load = 1;
T: permitted! = 1 & rolling in progress = 1;
FTC presence in working station = 0;
FTC obstruction = 0;
Timer = stop;
```

goods.
In state *M5* the second phase of the handshake began, with the task of advising the central calculator that the transfer has been completed successfully. The vehicle resets its signals in the same clock cycle and waits for a response. At this stage the product is physically inside the vehicle.

```
M: load permission?? = 1 & load = 1;
T: permitted! = 0 & rolling in progress = 0;
FTC presence in working station = 0;
FTC obstruction = 0;
Timer = 0;
```

The last state *M6* concludes the procedure and the second phase of the handshake. The central calculator has checked that the obstruction photocell does not present anomalies and afterwards resets its own signals. At this reception the vehicle considers the loading phase correctly completed and starts transit to the designated destination node.

```
M: load permission?? = 0 & load = 0;
T: permitted! = 0 & rolling in progress = 0;
FTC presence in working station = 0;
FTC obstruction = 0;
Timer = 0;
```

State *M10* occurs when the goods handling timeout is engaged; the time taken for the transfer is higher than standard one and a situation of discrepancy is announced. The central calculator reports the status to the vehicle and stops the procedure.

```
M: load permission?? = 1 & load = 0;
T: permitted! = 1 & rolling in progress = 1;
FTC presence in working station = 0;
FTC obstruction = 0;
Timer = 0;
```

State *M11* implies an unexpected reception of the handshake signals. The ensuing state involves an anomaly and requires the assistance of operators for verification.

```
M: load permission?? = 1 & load = 0;
T: permitted! = 0 & rolling in progress = 1;
  or
  T: permitted! = 1 & rolling in progress = 0;
FTC presence in working station = 0;
FTC obstruction = 0;
Timer = 0;
```

In state *M12* the procedure has not completed correctly; the obstacle photocell has intercepted an anomaly and blocked progress. Upon the occurrence of this irregularity, a notification is sent to the supervisor system and the maintenance intervention by the operators is awaited. The calculator alerts the vehicle by resetting only the *load* signal, which recognises the stall situation and stops the movement.

```
M: load permission?? = 1 & load = 0;
T: permitted! = 0 & rolling in progress = 0;
FTC presence in working station = 0;
FTC obstruction = 1;
Timer = 0;
```

2.6 Idle vehicles

Having vehicles without a mission is inevitable in an RGV system and, rather than forcing them to return to parking areas, it is better planning strategic points in the proximity of product release locations or loading bays.
There are two main strategies for the position of vehicles: static and dynamic. In the static strategy the vehicle parking locations are selected to minimise response time for a new movement request. Among them the distributed-positioning rule uses multiple stopping points, which represent the stations where it is possible to operate a loading action. The mission management algorithm for *idle* vehicles is carried out by the central calculator which, in relation to the state of the system and of each individual vehicle, assigns the plant point closest to it.
When a vehicle becomes idle, the calculator assign a re-positioning mission to it for the nearest stopping point, waiting for a subsequent task.
In the event that, during the journey, stationary vehicles are encountered, the re-positioning algorithm makes a request for their movement owing to the obstructed path. The same procedure is carried out when the selected loading station is occupied by other idle vehicles. To which a new plant point will be assigned. This strategy is valid if the number of stations is considerably higher than the number of vehicles in the system.

2.6.1 Assign strategic missions algorithm

Algorithm 3: Assign strategic missions

```
Input  : vehicles, missions, bays;
Output:
1 foreach vehicle do
      if vehicle has a mission to do then
          mission completed = false;
      job mission =
        vehicle has a mission to do && mission type is a job;
      if job mission is false then
          mission completed = true;
      if mission completed is true then
          station to stop = Get next closest bay where type is load;
          mission type = re-positioning;
          mission status = moving to destination;
          mission destination = station position;
          vehicle has a new mission assigned;
```

2.7 Re-positioning management

In EMS systems, displacement requests caused by the obstruction of the route are a frequent factor due to the presence of a single track on which the vehicles move. An idle vehicle placed in front of a loading bay, whilst awaiting the assignment of a mission, could cause an obstacle for all subsequent vehicles passing through that position. For this reason, a re-positioning request algorithm is essential for a correct system design.
The central calculator executes the algorithm, which is aware of the layout, the vehicles position along the route and the missions agreed.
When the calculator recognises that the previous vehicle is blocked it assigns a blocking flag and checks the tasks assigned to the successive vehicles. Re-positioning occurs when the vehicle has no mission assigned or when a request for re-positioning is in progress but with insufficient distance in order to clear the path.
The following algorithm reassign the position of all the idle vehicles which hinder the journey of a mission, until arriving at the condition for which the previous vehicle has a mission destination $<$ mission destination from the actual vehicle, so that the destination transmitted is further away than the destination of the blocked vehicle.

2.7.1 Re-positioning request algorithm

Algorithm 4: Re-positioning request

Input : vehicles, bays;
Output:

1 **foreach** *vehicle* **do**
    if *previous vehicle is blocking* && *mission status is empty* **then**
        station to stop = Get next closest bay where type is load;
        mission type = re-positioning;
        mission status = moving to destination;
        mission destination = station position;
        vehicle has a new mission assigned;
    if *previous vehicle is blocking* && *mission type is re-positioning* **then**
        if *mission destination from previous vehicle* ≥ *mission destination from vehicle* **then**
            new destination = mission destination from previous vehicle + vehicle size;
            station to stop = Get next closest bay where type is load and position is equal to new destination;
            mission destination = station position;

2.8 Position

In relation to the type of AGV system, different technologies are used to define the position: vehicles with laser triangulation use a laser device that rotates 360 degrees and scans the surrounding area; vehicles equipped with GPS adopt the differential technique with a fixed reference base in order to accurately recognise the position; and finally, for the vehicles on tracks, position is established along the translation axis by a BCL sensor, positioned on the driving wheel, which continuously reads a barcode string placed on the track. The barcode is a system for transmitting information by storing it in an optical figure composed of black and white lines, organised as parallel bars and spaces with different thicknesses. The barcode can be defined as a symbol for coding information in a format that can be acquired automatically and cyclically by appropriate optical readers. The aim is to forward the information in an automated processing system, guaranteeing its handling in an easy, secure and economical manner.
In rail systems, the information contained in the bar code represents a numerical value, such as the distance from the origin point. It involves an ever-increasing value that is unique within the single-loop up to the exchange point where there is a numerical jump to the origin point.

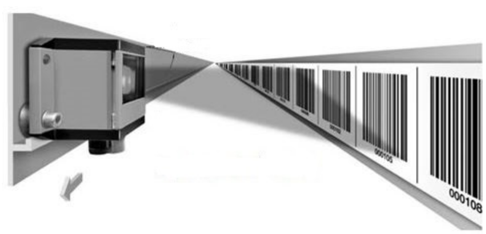

Figure 2.10: BLC sensor

In all barcode-based identification systems, four fundamental characteristics must be taken into account: reliability, automation, precision and speed.

1. Reliability: for each reading operation, it is required to perform a validation action before considering the information correct. In RGV systems a wrong validated reading can lead to a vehicle logical position different from the physical one. Which implies mistaken assessments by the central calculator and consequent possible collisions.

2. Automation: automatic and continuous data intake. Once installed and configured the device performs the reading process continuously.

3. Accuracy: barcode-based systems guarantee a high level of precision. In vehicles, the device reads from three to four codes simultaneously, carries out the arithmetic average and returns the value. This procedure reduces the error level due to worn or dirty strings.
   The position of the device is also important to ensure accuracy in reading. To read the barcode it is necessary to scan it along its entire length. If it is too low, a slightly oblique scan, especially if done from a distance, will not allow the optical reader to obtain the correct acquisition.

4. Speed: the reading operation takes place at each clock cycle of the internal calculator of the vehicle, to the order of 250 scan/s.

As previously indicated, the decoding of the value detected by the barcode reader is a fundamental part because it places the vehicle in a precise point of the track. To detect any possible reading errors, it is possible to check - if the path is one-way like in the single loop - that the new value sites in the same section as the previous one or, at most, in the next one. When this condition is not met, the vehicle reports a position that is not compatible with its movement and it is signalled as an error situation where the vehicle must stop and wait for confirmation or assistance by the operator.
Given that vehicles can only continue in one direction and that the route is subdivided into sections as shown in the previous chapters, the reading value during motion will increase within the same section.
This constraint guarantees better control over vehicle positioning. Indeed, the following value must correspond to the previous one plus a delta value, calculated in relation to the current speed of the vehicle. If the vehicle is travelling on a section with high speed, the delta will be greater than a stationary vehicle.
Assuming that at time t the vehicle is positioned at the validated position q of the section t and that the validity interval is calculated by:

$$delta = actualspeed \cdot clocktime \qquad [mm] \tag{2.1}$$

$$actualbarcodevalue \leq previousbarcodevalue + delta \tag{2.2}$$

we have that:

$$actualbarcodevalue \in actualsection \tag{2.3}$$

or

$$actualbarcodevalue \in nextsection \tag{2.4}$$

2.8.1 Barcode value validation algorithm

Algorithm 5: Barcode value validation

```
Input  : sections, delta [mm], previous barcode value [mm], current
         barcode value [mm];
Output: current position [mm];
current section = Get section from position (current barcode value);
upper bound = Add offset to position
  (previous barcode value, delta);
lower bound = Subtract offset to position
  (previous barcode value, delta);
upper bound section = Get section from position (upper bound);
lower bound section = Get section from position (lower bound);
foreach section do
    if current section ID is equal to upper bound ID then
        if start position of upper bound section ≤ current barcode
          value ≤ upper bound then
            result = true;
            current position = current barcode value;
    else
        if lower bound ≤ current barcode value ≤ end position of
          lower bound section then
            result = true;
            current position = current barcode value;
```

2.9 Speed

EMS vehicles, in order to travel the designated routes, need at least one electric motor, typically a first one for travel and a second one for moving rollers and other components.
The inverter distributes power and voltage to the motor, assigns fixed parameters in relation to usage, and enables the movements. The vehicle's calculator interfaces directly with the inverter, communicating in order to obtain information about the status of the motors and to control their speed, expressed in revolutions per minute.
Consent to movement comes about via reception of the enabling signal from the central system and the destination value that is different from the current one, with the addition of a tolerance range. The vehicle moves according to uniform rectilinear motion and uniformly accelerated rectilinear motion: given the distance from the destination, the current speed and the maximum speed of the current section, it is possible to calculate the set point for the next time instant.
The UARM provides vehicle acceleration and deceleration with constant values. This motion allows the vehicle to follow a regular and predictable speed profile. The maximum speed is calculated via three factors: type and characteristics of the motor, transit speed in the current section and maximum speed of the system, both expressed as a percentage.
Once reached, the vehicle will travel the section with uniform straight motion, excluding the presence of other vehicles or obstructions that require deceleration.
The UARM equation:

$$x(t) = x_0 + v_0 t + \frac{1}{2} a t^2 \tag{2.5}$$

$$v_\mathrm{t} = v_0 + at \tag{2.6}$$

The unidirectional motion where speed
and acceleration are in the same direction:

$$v^2 - {v_0}^2 = 2a \cdot (x - x_0) \tag{2.7}$$

Where:

- $x(t)$ is the position at time t;
- x_0 is the initial position;

· v_0 is the initial speed;

· a is the acceleration constant.

The increment of speed can happen to a stationary vehicle or a vehicle entering a section where the maximum permitted speed is higher than the speed allowed in the previous section. Acceleration takes place when the vehicle has already entered the new section; on the contrary, in deceleration, the vehicle must decrease its speed to have a maximum speed in the following section that is not higher than the allowed one. This implies a deceleration that takes place near to the new section, whilst the vehicle is still physically present in the previous section. This deceleration situation occurs during arrival at the destination node where the final speed must be zero.

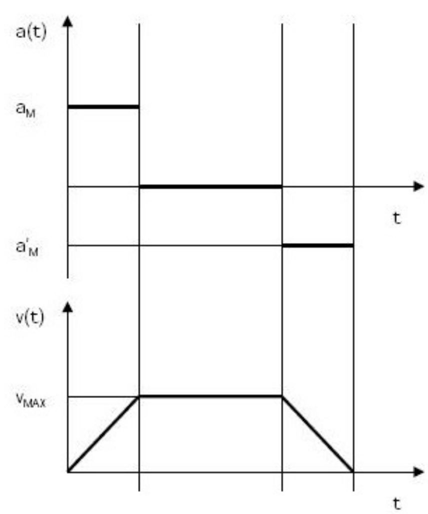

Figure 2.11: UARM speed diagram

The operational data are: translation speed between 1500 and 2000 mm/s in a straight line, between 1000 and 1500 mm/s in curvilinear section and in junctions; acceleration and deceleration about 200 mm/s2, while the most modern vehicles can have an acceleration of as much as 1000 mm/s2. Pallet handling time, loading or unloading from 20 to 30 seconds.

2.9.1 Speed management algorithm

The speed management algorithm is calculated internally the vehicle, so as to reduce the burden of computation to the central system and ensure greater responsiveness. When the vehicle is near a section, on which a lower maximum speed than the current one is configured, the deceleration procedure starts, so as to enter the section with an appropriate speed in regard to the required characteristics.
The algorithm searches for the next sections within the vehicle stopping space, calculated according to the UARM, evaluates the maximum speeds configured and identifies the upper limit. If the current speed is lower than the maximum speed allowed by the section and the vehicle is not decelerating, it is possible to accelerate by following the motion laws.

Algorithm 6: Speed management

Input : sections, system speed [%], motor parameters, acceleration constant, distance to target [mm], previous set point [$\frac{mm}{s}$];
Output: next set point [$\frac{mm}{s}$];

current section maximum speed =
  maximum speed from motor parameters $\cdot \frac{section\ speed}{100} \cdot \frac{system\ speed}{100}$;
next section maximum speed =
  maximum speed from motor parameters $\cdot \frac{next\ section\ speed}{100} \cdot \frac{system\ speed}{100}$;
next speed square= $target\ speed^2$
  $+\ 2 \cdot acceleration\ constant \cdot distance\ to\ target$;
next speed = $\sqrt[2]{next\ speed\ square}$;
speed upper bound =
  min(next speed, current section maximum speed);
speed lower bound = 0.0;
if *previous set point* < *speed upper bound* **then**
  next set point = previous set point +
   acceleration constant $\cdot \frac{clock\ time}{100}$;
next set point = min(next set point, speed upper bound);
next set point = max(speed lower bound, next set point);
return next set point;

The variable *target speed* can assume two values: it sets 0.0 in case of vehicle stopping at destination, or the previously calculated value by *next section maximum speed* in the case in which the vehicle is closer a section with lower permitted speed. The parameter *distance to target* represents the distance between the current position and the destination node; calculated using the algorithm 1: *Compute distance to target position* of guide path section. The value *next set point* is secondly converted in revolutions per minute and sent to the inverter that will take care of the direct communication with motors.

2.9.2 Speed management example

When a mission is assigned to the vehicle, whether it is a job or a re-positioning request, the vehicle starts from standstill and then continues with the acceleration phase until reaching the maximum allowed speed. The presence of sections with different maximum speeds entails a stepped increase in speed, as shown in the second graph. The second acceleration phase starts only when the vehicle is completely within the section. The x-axis represents the space, the y-axis represents the speed and the dotted line indicates the beginning of a new section. According to the law of UARM, it takes 2.7

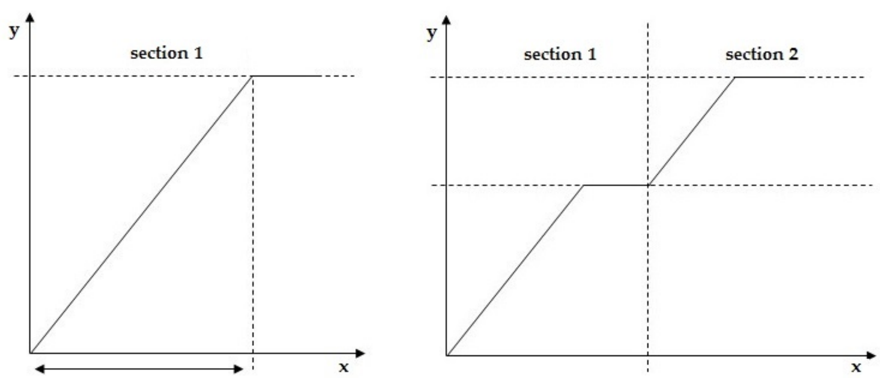

Figure 2.12: Acceleration phase example

1 start set point = 0.0 $[\frac{mm}{s}]$;
2 end set point = 1700 $[\frac{mm}{s}]$;
3 acceleration constant = 200 $[\frac{mm}{s^2}]$;
4 section 1:
   speed = 60%;
   distance required to reach the max speed = 2700 $[mm]$;
5 section 2:
   speed = 100%;
   distance required to reach the max speed = 4600 $[mm]$;

meters in section 1 to reach the maximum speed of 1020 mm/s through the acceleration constant set; while in section 2 are needed 4.6 meters, starting from 60% and reaching 100% of the speed. The speed percentage of section 1 puts an upper limit on the maximum speed so that the vehicle cannot accelerate until the end of the section. When the vehicle has the task of positioning in a destination value, whether it is a loading or unloading node or an obstruction due to the presence of other vehicles on the route, in relation to its current speed it must start the deceleration phase with sufficient time to stop, according to the law of motion. The speed reduction procedure for the presence of a new section follows the same principles with the appropriate ratios. Deceleration follows a constant profile both for the stopping procedure and for speed reduction near new sections.

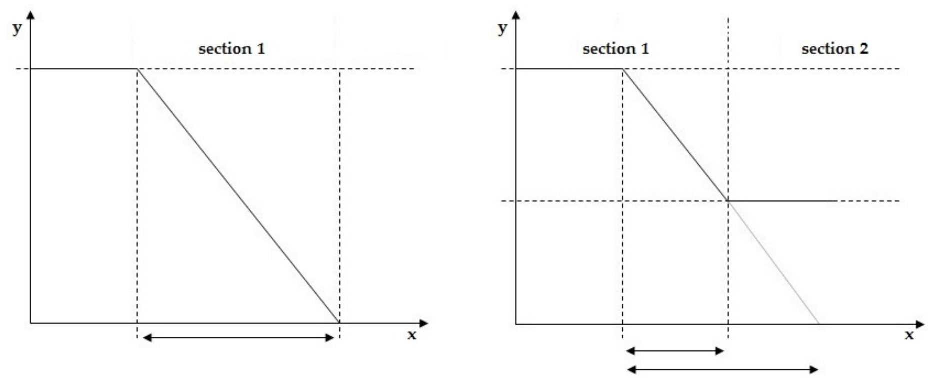

Figure 2.13: Deceleration phase example

current set point = 1700 $[\frac{mm}{s}]$;
target set point = 0.0 $[\frac{mm}{s}]$;
deceleration constant = 200 $[\frac{mm}{s^2}]$;
section 1:
  speed = 100%;
  distance required to stop = 7300 $[mm]$;

When a vehicle has to enter a section where the maximum allowed speed is lower than the current speed, although being physically in the previous section, the deceleration phase must begin in such a way as to arrive in the next section at the required speed.
Once section 2 has been identified as the next one to be transited by the

```
current set point = 1700 [mm/s];
target set point = next section maximum speed;
deceleration constant = 200 [mm/s^2];
section 1:
   speed = 100%;
   distance required to reach the new max speed = 5400 [mm];
section 2:
   speed = 50%;
```

vehicle, and as having a maximum permitted speed lower than the current one, the vehicle, using the speed management algorithm, at a distance of 5.4 meters from the beginning of the section starts the deceleration phase in order to respect the established limit for the new section.

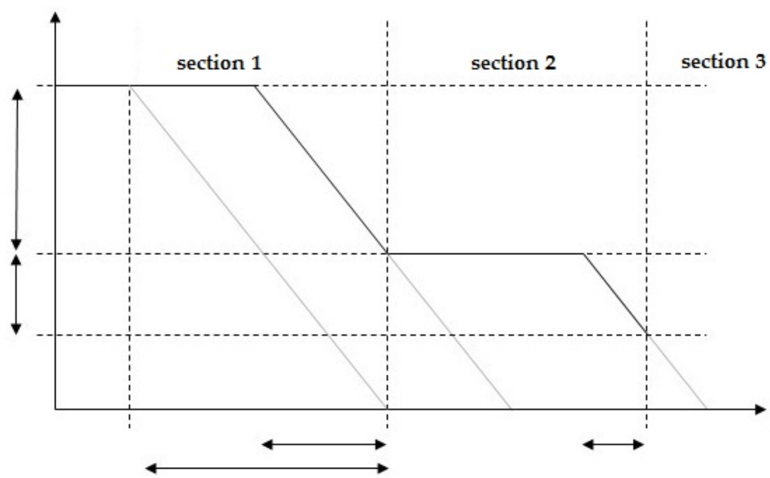

Figure 2.14: Multiple deceleration phase example

The same procedure is followed in the presence of contiguous sections with different speeds, such as section 1 where the speed is 100%, section 2 at 50% and, in conclusion, section 3 at 25%. The size of the sections is sufficiently large such that the stop ramp is always guaranteed; indeed in section 2 the vehicle continues in uniform straight motion up to 1,350 meters from the beginning of section 3.

Chapter 3

Mission scheduler

Scheduling involves a methodologies family based on mathematical techniques, which support the decision-making process and are used in many problems, in particular in manufacturing industries for transport and logistics. A scheduling problem consists, in its most general formulation, in the optimal management of limited resources, designed to carry out certain activities while respecting appropriate constraints; the target is to evaluate usage times of the resources and to assign material handling activities, called missions, to the individual AGVs; respecting constraints and following suitable optimization criteria.
Once the hypotheses are defined and accepted, the scheduling problem can be tackled and solved with a wide range of tools; in essence, these are algorithms that provide possible solutions, bearing in mind the production context in which they operate and the objective function to take into account.

3.1 The scheduler

The scheduling is a decision-making process frequently used in many industries in the manufacturing and services fields; it plays an important role in improving production flow. It deals with the allocation of resources to tasks, in defined time periods and with the aim of maximising/minimising one or more parameters.
Resources and tasks can take different forms. Resources can be machines in a work environment, processing units in a computer and so on. Similarly, tasks can be operations in production processes, arrivals and departures in airports, execution of programs in computing and others. Each task has definite properties including the priority level, moment of release, time required for execution and due date.
The objective functions can differ as well, from the minimisation of completion time of the last task to the maximisation of the number of tasks completed before their respective deadlines.
Scheduling is a complex operation with the aim to establish the workload of individual machines in the short term and to assign the planned missions. Precisely because of this complexity, the objective that drives this operation is not the costs minimisation, as the case for medium and long-term planning, but rather the optimization of the parameters that are representative of the system performances.
Many companies use schedulers - software supports dedicated to the scheduling phase - to process data relating to the distribution plans. Normally they have a wide range of solutions and make it possible to vary in a multitude of ways, both the objective functions and the methods - typically heuristic - that are used for scheduling.
In a production context, the scheduler must interact with many other functions located within the corporate IT system. Interactions are system-dependent and can differ greatly from one situation to another. The local network of terminals, workstations and machines, connected directly or indirectly to the central system, is used to insert and to retrieve new information.
Ideally, the interaction should enable the quality of data and guarantee detailed and complete tasks, which help the scheduler to find the solution and maintain a high level of efficiency. Unfortunately, in real production environments, we rarely have tasks with rich information and, since the evolution of the system is dynamic, the construction of a simple deterministic model ends up being incorrect and inadequate.

To this end, online models are used when there is no need to know, at time zero, all the operations that must be scheduled, and they make it possible to work with tasks with scant information. This is known as non-clairvoyant scheduling and is more able to adapt to modern working environments.
It may not be immediately clear the impact that the scheduling has on the objectives involved, but studies and practice have shown that the use of a suitable scheduler is considerably important for the performance of the system and it is opportune to dedicate time and effort to find an appropriate one. The realisation of a scheduling algorithm can be complicated both from a technical and implementation point of view. The types of difficulties encountered on the technical side are the same as in other forms of combinatorial optimization and stochastic models, while the difficulties of implementation are a completely different kind - these could depend on the accuracy of the model used for the analysis and the reliability of the input data. Proper implementation requires time and research in order to achieve reasonable efficiency and adequate control, reaching or approaching the predetermined objectives.

3.2 Scheduling terminology

In all scheduling problems the number of tasks and resources is considered finite. The number of tasks or operations is denoted by the letter n and the number of resources or machines by m. Typically, the lower index j refers to an operation while the index i refers to a machine.
If the execution requires a number of steps then the pair *(i,j)* refers to operation j performed on machine i.
The main tasks properties are:

· Execution time: p_{ij}
The field p_{ij} represents the time required to carry out operation j on machine i. The index i can be omitted if the time does not depend on the assigned machine or if only one machine can perform that specific operation.

· Moment of release: r_{j}
The moment of release of operation j, often called *ready date* or *activation date*, represents the instant in which the operation enters the system and is ready to be computed.

- Due date: d_j
  The due date of operation j represents its completion date or the commitment that the company has made to the customer as regards the conclusion of the order. The conclusion of a job, after its due date, is allowed but incurs in penalty. Whereas when the due date is considered mandatory and absolutely must be respected, it is called *deadline*.

- Weight : w_j
  The weight w_j of the operation j is essentially a priority factor, which denotes the importance of j with respect to the remaining operations in the system. For example, the weight could represent the products sequence during the loading phase in a truck.

As far as resources are concerned, different typologies are identified, such as processing on: single machine, identical machines in parallel, machines in parallel but with different speeds and so on. Among them, we examine in detail only the presence of identical machines that carry out operations in parallel, which will be the focus of this thesis and scheduler.

- Same machines in parallel: P_m
  In the system, there are m identical machines that are able to carry out defined operations in parallel. Operation j requires a single process and is assigned to a single machine i.

Some of the main features about machines are:

- Precedence constraints: *prec*
  The precedents occur in environments with single or parallel machines and require the conclusion of one or more operations before the others may begin their execution. There are different forms of precedence: if each operation has at least one predecessor and one successor, it is called *chain*. If it only has a successor, we talk about *in-tree* whereas, in conclusion, if it only has a predecessor, it is defined *out-tree*.

- Blocking: *block*
  Blocking is an event that occurs mainly in processing flows. If the resource has a limited buffer then, when they are all full, it may not be possible assign new operations; they are placed on hold until a subsequent space release.

The objective functions differ considerably depending on the scheduler context; many functions are in time domain and concern the completion time of the tasks or of further operations to be performed subsequently. The completion time of operation j on machine i is denoted C_{ij}. The instant in which operation j leaves the system is defined as C_j.
Some examples of objective functions are:

· Makespan: $C_{\max}$
The makespan, defined as *max(*C_1*,* C_2*, ...* C_n*)* is equivalento to the completion time of the last operation to leave the system. A minimum makespan value usually implies correct utilization of machines.

· Maximum Lateness: $L_{\max}$
The maximum lateness is defined as *max(*L_1*,* L_2*, ...* L_n*)* and measures the worst violation of the due dates.

Terminology makes a distinction between sequence and schedule. A sequence usually corresponds to a simple permutation of the n operations or the order in which operations are to be processed on a given machine.
The schedule refers to an allocation of operations within a more complicated setting of machines, allowing multiple policies such as precedence, blocking and more.
In recent years, a great deal of research on scheduling has been devoted to discovery efficient algorithms. The results have shown that it is often not possible to obtain optimal solutions in polynomial time; these problems are called *NP-hard*. Confirming that a problem is NP-hard requires a formal mathematical demonstration. Research has also focused on the boundary line that distinguishes problems that can be solved in polynomial time and in NP-hard solutions.

3.3 Online scheduling

In the theory about main scheduling problems, it is assumed that the input data are all known at time zero so as to allow the decision-making process to determine the entire schedule, thanks to all the information at hand. This standard paradigm is often referred to as *offline scheduling*.
In manufacturing environments, it is difficult to know or predict in advance the flow of operations to be carried out, especially in contexts such as logistics and transport where the handling material is irregular and unpredictable. In this regard, the adoption of offline scheduling would not be possible, and a new category of problems must be addressed.
In *online scheduling* the decision-maker does not know how many operations are going to be processed and their respective properties. It becomes aware of the existence of an operation only when it is released by the system and presented to him, which can take place at any time during processing.
The scheduler must be frequently able to accept new operations and detect the correct schedule with the information currently available. The online scheduling problems seen in the literature are typically referred to *online over time scheduling problems*.
A well-known algorithm for a simple resolution to online over time scheduling problems goes by the name of *LIST scheduling*. The operations of the decision-making process are initially ordered according to established criteria and subsequently inserted into a list. Whenever the assignment of a task to a resource must be considered, it checks the list and selects the next operation. When a resource completes an operation, the decision-maker runs through the list and assigns the next one. When an operation is issued and the process has assigned the established priority to it, the operation may have to wait until one or more resources become idle before being assigned and start its processing. If the objective function is a regular performance measure, then it may not take sense for the decision-maker to leave a machine idle when there are still one or more operations on the list.
This procedure minimizes idle states in the machines and is perfectly able to handle parallel machine problems, identifying a valid deal between simplicity and precision of the solution.
Online scheduling in a parallel machine environment has received a significant amount of attention during the last couple of years. Because it establishes a bridge between deterministic and stochastic scheduling. In stochastic scheduling, decisions have to be made with high conditions of uncertainty

and with a limited amount of information available. However the stochastic paradigm is still quite different from the online paradigm; results show that the solutions obtained in online scheduling often correspond to bounds in stochastic scheduling.

3.4 Types of algorithm - the dispatching rules

There are three main categories of scheduling algorithms:

- *Exact algorithms* can find optimal solutions;
- *Approximation algorithms* produce solutions that are very close to optimal and that are guaranteed to be within a fixed percentage of the actual optimum. They are computationally fast and have polynomial complexity;
- *Heuristic algorithms* obtain results which are not guaranteed to be close to the optimum. Heuristic methods tend to be rather generic and can be adapted to a wide variety of problems. The performance of there methods is often evaluated empirically.

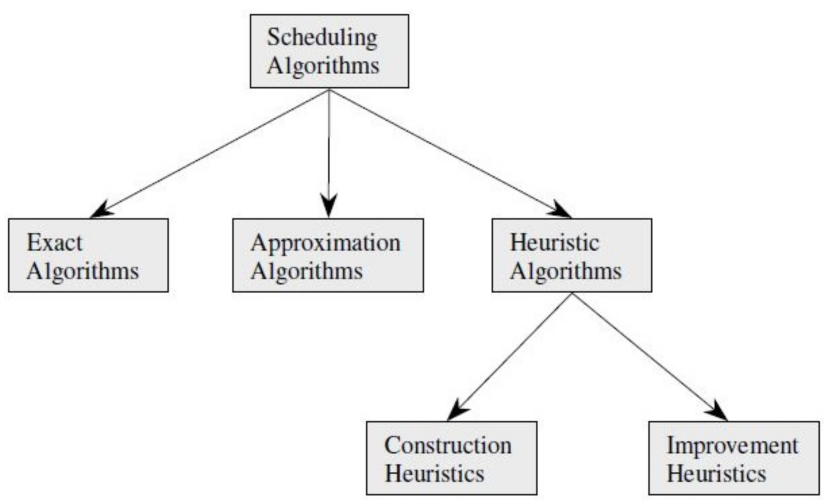

Figure 3.1: Types of algorithm

Heuristic algorithms can be classified as construction heuristics or improvement heuristics: in construction heuristics, the initial phase takes place without a schedule and each time the system adds the operation entering the system, whereas improvement heuristics begin with a schedule of the present operations and, in a given time frame, try to find a better solution than the current one, from the study of similar schedules.
Dispatching rules are an example of construction heuristics. Their main advantages are simplicity of implementation, even for particular cases, speed of calculation and the possibility to find good solutions in relatively short time frames. Whereas their disadvantages concern the identification of unpredictable schedules when the rules that determine it are highly articulated.
Dispatching rules can be divided into basic rules and composite rules; composites rules are obtained by the combination of two or more basic rules and can guarantee significant improvements. The composite dispatching rules are expressions that combine a number of n rules; each rule is ordered via a set of specific measures. The result can be weighted by multiplication factors with respect to the application context.
The ordering of the rules can be fixed, chosen by the designer through decision-making constraints, or variable, in relation to special operations or time functions.
Much research has been carried out in the last decade concerning dispatching rules and notable rules have been studied in the literature. From a practical point of view, the objective functions are considerably complicated, for example a realistic one is given by the combination of different rules that are fixed, variables and in the time domain.
In the following sections, three algorithms are presented which adopt dispatching rules as a scheduling strategy and which, via fixed rules, determine the priority of transport missions in a logistical environment, before assigning them to AGV vehicles for their execution.

3.5 Mission scheduler

The mission scheduler describes a set of algorithms aimed at determining the correct sequence of operations to be performed, respecting complex logistical criteria, and defining the assignment of missions to AGV vehicles.
The following scheduler, which adheres to dispatching rules theory, adopts a set of fixed and ordered rules with regard to technical specifications received from the transport management system TMS, in order to define the priority of material handling missions during the input and output phase from automated warehouses equipped with autonomous guided vehicles AGV.
It is known that during the transport of goods, the number of clients that the vehicle can satisfy typically ranges from 3 to 8 and it is necessary to respect internal constraints regarding positioning of the products so that the driver can unload the load in the shortest possible time.
The most commonly practice is the Last in First out policy, in which the last product loaded onto the vehicle is the first to reach the designated destination, followed by the others. The data - which will become rules within the scheduler - deriving from the transport management system include:

- *Loading station*: is the place where the truck is located - this also corresponds to the destination point of the missions;
- *Activation date*: the date to which the operation of entry onto the warehouse or sale of products corresponds;
- *Vehicle's daily number*: a unique number to define the daily sequence of the trucks;
- *Internal position*: the product position with respect to the truck.

The remaining useful data collected from other systems such as the company management system, include:

- *Movement type*: the type could be an entry into the warehouse, an internal movement or a sale of goods;
- *Product*: the product with the associated mission.

Combining these data enables a correct calculation of the priority value to be assigned to each mission and subsequently to each vehicle. Missions are created when a product requires a displacement, coinciding with a handling

phase within the warehouse. Reasons may vary, such as for example the entry of new finished or semi-finished products for which the transportation is required from acceptance station to internal location, or the sale of goods for which displacement takes place from the warehouse location to the destination bay for loading the truck.
In an environment where the regular flow of handling materials coming into and leaving the warehouse is higher than 3000 units per day, it is essential to guarantee correct scheduling in order to satisfy transport constraints and reduce waiting times. The priority value, once attributed to the corresponding mission, is used to establish the correct sequence of the goods to be handled, and subsequently used by AGV vehicles to define which of the products ready for loading has constraints that are more stringent than others. The assignment algorithm checks the vehicles position on the circuit and decides in relation to their status, which can satisfy the current mission.

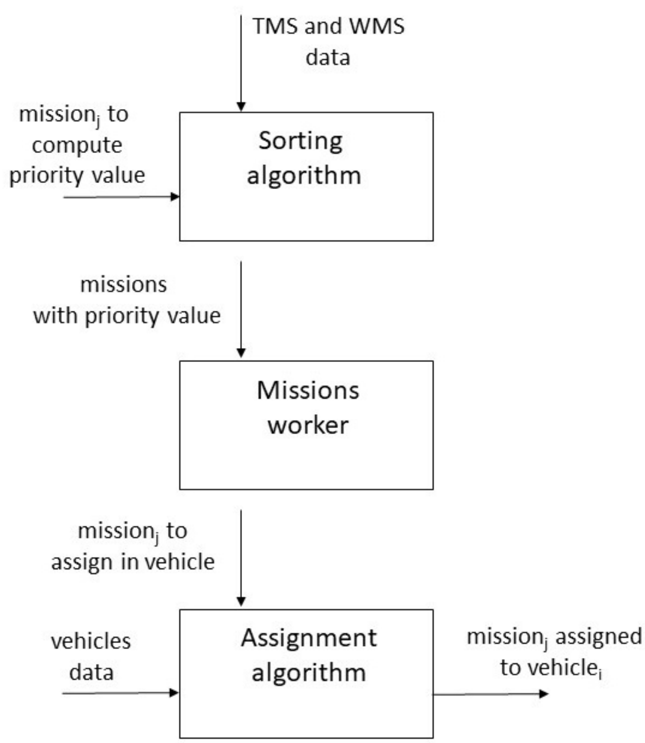

Figure 3.2: Mission scheduler

Returning to the terminology treated in paragraph 3.2, the nomenclature adopted for this scheduler corresponds to missions for the identification of tasks and to vehicles for resources. The initial input for the sorting products algorithm is $mission_j$ and the final output from assignment algorithm is given by the pair ($mission_j$, $vehicle_i$).
The algorithms shown are implemented and processed in the warehouse control system WCS, the output are forwarded to the central calculator on site and, in conclusion, sent as commands to the individual vehicles.

3.5.1 Mission data type

The mission entity, presented in the image, identifies a specific request for movement within the warehouse. When the movement is completed, whether correctly or not, the mission takes a completed status and is kept into the database for prospective company performance studies. The mission and the product item, guarantee the correct sequence of operations to be performed. The fields that require greater analysis are:

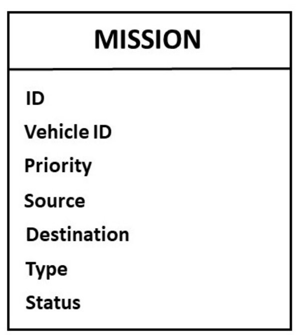

Figure 3.3: Mission data type

- *Mission type*: makes it possible to distinguish between movements due to sales or entry of products and movements due to obstructions along the route. The values it can assume are *job* or *re-positioning*;
- *Mission status*: necessary to identify the current status of the mission in the system; this field is of fundamental importance for following the expected handling flow. The sequence shown, except for step 10, is the one intended when the magazine is in automatic mode and no error

occurs during each operation.
The values that it can take are:

1. *Undefined*: when the mission is generated but its respective priority has not yet been calculated via the sorting algorithm;
2. *Computed*: the priority value has been calculated for the current mission, but all the missions with the same destination do not yet have an associated value;
3. *Ready*: the mission is waiting to be the next mission to execute;
4. *To be assigned in vehicle*: the current mission has priority over the others and the mission scheduler calls on the assignment algorithm to identify the best available vehicle;
5. *Moving to source*: the mission is assigned in vehicle and the physical movement is in progress, the empty vehicle is moving towards the source of the product, which can be a plant entry point or a location within the warehouse;
6. *Load product*: the loading cycle of the product onto the vehicle is active. This procedure is described in section 2.5.
7. *Moving to destination*: the loaded vehicle is moving towards the mission destination, whether a bay for loading onto truck or an internal location;
8. *Unload product*: the last movement that the vehicles carry out before completing the task, corresponds to the unloading of the product at the destination. Also this procedure is described in section 2.5.
9. *Done*: all the operations have been successfully completed and the mission scheduler can undertake the next mission;
10. *Error*: shutdown due to anomaly.

All the missions contain the values for the source and destination fields, as they are inserted during the generation of the mission, whereas vehicle ID is added once the assignment algorithm has identified a vehicle capable of carrying out the mission. The mission entity is essential for simple and controlled management of the operations achieved by the automated warehouse.

3.5.2 Sorting products algorithm

The algorithm takes into account one mission at a time and, in relation to those with the same destination station and with a previously defined priority value, it performs the calculation according to the TMS rules expressed in the previous section.
Assuming that all the important data from the other systems has been correctly received, it is possible to determine the priority of the mission at hand. The default priority value is indicated with X, meanwhile Y indicates the increase or decrease index as regards the priority of the mission to be examined. The algorithm is initially presented via a tree diagram and subsequently via pseudo-code. The leaf node expresses the value to be applied to the priority attribute.

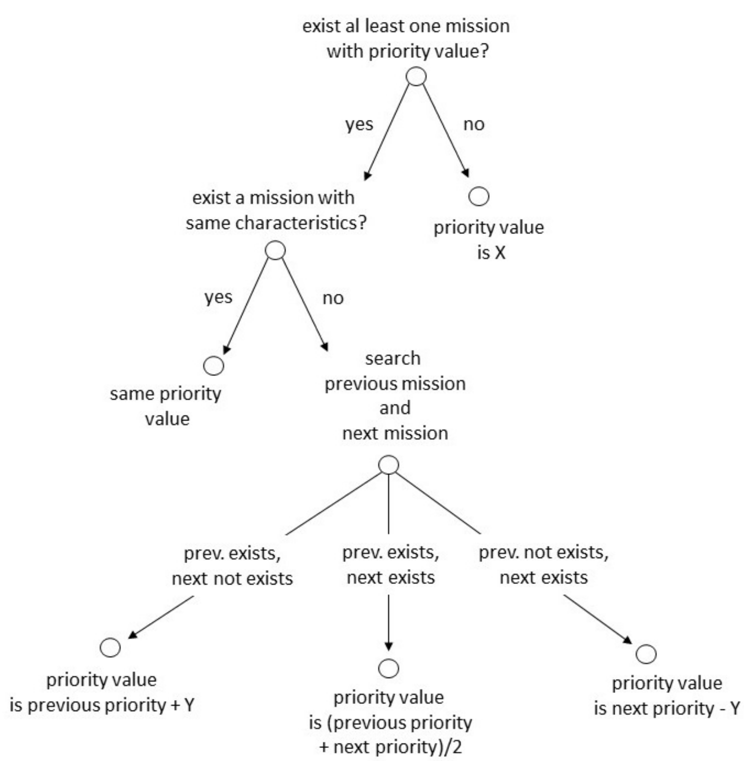

Figure 3.4: Sorting algorithm

The assigned mission priority field corresponds to the weight property w_j

analysed in the section concerning the terminology adopted in scheduler. The X and Y parameters can assume any value as long as it is different from zero when the sorting of sold products is required; indeed, if Y = 0 all the missions take on the same priority.
On the other hand, in the shipment phase, Y must be sufficiently large as to accommodate the generation of missions with activation dates much greater than the current date if this is envisaged, so as to avoid priority values with high number after the comma.
The pseudo-code concerning the sorting algorithm, the application of the tree diagram of the previous figure, is shown below. It is assumed that the necessary data are available and that $mission_j$ is the input value of the algorithm so as to determine a correct output.
As appears from the code, the detection for what immediately precedes and immediately follows the mission is extended when the result gives a negative outcome, thus searching for an ever-wider set of solutions.
In the next chapter, we analyse the time frame required for the completion of all the missions, and the computational cost of each algorithm. The sorting algorithm takes relatively short time frames, considerably lower than those required during the acquisition of data, obtained via database queries and web service invocations to other systems.

Algorithm 7: Sorting products

Input : missions, $mission_j$, destination bay, data from TMS, default value X, interval Y;

Output: $mission_j$ *priority*;

missions with priority = Get all the missions where status is equal to computed and destination is destination bay;

if *missions with priority has at least one mission* **then**

  sorted missions with priority = Sort missions with priority by increasing $ACTIVATION\ DATE$, then by $TRUCK\ NUMBER$, then by $ORDER\ IN\ TRUCK$;

  same mission = Search from sorted missions with priority the first mission with $ACTIVATION\ DATE$ equals to $mission_j$ *activation date*, $TRUCK\ NUMBER$ equals to $mission_j$ *truck number*, $ORDER\ IN\ TRUCK$ equals to $mission_j$ *order in truck*;

  if *same mission exists* **then**

    $mission_j$ *priority* = same mission priority;

  else

    previous mission = Search from sorted missions with priority the last mission with mission priority;

    next mission = Search from sorted missions with priority the first mission with mission priority;

    if *previous mission exists && next mission not exists* **then**

      $mission_j$ *priority* = previous mission priority + Y;

    else if *previous mission not exists && next mission exists* **then**

      $mission_j$ *priority* = next mission priority - Y;

    else if *previous mission exists && next mission exists* **then**

      $mission_j$ *priority* = (previous mission priority + next mission priority)/2;

else

  $mission_j$ *priority* = X;

Algorithm 8: Search from list the last mission with priority value

Input : missions, $mission_j$;
Output: previous mission;

1 Search from missions the last mission with
$ACTIVATION\ DATE$ equals to $mission_j$ *activation date*,
$TRUCK\ NUMBER$ equals to $mission_j$ *truck number*,
$ORDER\,IN\,TRUCK < mission_j$ *order in truck*;

2 If it doesn't exist then
Search from missions the last mission with
$ACTIVATION\ DATE$ equals to $mission_j$ *activation date*,
$TRUCK\ NUMBER < mission_j$ *truck number*;

3 If it doesn't exist then
Search from missions the last mission with
$ACTIVATION\ DATE < mission_j$ *activation date*;

Algorithm 9: Search from list the first mission with priority value

Input : missions, $mission_j$;
Output: next mission;

1 Search from missions the first mission with
$ACTIVATION\ DATE$ equals to $mission_j$ *activation date*,
$TRUCK\ NUMBER$ equals to $mission_j$ *truck number*,
$ORDER\,IN\,TRUCK > mission_j$ *order in truck*;

2 If it doesn't exist then
Search from missions the first mission with
$ACTIVATION\ DATE$ equals to $mission_j$ *activation date*,
$TRUCK\ NUMBER > mission_j$ *truck number*;

3 If it doesn't exist then
Search from missions the first mission with
$ACTIVATION\ DATE > mission_j$ *activation date*;

3.5.3 Mission worker

The Mission worker is the entity that keeps track of the mission status as regards the destination point, distinguishing the missions as past, in progress and future. Following the sorting algorithm, it takes from the database all the missions ready for execution, subdivides them according to the destination station, checks the last one executed and proceeds with the next one. This algorithm has the task of determining to which mission, among those awaiting assignment, it will first assign the resource once it becomes available. An example is the case in which there are three missions ready for execution, each with a different destination but respectively having maximum priority within their set, and only two currently available vehicles.
The mission worker algorithm must choose which of the three missions assign to the two vehicles, using criteria such as the type of operation - sales require precedence with respect to entry flow to the warehouse - and truck waiting time. This solution is based on the well-known *Earliest Release Date first* policy in relation to the arrival of the transport vehicle in the warehouse. This rule attempts to equalize the waiting times of the trucks to minimize the variance of the waiting times. The respective objective function is:

$$Var(\sum(C_{\mathrm{j}} - r_{\mathrm{j}})/n) \tag{3.1}$$

The mission worker follows the List scheduling structure discussed in section 3.3, with some variations due to the application context and the presence of parallel machines.
Missions to be executed is the matrix that contains all the missions to be carried out and in progress, subdivided by destination and sorted by mission priority, columns and rows respectively. At each conclusion the worker updates the matrix, always keeping the next missions to be scheduled in first position. The current scheduler does not adopt a list to keep track of operations but a matrix; in this regard, it is called a *Matrix scheduling.*

Algorithm 10: Search next mission to be executed

Input : missions, destination bays, data from TMS;
Output: next mission;

1 **foreach** *destination bay* **do**
    mission in same destination = Get missions where status is between ready and done and destination is equal to destination bay;
    sorted missions in same destination = Sort mission in same destination by increasing mission priority;
    missions to be executed = Insert sorted missions in same destination;

2 next missions to be executed = Get first mission from each column in missions to be executed where status is ready;

3 sorted next missions to be executed = Sort next missions to be executed by increasing operation type and truck arrival time;

4 next mission to be executed = Get first mission from sorted next missions to be executed;

When the missions are extracted from the matrix, those whose status is in progress are not taken into account. The final result becomes the input for the next algorithm, indeed *next mission to be executed* is the $mission_j$ indicated as input for the assignment algorithm. The assignment of the mission happens via the assignment algorithm which has the task of identifying the best AGV vehicle capable of satisfying the request.
The transition to the next mission takes place only after upon completion of the previous one, the vehicle alerts the supervision system that the requested operation has been completed and that it is waiting for a new movement to carry out. The Mission worker simultaneously updates the mission pointer and searches for the next one in the matrix.

3.5.4 Assignment algorithm

Once the calculation and identification phase of the mission is completed, it is necessary to assign the movement to the vehicle in order to complete the operation. The assignment algorithm determines, based on the euclidean distance from the mission's source point and other parameters shown below, which is the most suitable vehicle for processing the task. Once it has been identified, the mission is assigned, and the status of the Mission worker is updated in mission moving to source point.
The current algorithm is affected by two main constraints of scheduling: *blocking* and *precedence*, both analysed in the section 3.2. Indeed, the assignment is blocked if no vehicle is available for the execution of the selected mission. Whereas precedence imposes the constraint that the mission must be concluded before the assignment of the next one and the updating of the matrix.
The search ends only when the vehicle is identified. In the event that there are none, it is irrational to continue with other missions of different categories since there are no vehicles available. The completion of an operation is awaited, before moving on to the assignment of the next mission.
The *reset current mission in vehicle* function removes the previous mission assigned to the vehicle in order to give it the new one. The operations that can be cancelled concern only the movements due to obstruction of the route; the vehicle in the previous state was idle and was positioned in front of a loading station according to the *Assign strategic missions* algorithm studied in chapter 2. Subsequently a second vehicle, needing to pass through the same section, due to the obstruction caused by the previous vehicle, has re-

quested its displacement via *re-positioning*. All movements of this type can be cancelled because they do not involve real taks for the warehouse.

Algorithm 11: Assign mission to vehicle

Input : $mission_j$, vehicles;
Output: pair($mission_j$,$vehicle_i$);

1 $vehicle_i$ = Get closest assignable vehicle with position equals to $mission_j$ source position;
2 **if** $vehicle_i$ *exists* **then**
    if $vehicle_i$ *has already a mission* && *mission is re-positioning request* **then**
        reset current mission in $vehicle_i$;
    assign $mission_j$ to $vehicle_i$;

The algorithm output is given by the association $mission_j$ with $vehicle_i$. In the next chapter we will look at the calculation times required in different vehicle availability conditions.

Get closest vehicle algorithm

The *Get closest assignable vehicle* function searches among all the vehicles in the system for those who are currently not carrying out other movements and who are able to stop in front of the mission source position. It excludes vehicles that are undergoing maintenance, those that have other job type missions in progress and those which have a load on board, whether physical or logical.
Out of those remaining, it determines the interval between current position and mission source point. The vehicle to which the mission is assigned is that which has the minimum euclidean distance, and which is able to interrupt its journey in the established location. An example is an idle vehicle moving at maximum speed near to the mission origin. This may be the vehicle with minimum distance, but it will not be able to stop with the correct deceleration ramp defined by the UARM and analysed in the previous chapter. According to these considerations, the Get closest assignable vehicle function does not consider this vehicle to be a valid solution and searches for the next

Algorithm 12: Get closest assignable vehicle

Input : *$mission_j$* source position, vehicles;
Output: *$vehicle_i$*;

minimum distance = X;
foreach *vehicle* **do**
    $vehicle_i$ = Get vehicle availability where position is equal to *$mission_j$* source position;
    if *$vehicle_i$ is not available* **then**
        continue;
    actual distance = Get distance from *$mission_j$* source position and *$vehicle_i$*;
    if *actual distance ¡ minimum distance* **then**
        minimum distance = actual distance;
        vehicle with minimum distance = *$vehicle_i$*;
return vehicle with minimum distance;

one.
The X value initially assigned to minimum distance represents a very high value; a realistic example is the maximum distance between vehicle and station. In the actual project it corresponds to the length of the circuit.
The *Get vehicle availability* function is the one which checks that the vehicle is not carrying out other movements, that it is not under maintenance and that the stopping time in the established location follows the deceleration gradient envisaged by uniformly accelerated rectilinear motion.

3.5.5 Mission status updater

The *Mission status updater* function updates the mission status in relation to the conclusion of the operation. The evolution of this field can be considered as a finite state automaton which makes it possible to describe mission behaviour in a formal way. In what follows, only the steps concerning the Mission scheduler are analysed, from the value *undefined* up to *to be assigned in vehicle*. The update of the last 4 values comes from the reception of the operation completion data, received by the vehicle and by the central calculator that manages the whole fleet of the system.

Algorithm 13: Mission status updater

```
Input  : missions;
Output: mission_j status;
1 foreach mission do
    switch status do
        case undefined do
            invoke the sorting products algorithm;
            if mission_j priority has a value then
                mission_j status = computed;
        case computed do
            if all missions with same destination
             have the priority value then
                mission_j status = ready;
        case ready do
            invoke the mission worker;
            if mission_j is the next mission to be executed then
                mission_j status = to be assigned in vehicle;
        case to be assigned in vehicle do
            invoke the assignment algorithm;
            if mission_j has a vehicle assigned then
                mission_j status = moving to source;
```

Chapter 4

Results

Results about the mission scheduler algorithms are analysed and commented on below. In particular, we wish to determine computation times under different handling conditions. In the analysis, we will ignore considerations regarding the device hardware on which the program runs. We fully acknowledge that computation times depend directly on the underlying hardware, and that more powerful equipment will yield shorter times. The output, however, remain constant as a whole, respecting with appropriate proportions the results and the graphs shown.

4.1 Sorting products algorithm

The next table shows the calculation times, expressed in ms, for the sorting algorithm. The content shall be interpreted as follows: given the mission j as input, the algorithm uses N ms to calculate the priority value as output.

missions number	*computation time*
mission 1	656
mission 2	974
mission 3	1364
mission 4	1010
mission 5	804
...	...
mission 150	781
mission 151	619
mission 152	972
...	...
mission 298	1574
mission 299	1071
mission 300	621

Table 4.1: Sorting algorithm results table

In the sections attached, the first graph shows all calculation times of the algorithm. It is interesting to notice that, even though there are computation differences between missions, the time remains constant until the last execution and that the interval is given by: $1863 - 555 = 1308$ ms respectively for maximum and minimum value detected. Deviations arise from calls to the TMS service, which could introduce delays due to the telecommunications network. Still, delays are irrelevant relative to the total execution time, which always tends to remain within a two-second interval.
Data collected date back to Monday 25/11/2019, where a total of 300 missions were prepared. The algorithm took about five minutes to determine all priority values. It should be noted that the system is rarely required to calculate a single, 300-missions block simultaneously; usually, requests comprise 60/70 missions, corresponding to the movement necessary to load a truck.

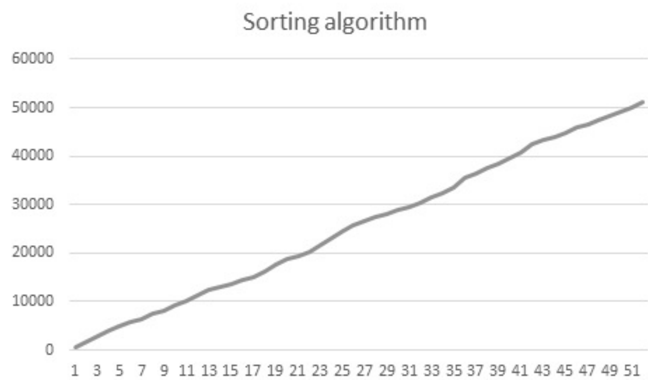

Figure 4.1: Sorting algorithm computation time

As shown in the table, the time used remains constant for all mission analysed. Therefore, the computational cost is polynomial, assumed O(n), and follows the trend shown in the graph, which represents the first 50 calculated missions.
The first mission should require the minimum calculation time since there are no priority values in the database and the search for the immediately previous and immediately next is unnecessary. Subsequent missions require a higher number of calculations and searches but, even when the search expands its range, the average time, required by the algorithm, remains unchanged; this differences are not detected, and the algorithm shows similar times under all conditions.

4.2 Mission worker

The Mission worker, among the three models presented in the scheduler, is the one which requires the minimum calculation time to determine the output, the $next\ mission\ to\ be\ performed$ value. This is so because the information it needs lies in memory, the algorithm is not subject to constraints, differently than the assignment algorithm, and there is no need to make queries or calls to external services. Among its main actions, we observe the results for the construction of the scheduling matrix and for the ordering and choice of the next mission to be performed. Results are broken down into two cases: case 1 represents a real situation, in which there are 5 parallel movements, sales or entries of products in the warehouse, and 10 missions for each movement. Case 2 includes 10 movements and 50 missions, a situation of high material flow. During the construction of the matrix, movements represent the columns and missions the rows.
The next table shows the calculation times, expressed in ms, of the mission worker. The content shall be interpreted as follows: given the case C, the worker needs N ms to create the $scheduling\ matrix$ and M ms to identify the $next\ mission\ to\ be\ performed$.

case	*matrix scheduling time*	*next mission time*
case 1	308	21
case 2	2275	34

Table 4.2: Mission worker results

Results show a calculation time that follows a polynomial trend, proportional to the size of the matrix, indeed case 1 takes 0.3 seconds to build the matrix and 0.02 seconds to choose the next mission, while case 2 takes 2.2 seconds to build the matrix and 0.03 seconds to identify the mission.

4.3 Assignment algorithm

The following tables show the calculation times, expressed in ms, of the assignment algorithm. The content shall be interpreted as follows: given the $mission_j$ as input, the algorithm uses N ms to produce the output ($mission_j$, $vehicle_i$).

mission number	*vehicle ID*	*assignment algorithm calculation time*
mission 1	vehicle 5	1656
mission 2	vehicle 4	1344
mission 3	vehicle 3	1390
mission 4	vehicle 6	1781
mission 5	vehicle 7	1656
mission 6	vehicle 1	1953
mission 7	vehicle 8	1671
mission 8	vehicle 2	1453
mission 9	vehicle 5	1391
mission 10	vehicle 4	2130

Table 4.3: Assignment algorithm case 1

Case 1: dating back to Monday 25/11/2019, involves the realisation and subsequent request for assignment of one mission at a time, with all vehicles available for the execution. With this test, we can determine the real calculation times when there is at least one vehicle available for task execution. Results show that the algorithm takes about 1.5 seconds per mission and the output is smooth for all the input missions.
The case 1 looks at regular availability of vehicles in the system and it means that the number of parallel operations is lower than the number of resources.

Case 2: dating back to Wednesday 27/11/2019, shows the activation of 10 parallel missions with 8 vehicles in the system, so the number of operations exceeds the number of available resources. Accordingly, $10 - 8 = 2$ tasks will have to wait for at least two idle vehicles before their processing time. Results indicate that the algorithm takes about 1.5 seconds from mission 1 until mission 8 to determine the output, scenario very similar to case 1. In subsequent tasks, however, the situation changes considerably.
Task 9 features a high peak due to the maximum waiting time because of

mission number	*vehicle ID*	*assignment algorithm calculation time*
mission 1	vehicle 2	1296
mission 2	vehicle 6	1312
mission 3	vehicle 4	1359
mission 4	vehicle 5	1563
mission 5	vehicle 1	1921
mission 6	vehicle 3	1784
mission 7	vehicle 8	1478
mission 8	vehicle 7	1184
mission 9	vehicle 5	308895
mission 10	vehicle 4	91156
mission 11	vehicle 2	21468
mission 12	vehicle 6	17960
mission 13	vehicle 1	22653
mission 14	vehicle 8	30301
...	...	...
mission 30	vehicle 4	37031
mission 31	vehicle 3	42906
mission 32	vehicle 8	47520
mission 33	vehicle 1	33751

Table 4.4: Assignment algorithm case 2

the vehicles have recently and simultaneously started the execution of the previous task. It follows that the algorithm fails to return an output until a vehicle completes its tasks. This happens in about five minutes; this indicates that, among the eight moving vehicles, that vehicle 4 is the first to finish and took just over five minutes for its execution.
Task 10 shows a similar profile, with a reduced calculation time due to the higher number of vehicles nearing the completion of their tasks. In fact, the algorithm takes about one minute and half to calculate the output. By adding this to the previous five minutes, we note that vehicle 2 completed the task in six and a half minutes, which is very close to the average movement time. These values arise from the *blocking* feature, analysed in paragraph 3.2, under which the algorithm operates.
In conclusion, we should note that, from task 30 on-wards, calculation times concerning the wait for an available vehicle level off at 30-45 seconds for all

tasks. This phenomenon is due to a more uniform distribution of task duration, contrary to what happened in the first 15 seconds of this case.
The final table shows the results obtained by arithmetic average of cases 1 and 2, respectively for regular and poor availability of vehicles in the system. The attachment 2 shows a graph about this scenario: the horizontal axis

vehicles availability	*computation time*
regular	1642
low	40302

Table 4.5: Assignment algorithm average

indicates the time domain, the tasks assigned to vehicles appear under the time axis and the arrows display the calculation time about each mission as regard the vehicles availability.

4.4 Attachments

Attachment 1:

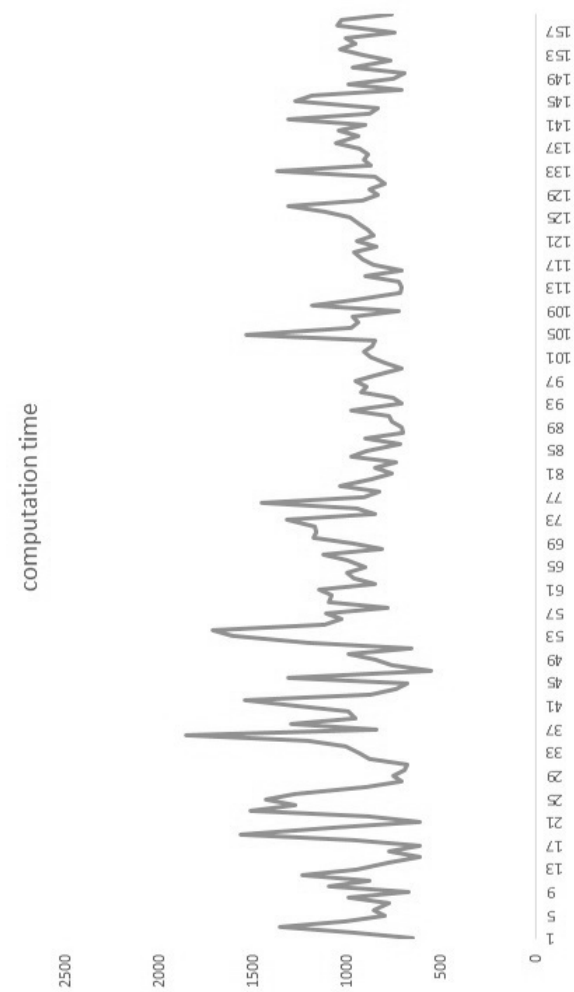

Figure 4.2: Sorting algorithm results

Attachment 2:

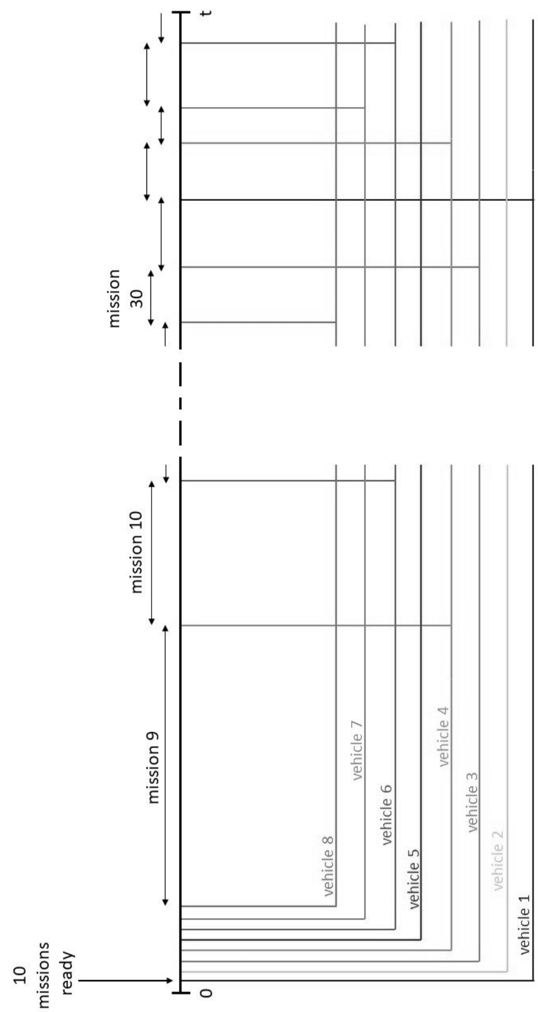

Figure 4.3: Assignment algorithm results

Conclusion

Loading or unloading a truck correctly is not just a question of safety: to reduce transport prices, the number of required journeys and pollution, the transport order preparation has recently become a real job for experts.
In this thesis project, a study was carried out on the logistics of the warehouse, and a scheduler was implemented for the efficient management of tasks assigned to RGV vehicles.
Three important criteria were taken into consideration when drawing up the mission scheduler: the first concerns the quality of the solution obtained, the second the amount of computation time to guarantee the solution and the third the time needed to develop and implement heuristic algorithms. Academic research typically evaluates only the first two aspects while, in an industrial environment, the third is considered essential. In real plants, the time required for implementing an algorithm must be restricted, and this is one of the main reasons why, in practice, heuristics with local research are more popular than more elaborate techniques.
The modules presented provide many advantages to the logistic processes. The adoption of the sorting algorithm, which follows fixed rules for the loading of products onto the vehicle, allows the driver - once he has reached the respective customer - to reduce unloading times and the number of trips required. The product ordering sequence is a procedure carried out completely automatically by the system, whereby it assigns the correct priority to the missions. The use of these new techniques reduces the operator's task just to check up and supervision. The results showed that, from the computational point of view, the algorithm follows a polynomial trend. It is able to determine the priority of 60 missions in approximately one minute.
The mission worker, the entity managing the ongoing missions, minimise the waiting times gap of the different trucks, giving priority to sales movements over entry movements, which typically occur from internal production facto-

ries. If a random solution is adopted, which does not optimise the process, the consequence is that some trucks could wait a considerable amount of time before the loading phase is completed. Its computational cost follows a polynomial pattern and is relative to the number of parallel missions that are performed. The calculation times are lower than the sorting algorithm, the scheduling matrix requires about 300 ms and overall is able to handle hundreds of missions in 50 ms.
Concluding with the assignment algorithm that identifies which, among the available vehicles, ensures greater benefits for carrying out the current mission. Different choices would involve long waiting times for the products before picking and consequently a delay in the average time of the entire warehouse. The calculated time required when there is at least one vehicle capable of satisfying the request is relatively low, between 1000 ms and 2000 ms. Otherwise, as analysed in the results chapter, there may be high peaks in waiting times in relation to the current operations running on the vehicles, with a maximum value equal to the longest movement – 1.
Personally, I am satisfied considering the scheduler and the results obtained. We have succeeded in creating a product that satisfies all the features expressed by the customer, respecting the constraints and developing a model able to identify valid solutions in reduced times. Given the regular movement flow, the mission scheduler takes a maximum of 3 seconds from the initial input point to the final output point.
Future developments include the introduction of new rules that will allow the algorithm to identify better solutions than the current ones and to enhance the dispatching rules adopted via new composite rules, for example given by the combination of different basic rules and functions that consider the waiting time in order to be executed.
An improvement would be researching a solution to avoid the precedence constraints in the assignment algorithm, which involves waiting for a completed mission before the next execution. A valid alternative could authorise the movement start but apply a check to ensure there are no previous missions pending before the stage when the vehicle deposits at the final destination.
In conclusion, a further objective is to increase the set of vehicles valid for a mission assignment. In this case, not only those who are not carrying out any operations are considered but also those who will shortly complete a mission close to the source of the next mission. This policy makes it possible to extend the set of feasible solutions, reduce time and movements with idle vehicles and improve warehouse efficiency.

The goal we are aiming for is an automatic warehouse that is effective, efficient, simple and fast and that supports the internal activities of the company and continuously improves customer service.
The development and testing of the RGV system, including the three algorithms, discussed in chapters two and three, involved the Onit automation team and me for about 8 months.
I would expressly thank the professor Silvano Martello, the professor Michele Monaci and my colleagues. The first two for the support and the time they gave me while I was working on this thesis, the latter for having involved me in this challenging project and encouraged me at all times, working with you is like being with a family. Thank you, everybody.

Bibliography

[O H97] C. Rajendran O. Holthaus. "Efficient dispatching rules for scheduling in a job shop". In: *International journal of production economics* (1997).

[E F11] A. Regattieri E. Ferrari. *Logistica integrata e flessibile*. Esculapio, 2011. ISBN: 9788874884643.

[R B12] S. Martello R. Burkard M. Dell'Amico. *Assignment Problems - Revised Reprint*. Siam, 2012. ISBN: 9781611972221.

[Mar14] Silvano Martello. *Ricerca operativa*. Esculapio, 2014. ISBN: 9788874887965.

[Mar15] Angelo Renoldi Mara Bergamaschi. *Logistica e supply chain management*. Pearson, 2015. ISBN: 9788865183663.

[16] "A new mathematical programming formulation of the Single-Picker-Routing problem". In: *European journal of operational research* (2016).

[Pin16] Michael L. Pinedo. *Scheduling - Theory, algorithms and systems.* Springer, 2016. ISBN: 9783319265780.

[17] "Designing efficient order picking systems by combining planning problems: State-of-art classification and review". In: *European journal of operational research* (2017).

[Bal19] G. Balestri. *Le basi della logistica*. Hoepli, 2019. ISBN: 9788820343354.

[D S] J. Tanchoco D. Sinriech. "The centroid projection method for locating pick-up and delivery stations in single loop AGV systems". In: *Journal of manufacturing systems* ().

[T L] M. de Koster T. Le-Anh. "A review of design and control of automated guided vehicle systems". In: *Rotterdam school of management* ().

List of Algorithms

List of Figures

List of Tables