[193031] A Distributed Approach for AGV Scheduling

Martin Ventura [a,c], Santiago Sierra Quintero [a,c], Daniel Salomon Barrios [a,c], Nelson Calderon [a,c]
Gabriel Zambrano Rey [b,c]

[a] *Undergraduate students, Industrial Engineering,*

[b] *Associate Professor, Project Director, Department of Industrial Engineering*

[c] *Pontificia Universidad Javeriana, Bogotá, Colombia*

Abstract

The implementation of Industry 4.0, where robotics mix with information and communication technologies to increase efficiency in Flexible Manufacturing Systems (FMS), is at its peak. Automated Guided Vehicles (AGVs) have become increasingly popular because they increase transportation flexibility, reducing transportation costs and overall process times. The AGV scheduling problem has been mostly pointed towards time optimization only using centralized approaches where the scheduling of production does not change and it is considered static. FMS in real life are dynamic environments that demand flexibility, as well as reactivity, to deal with changes in production conditions, such as machine breakdowns, rush orders, layout changes, lack of raw materials, among others. Therefore, there is a need for a dynamic approach to the AGV scheduling problem that addresses real life unexpected situations more efficiently, aiming for time saving at the same time. The purpose of this project is to design and implement, in a simulation environment, a distributed approach to the AGV scheduling problem that deals better with real-life FMS changing conditions. Results show that although our approach is based on the MSM heuristic, good performance measures in real time were obtained comparing with other optimization algorithms.

Key words: Industry 4.0, AGV scheduling, distributed scheduling, FMS

1. Problem statement and justification

According to recent studies by the Asociacion Nacional de Empresarios de Colombia (ANDI), there has been some meaningful increments in the awareness of Industry 4.0 in Colombia in the past years. The awareness of the new wave has increased from 43.7% in 2016 to 65.2% in 2017. Several polls indicate that 58.4% of Colombian enterprises are digitally transforming their companies, most of which come from manufacturing and service-providers. "*The society has transformed digitally and successful organizations are those that understand it and also transform*" [1]. Cloud computing, business intelligence and data analytics are the most popular digital technological tools used by Colombian companies. It is important to note that those concepts are key in automotive, health and manufacturing industries, as well as in education. Robotics, which is another digital technology, is currently at boom and the trend is unlikely to stop. As seen in *Fig. 1*, 2017 robot sales increased in a staggering 30% to 381,335 units in 2017. Robotics is playing an ever more important role in almost every industry because automation enables rapid production and delivery of customized products at competitive prices. Moreover, robots work around the clock with a consistent standard of quality and perform an increasing range of so-called 3D (dull, dirty and dangerous) tasks, improving workers' health, safety and job satisfaction. Furthermore, it is becoming easier to link robots into manufacturing production systems [2].

Regarding future investments, 49.2% of executives foresee to invest between 50,000 to 100,000 USD and 30.2% between 100,000 to 500,000 USD. Their principal objective is to automate processes, reducing overall costs, increasing productivity and reducing times. Challenges faced by enterprises, as stated by ANDI, include: lack of culture (74.1%), ignorance (61.6%), economic funds (56.3) and lack of mentality (50.9%). In addition, the current industrial revolution or Industry 4.0 conceives the integration of information and communication technologies and industrial technology in order to build cyber-physical systems [3]. The main objective of Industry 4.0 is to create a production model in which people, products and services work in synergy to assure customer satisfaction. For instance, the implementation of Industry 4.0 principles foresees a peak of productivity by 30% according to Germany's electrical industry.

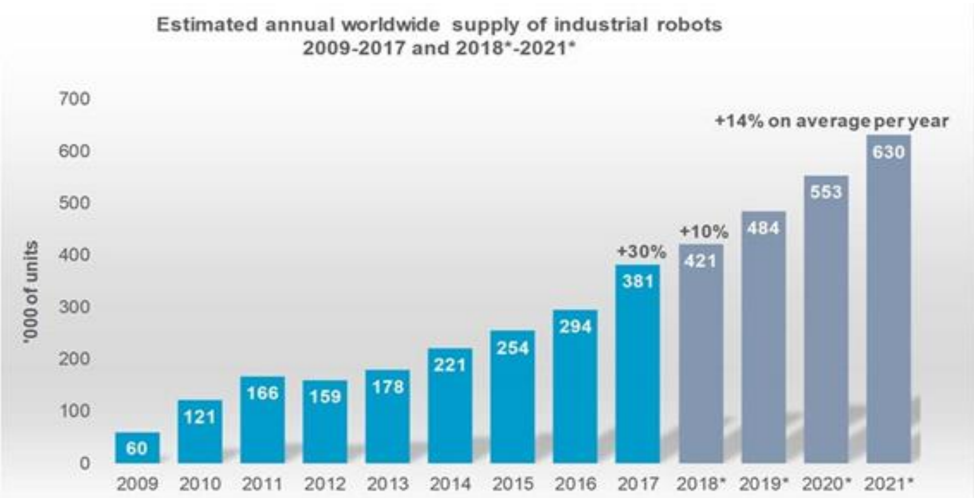

Figure 1. Estimated annual worldwide supply of industrial robots. Source: IFR

1.1. Problem statement

Technologically speaking, nowadays there are important advances that allows companies to foresee such digital transformation. For instance, airplanes are capable of flying by themselves, robots perform high-risk surgeries, and in some places, people make 98% of their transactions using an electronic device. As well, enterprises around the world have begun to use advanced technology equipment to reduce costs, get more satisfaction of clients and assure efficiency to be more competitive. Particularly, material handling constitutes up to 30-70% of the total operational costs, and it is considered a non-added value activity. Consequently, automated guided vehicles (AGVs) are becoming more popular to improve manufacturing performance, because of their flexibility [4].

AGVs (see *Fig. 2)* consist of one or more computer-controlled vehicles that run in a warehouse floor or main factory carrying several manufacturing duties, uploading and unloading manufacturing materials without the need for an onboard operator that drives the vehicle. Thus, AGV tasks include having the right material at the right place on a plant and at the right time. However, this is unlikely to happen with traditional manual material handling equipment and without proper planning of the scheduling of the material handling tasks, in coordination with manufacturing scheduling. Then, AGV scheduling is the integration of all the multiple tasks that AGVs can perform under a certain period, taking into account the shortest route, costs and overall efficiency. Moreover, AGVs usually can only carry out one task at the time and need to recharge their batteries periodically. Maximum capacity and speed are intrinsic to every AGV and can be homogeneous or heterogeneous within the group. Furthermore, AGVs can, as any manufacturing resource, break down or need maintenance, affecting the manufacturing scheduling and efficiency.

Figure 2. An AGV unloading boxes. Source: Interact Analysis "Mobile robots 2018"

Taking into account that flexibility is a necessary feature because manufacturing conditions may vary widely in the production site within a single day, AGV scheduling has to be modified and adapted several times. The purpose of the schedule is to optimize time budget for non-value adding activities, such as transportation, by minimizing or maximizing different production objectives (i.e., total tardiness and earliness). According to new revolutionary philosophies, such as lean manufacturing and industry 4.0, it is imperative to narrow process times in order to achieve Just in Time (JIT) production schemes. Then, AGV scheduling plays a key role in time budget optimization for companies, and given that technology development is going faster than ever, it is necessary to explore new scheduling paradigms different from centralized methods (i.e, linear programming) guaranteeing sufficient flexibility and reactivity.

As mentioned before, the scheduling operation can be executed with a centralized approach (*Fig 3.a*), in which there is just one controller making schedules for all AGVs. Or, a distributed approach (*Fig 3.b*) can be conceived in which there are multiple controllers, one in each AGV, responsible for making the AGV's own schedule in coordination and cooperation with other AGVs and manufacturing resources. Any FMS system using AGVs faces the problem of optimal vs reactive scheduling of AGVs in the system. For example, a move request occurs when a part finishes at a workstation. If more than one vehicle is empty, the vehicle which would service this request needs to be selected. Also, when a vehicle becomes available, and multiple move requests are queued, a decision needs to be made as to which request should be serviced by that vehicle. These schedules obey a set of constraints that reflect the temporal relationships between activities and the capacity limitations of a set of shared resources. The uncertain and ever-changing nature of the manufacturing environment makes it virtually impossible to plan all moves and possible perturbations ahead of time. Hence, AGV scheduling requires dynamic decision-making, which are dependent on the state of the system, the state of the AGV and the processing tasks going on at the workstations. The problem can be summarized in the following research question: *How to design a dynamic AGV scheduling system that has enough reactivity and flexibility to deal with real life perturbations?*

The purpose of this project is to design a distributed scheduling approach that looks for efficient transportation times, expanding the overall productivity of the manufacturing facility, in order to respond more effectively to perturbations. Taking into consideration that Cyber-Physical Systems are constituted by the physical and the virtual world (called the digital twin), the proposed approach focused on the digital twin and thus this project will be implemented in a simulation environment. One of the advantages of the digital twin is the possibility to validate various manufacturing control strategies under diverse and perturbed scenarios.

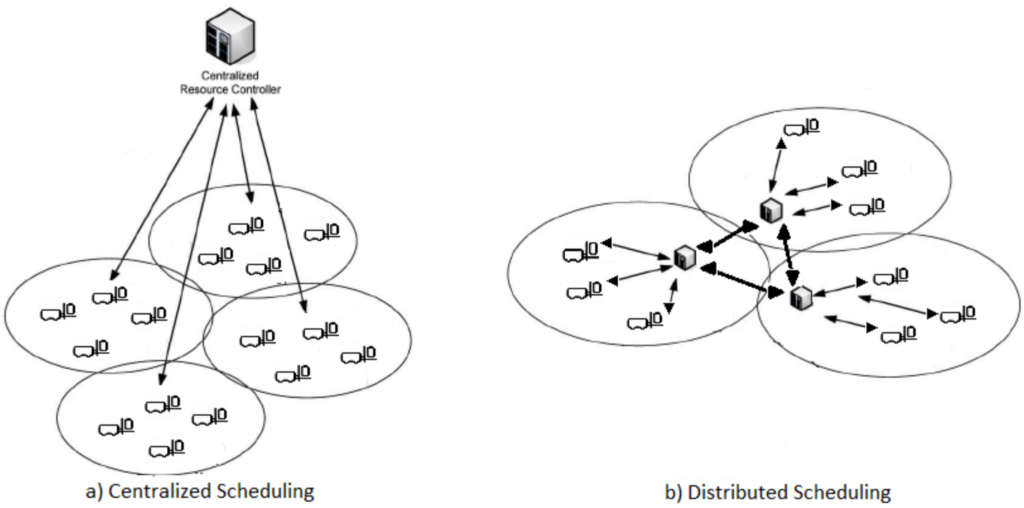

Figure 3. Comparison between centralized and distributed approaches. Source: Self-created

1.2. Justification

The AGV scheduling complexity grows exponentially when tasks increase given that it is a combinatorial NP-Hard problem. For 5 tasks and 3 AGVs there are 2520 possible solutions; for 20 tasks and 5 robots $2.58x10^{22}$ possible solutions; for 50 tasks and 10 AGVs the number of possible solutions grow to a staggering $1.62\ x10^{74}$ that is about the total number of atoms there are in the Universe! Given the computational complexity of the problem for multi-AGV scheduling, delimitations of real-life events and approximation methods must be used to get to a good solution in an accepted amount of time. Many authors have approached this problem in different ways, mostly with centralized and static methods, considering no changes in the FMS, no machine breakdowns, no flexibility, no communication within the AGVs, etc. So, the next question arises, *to what extent do these approaches reflect an ever- increasing complex and dynamic reality?* Our hypothesis is that better results can be achieved using a dynamic and distributed approach, aiming for good and instant solution. Such an approach would better imitate the constant changing reality in which conditions may alter the scheduling on a regular basis.

2. Literature review

In this section we present some relevant previous works related to static and dynamic approaches for manufacturing systems, highlighting advantages, disadvantages and methodology used in every work. Table 1 summarizes this review.

Qiu, Hsu and Wang claimed in [7] the lack of controlling software capable of managing a high quantity of AGVs is the major problem for the development of this system. Therefore, scheduling and routing are important tasks no longer trivial, which need to be developed for this technology. Finding a route involves that it should detect first if there is an existing route. On the other hand, the main objective of scheduling is to set dispatch tasks to every AGV always looking to minimize the number of them involved. Some of the main hazards when scheduling and routing are collisions, congestion and deadlocks. As the number of vehicles and jobs involved in the system increases, the complexity of scheduling also does. Another important issue that should be pointed out is the AGV technology in terms of sensory and decision-making capabilities, because great part of all the mobility and responsibility now lies on the algorithms of AGV scheduling and routing.

Static approaches of AGVs scheduling:

Rahman and Nielsen [5] address the AGV scheduling with a centralized approach consisting of two different problems: first, find an available AGV and second assign a sequence of tasks for that AGV. The authors demonstrate that as the quantity of AGVs and tasks increases so does all possible combinations to solve the problem, thus determining the multi-AGV scheduling problem as a NP-Hard problem. Thus, to overcome the complexity of the problem, the authors made the following assumptions. AGVs only take loading and unloading tasks. Also, AGVs are homogeneous (same speed, same weight capacity, etc.); materials in the warehouses are always available; AGVs don’t break down and have no maintenance costs; AGVs recharge while loading materials so they never run out of battery; new tasks cannot be assigned to an AGV that has already begun executing a task; and AGVs work 24/7. In this case the author’s objective was to minimize the total earliness and tardiness at the delivery points and they provide a time window of 30 units of time from the earliest possible start to the latest possible finish of unloading materials at each delivery point. To solve the problem, the author proposed two different meta-heuristics, a Genetic Algorithm (GA) and an Iterated Greedy (IG), to be able to compare results with each other. In both cases, the initial solution is given by a heuristic rule based on earliest due date rule. The proposed approach is considered static, meaning that information is all set before the AGVs are deployed and the solution is no revised, thus, perturbations are not taken into consideration. The statistical analysis of the results consists of a McNemar's test to compare which algorithm performed better. The results were run in 48 instances and 8 case scenarios for 20, 50, 100, 150 tasks with 10 and 20 AGVs for each method. In conclusion the IG performed better than the GA in most cases.

As well, Wenbin Gu, Zuo Li and Yuxin Li in [6], focused in the analogy between the scheduling of AGVs and the hormone regulation system. Consequently, the authors researched and made a comparison for BIA (biological intelligent algorithm) to HA (static scheduling) and MAS. The synthesis in those kinds of operations is that, when an operation task is executed the transportation task emerges. Next to it, the AGV receives the information, check the status and the machine receives information about the HSS (hormone secretion speed) of each AVG. The last step is to choose the AVG with the smallest value of HSS and it will do the task. Researchers also made an experiment where they illustrate the exceptional performance of BIA against HA and MAS. The simulation of the procedure was made by a computer for 6 different jobs. The results show that the mean total deviation of the MAS plus the HA is 19.76 for De1% and for the BIA is 4.32 in the De2% and -10.3 in the De3%. The conclusion is that BIA is not better than MAS in makespan but in the other results is noticeable that 83.33% of outcome of BIA are better than MAS.

Dynamic approaches of AGVs scheduling:

In [11] Pu and Hughes approached their investigation considering a flexible manufacturing environment, arguing that the main characteristic of this environment is that products are constantly changing, reason why the scheduling system should be dynamic and efficient. Therefore, in order to achieve different optimization goals, they proposed a modular architecture where each module contains similar heuristics. Then, they choose two heuristics and evaluate their performance separately by using both at the same time. On one side they use the computation time heuristic, consisting in scheduling the first X parts, once completed, the next X parts are scheduled. This continues until all parts have been scheduled. This heuristic has the possibility whether the user specify priorities, or a designed function does it, based on factors such as operations, start times, and due dates. The results of this heuristic showed that as the test datasets gets larger, the computation time increases. Therefore, the computation time heuristic performs better on the larger data-sets. Results showed that the longer the schedule, the less the impact of a few conflicting jobs encountered at the end of scheduling. Additionally, the authors worked in conjunction both heuristics even though they faced to trade off. The results indicate that the larger the number of parts to be scheduled, the less of a trade-off there is when using these two heuristics together. The medium and large data sets are affected less by the two heuristics used in conjunction. The authors demonstrate within this evaluation, that when multiple heuristics are used together to create a schedule, a level of control can be obtained. These heuristic modules are easy to understand and can be added or removed to achieve different optimization goals.

The interesting multi-agent manufacturing system presented in [21] is a distributed system that involves cooperative and autonomous agents, with specific roles to achieve the common goal of manufacturing products. Authors describe two types of agents, product agents that know what should be done to make a product, and equiplet agents (or production machines) that know how to perform production steps of each products. Moreover, they implemented three separate blackboards. They used first, the BB-steps blackboard so that equiplet agents announce its production steps. Second, the BB-planning blackboard in which the information of time steps of every equipment is read by production agents. Last is the BB-logfile used to build a knowledge database of the performance of every equiplet. Therefore, every agent can access information at any time at the blackboard. Additionally, they exposed how AGVs interact in the system, pointing out that the time slots of the planning system they proposed, are based on the consideration that every product will arrive by an AGV. Meaning that, travel times are considered by the blackboard system to make the scheduling plan.

The authors presenting their multi agent system in [22] based the architecture in holon agents. For this, they classified every agent in the system in two groups: physical and logical, i.e an AGV is represented by an AGV agent type (logical) and an AGV resource type (physical), they both form the AGV holon. Additionally, they describe a manager agent as in charge of initiating the system and creating jobs, throughout a global database that can be access by every agent to get information about the system. Likewise, AGVs have access to this blackboard in order to calculate the most appropriate task for them. For this task determination, AGV agent "*makes reasoning* *based on the deadheading trip time and its waiting time by considering earliest pickup time of respective operations on the blackboard".* This interaction between agents and blackboards is via messages properly built by the authors. From this paper, one of the most relevant conclusion they made was that their proposed system characterizes by its adaptability in generating schedules dynamically for newly arrived jobs.

Ghavamzadeh and Mahadevan investigated in [10] the use of hierarchical reinforcement learning (HRL) arguing that coordination skills are learned much more efficiently if agents have a hierarchical representation of the task structure. They present a hierarchical multiagent RL algorithm called Cooperative HRL, in which communication between agents is free, and also present a cooperative multi agent HRL algorithm called COM-Cooperative HRL with communication cost. The authors exemplify the hierarchy of task throughout a 5-by-5 grid in which there are 2 taxis and 4 passengers, so taxis must go to the location of a passenger, pick her up, go to her destination station, and drop her there. The overall task is decomposed into primitive actions, and non-primitive subtasks. For example, get passenger B is decomposed into two subtasks: Pick B and Navigate B, but Navigate B uses four primitive actions (North, South, East, West). As mentioned earlier, cooperative subtask are those subtasks in which coordination among agents has significant effect on the performance of the overall task, reason why they are defined at the highest level of a hierarchy. Assuming that learning is distributed, communication is free, agents are cooperative and homogeneous, and that there are four AGVs in the environment, authors compare the experimental results, and demonstrate that the cooperative HRL algorithm achieves higher throughput than the single-agent HRL and the selfish multi-agent HRL algorithms. They also conclude that the Cooperative HRL algorithm outperforms three widely used industrial heuristics for AGV scheduling (Highest Queue First, Nearest Station First, First Come First Served). As to COM-Cooperative HRL, they add a communication level in the hierarchy in which an agent decides whether to communicate with other agents to acquire their actions or do not communicate and selects its action without inquiring new information about its teammates. They conclude from experiments, that the COM-Cooperative HRL algorithm learns slower than Cooperative HRL, due to more parameters to be learned. Also, as communication cost increases, the performance of this algorithm becomes closer to the selfish multi-agent HRL algorithm, because agents learn not to communicate and to be selfish.

At last, authors conclude that the use of hierarchy speeds up learning in cooperative multi-agent domains by making it possible to learn coordination skills at the level of subtasks instead of primitive actions. Agents in the HRL method make decisions in epochs of variable length. Therefore, authors based their study in a semi-Markov decision process (SMDP), because it has become the main mathematical model underlying the HRL methods. Decision epochs are framed in the fact that when an agent completes an action, the agents whose activities have not completed are not interrupted, next decision epoch occurs only for the agents that completed their actions. With this, authors seek to design a decentralized multi-agent RL algorithm.

Title	**Reference**	**Scheduling Approach**	**Methodology**
Scheduling automated transport vehicles for material distribution systems.	Rahman and Nielsen (2018).	Static (centralized)	Genetic Algorithm (GA) and Iterated Greedy (IG)
An intelligent approach for dynamic AGV scheduling problem in the discrete manufacturing system.	Gu, W., Li, Y., Li, Z., & Qian, Y. (2019).	Static (centralized)	Biological intelligent algorithm (BIA) Multi Agent System (MAS)
Integrating AGV schedules in a scheduling system for a flexible manufacturing environment	Pu, P. Hughes, J. (1994)	Dynamic (distributed)	Computation time heuristic Schedule length heuristic
A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems.	Erol, R., Sahin, C., Baykasoglu, A., & Kaplanoglu, V. (2012).	Dynamic (distributed)	Multi Agent System (MAS)

Learning to Cooperate using Hierarchical Reinforcement Learning.	Ghavamzadeh, M. (2006).	Dynamic (distributed)	Cooperative hierarchical reinforcement learning (HRL) COM-Cooperative HRL

Table 1. Summary of literature review

In conclusion, static approaches make several assumptions to simplify the problem. On the contrary, distributed approaches are mostly implemented using multi-agent based systems using various paradigms that can be taken as reference for this project, which can also consider and use local decision-making rules inspired from centralized-static approaches.

3. Objectives

This project looks to answer the following research questions: *How to design a distributed approach to solve the automated guided vehicle (AGV) scheduling problem within a flexible manufacturing system*? *What kind of information must be considered for AGVs to execute their local decision-making processes*? *What is the relationship between AGVs and the other manufacturing resources*? Therefore, the main objective is to design and implement a distributed approach that solves the AGV scheduling within a simulated automated manufacturing system. This main objective will be achieved by accomplishing the following specific objectives:

- Explore the different distributed scheduling paradigms and choose one for the AGV scheduling problem.
- Design a distributed approach for the AGV scheduling problem based on the chosen paradigm.
- Implement the proposed approach using a manufacturing simulation software.
- Validate and evaluate, through various scenarios, the proposed approach and report results based on performance indicators for a specific case study.

4. Design statement

The purpose of this project is to design a distributed AGV scheduling approach, that looks for efficient transportation times, in order to respond more effectively to perturbations within a manufacturing environment. This approach will be implemented in a manufacturing simulation software.

4.1. Design requirements

The main requirements for this project are:

- Improving flexibility: The optimization algorithm must be able to perform adequately under various different scenarios with different parameters. Simulations must account for both theoretical and more complex real-life situations. The algorithm should in any case show reliable results.
- Scalability: The proposed distributed approach should be able to work with different fleet sizes and number of manufacturing resources.
- Manufacturing environment: The proposed approach must be validated in a manufacturing simulation software that represents a realistic FMS in terms of resources and production conditions.
- Communication: The proposed approach must allow quick and real time information to every agent in the system, enabling them to react towards latest changes.

4.2. Design constraints

The following assumptions and constraints will be made:

- This project does not focus on AGV routing (i.e, construction of paths) because it is assumed that AGVs are capable of detecting the shortest paths for each individual transportation task. Herein, a route is a set of points that the AGV must visit.

- The designed software will be a proof of concept to demonstrate that AGV distributed behavior can work effectively in a real shop floor.
- AGVs can only carry out one task at once.
- AGVs have limited battery life and need to be recharged periodically.
- AGVs can be broken or having maintenance done so their availability must be taken as a constraint.
- Simulation software used will depend on availability, license requirements and feasibility to apply the distributed method.
- Parameters concerning AGVs such as speed, loading capacity, etc. are known and considered static.
- AGVs can be heterogeneous.

4.3 Norms and standards

The methodology that will be used in this project is described by the ISO 13053-1 for the quantitative methods in process improvement. This methodology typically comprises five phases: Define, Measure, Analyze, Improve and Control ("ISO 13053-1, Quantitative methods in process improvement — Six Sigma — Part 1: DMAIC methodology," 2011).

4.4 Case Study

A possible case study is inspired on the original manufacturing system AIP-PRIMECA FMS of Valenciennes University (France), which has seven workstations, placed around a conveyor belt system with transfer gates, which employs self-propelled shuttles to transport the products along the track's blue line (see *Fig. 4*). Each product enters and leaves via the loading/unloading node located at node (m1). Three assembly workstations (m2, m3, m4) equipped with Kuka robots and an automated inspection unit m5 are placed around a transportation system. Six types of components are available (see *Fig. 5*). In the experimental case, the FMS can manufacture several products by mounting raw components to form letters (e.g., B, E, L, T, A, I and P). They are built by mounting components on a plate. Each type of job has a sequence of operations to be executed by the machines from a set of operations: O1, O2, O3, O4, O5, O6, O7 and O8. In this case study the linking paths can be replaced from a conveyor belt to AGV paths or AGV paths can be added externally. Other possible case studies will be inspired on theoretical benchmarks and the new industry 4.0 laboratory at the Faculty of Engineering.

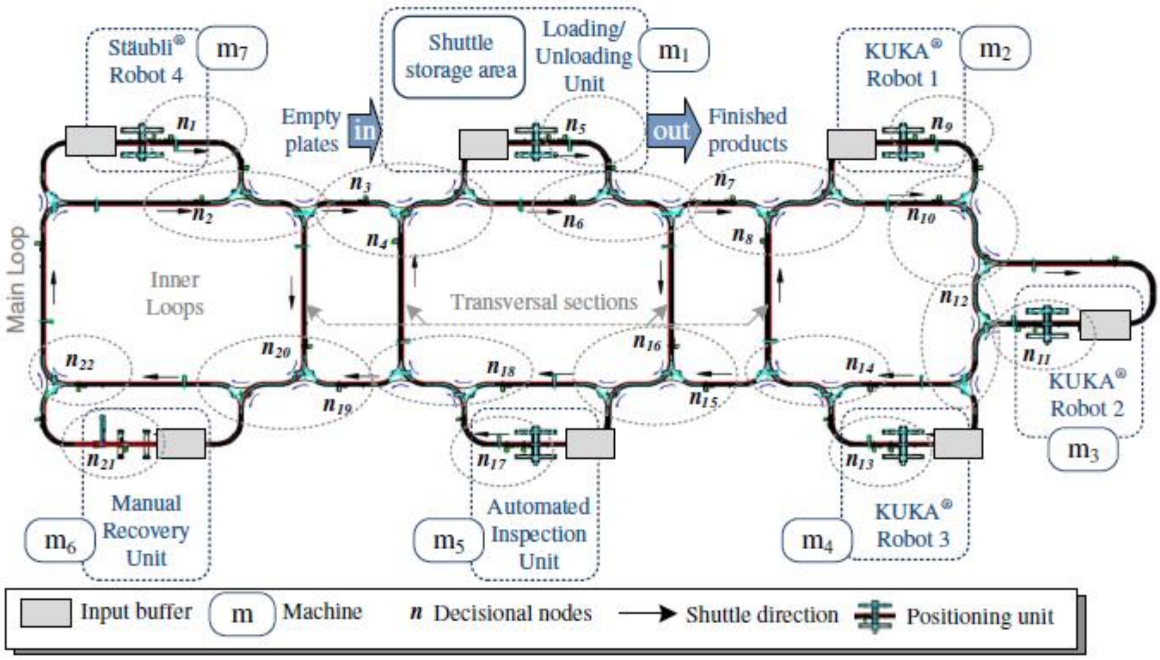

Figure 4. The AIP cell's layout. Source: Valenciennes University

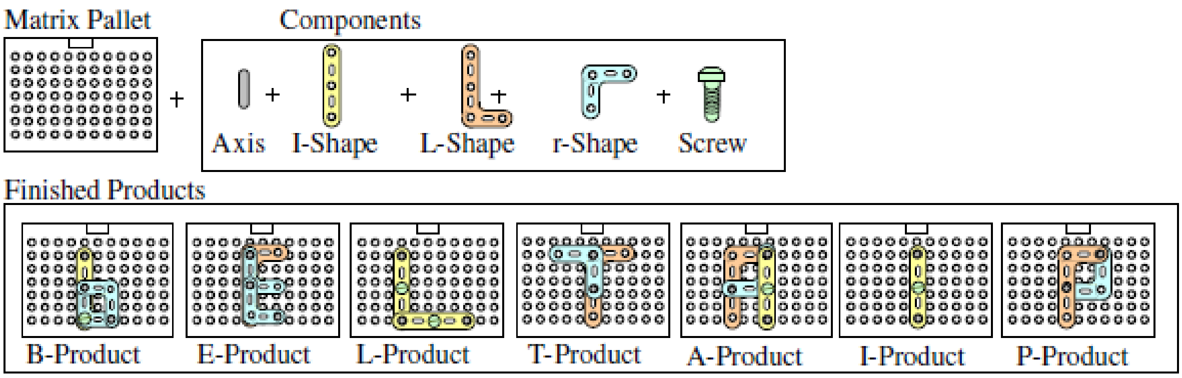

Figure 5. Product types and components. Source: Valenciennes University

5. Project methodology

In this section, the methodology applied on this project is reported based on specific objectives presented in section 3. However, results obtained from the validation procedures are reported and analyzed in the next section.

Objective 1: Explore the different distributed scheduling paradigms and choose one for the AGV scheduling problem.

Ever-changing markets push leading industries to be more adaptative to change. This scenario responds to the need of high-quality customized products at lower prices. Hence, manufacturing systems must be able to respond to changes in volume, product-mix, variety, quality and other external factors within an appropriate lapse of time. The above-mentioned can be achieved through dynamic approaches based on distributed control architectures (DCA), and decision-making delegated to more than one entity. Such architectures involve autonomous entities (agents in multi-agent systems) capable of making decisions without external direction, as a response to situations presented in their environment. Additionally, DCA enables controlling, analyzing and studying the performance of flexible manufacturing systems (FMS) along with buffer capacity, scheduling rules and sequencing flexibility as main factors. Recently, successful results have been achieved by using distributed approaches like multi-agent systems (MAS) to solve complex and dynamic manufacturing scheduling problems, such as the flexible job shop scheduling [12]. Therefore, the inherent trait of flexibility empowers the FMS to handle the ongoing market disturbances, positioning the dynamic approach as the main focus of this application, and leaving aside the traditional approaches that unfortunately do not cover the needs of the actual industry 4.0 requirements.

Henceforth, we describe four distributed scheduling paradigms seeking to decide which one fulfills the design requirements of this project, highlighting the fact that they can be implemented in any of the different multi-agent methodologies.

- **Blackboard:** The information of the task is via blackboard, a non-physical space where duties are written and read by agents, here those entities communicate indirectly and is the only way to schedule all involved jobs.
- **Contract Net:** The Contract Net Protocol (CNP) is an interaction protocol that involves two types of entities, an initiator with the goal of selling a product, and a responder that buys the product. The best bid offered by the responder, according to the aimed objective, is the one the initiator chooses.
- **Stigmergy:** In stigmergy, involved agents do not communicate with each other, they are following records from other agents constantly and using it to accomplish a group task. Using incentives is the way agents execute orders. All decisions are based on every individual experience taking a specific route, try and failure many times could determine which the most taken way is.
- **Potential fields (PF):** The PF concept is based on the attractiveness between agents in the system. The greater attractiveness emitted is the main factor for forward decisions in the manufacturing environment.

Every distributed paradigm has flexibility as an inherent characteristic to respond and adapt to different problems in the system. Meanwhile, Table 4 exposes three design requirements already declared and defined in section 4.1.

	Communication	**Scalability**	**Manufacturing environment**
Blackboard	Transport agents write/read in a blackboard where a central agent makes decisions according to the processed data. Communication is mostly in one direction for writing and only for task assignation on the opposite way, avoiding long negotiation iterations [23]	Can be implemented in big scale process. Size don't affect, moreover, a greater quantity of entities could provide more data and better solutions.	The facilities involved for this approach consist of two participants, AGV which enters information and is the task executer. The system, on the other hand, is responsible for assigning duties according to the information obtained.
Contract-NET	Direct communication between the initiator and the responder by exchanging messages. The amount of communication can be overwhelming and sometimes cause deadlocks [16].	Since the initiator sends the contract to the responder and starts to communicate directly with each agent, the number of responders in the systems affects drastically the answering task that the initiator must do.	AGVs will act as responders in this protocol, establishing direct communications with the initiator.
Stigmergy	Many individuals take different paths and then other entities take the most traveled route following incentives. The convergence rate can be larger than other paradigms.	Because of its nature, it can work with huge number of entities. Though, it demands more computational complexity and resources what makes it not scalable.	The manufacturing environment should have two different levels of application. A virtual level, where stigmergy approach makes its process in accelerated time and physical level where machines interact with each other using beacons and directional sensors.
Potential Fields	The PF emitted by the resources is how products are attracted. So, communication is throughout the attractiveness sense of products when making a decision. Communication is light because it does not require negotiation.	Many resources emitting PFs may disturb the sense of attractiveness of the product when deciding. It requires decision-making markers that are more suitable for products in conveyors rather than free-navigation AGVs.	In FMS with AGVs, products will sense PF from the nearest AGV available to accomplish the task, but resources must emit also PF to attract AGVs to the next step in the process of each product, which may complex the behavior of the system.

Table 2. Summary of dynamic paradigms

From the above and seeking to apply the best paradigm to accomplish the objectives of this work, the decision of which paradigm to use was based on the above-mentioned requirements. We conclude that blackboard protocol is the most appropriate for our model, as it maps the information of the whole system through the blackboard, simplifying communication processes between agents, meanwhile it offers real-time information by the manager agent and local decision-making for other agents. Finally, the blackboard protocol helps to easily calculate global indicators, which is of the utmost importance in a manufacturing setting, particularly when using a decentralized approach.

Objective 2: Design a distributed approach for the AGV scheduling problem based on the chosen paradigm.

To understand the proposed approach, it is important to start by describing the FMS environment, particularly the scope of the scheduling decision-making, within FMS including AGVs.

2.1 Flexible manufacturing system description

Indrayadi et al. [17] defines FMS as a system that is capable of producing a variety of products with minimal time lost in changeover from one product to the next, with two types of flexibility included in the system: routing flexibility and resource flexibility. Moreover, FMS involves two important aspects, firstly a set of physical entities each with specific tasks, i.e. resources, servers, AGVs, and secondly, a flow of information that engages different parameters such as due dates, sequencing operations for scheduling, and process planning data. The above-mentioned aspects are related to each other because the information flow initiates the physical flow of parts in the system. Hence, AGVs are responsible of making this physical flow possible between entities. As mentioned by Valmiki et al. [18], the development of AGVS for the transportation of materials between work stations brings many benefits to a manufacturing system, because it helps to control the flow of material, seeking always to have the right materials at the right place and at the right time.

Thus, in every built model of our design, the AGV role is to thoroughly transport the materials from the warehouse to its corresponding server or station, and the finished product from each individual server to the rack. However, there is a decision-making process made by every AGV, where they should decide the following, firstly, to accept or decline the appointed task by the system at a moment ***t*** based on an available status and the amount of battery units during this moment. Next, the AGV decides which material to pick, based on the priority of every material, and picking the one with the highest priority. Additionally, for all design models of this work, we assume AGVs are homogenous, meaning they have the same speed, load capacity (1 unit) and battery (measured in battery units, 100 units). Equally important to mention that, the proposed approach was design under the assumptions and restrictions detailed in section 4.2.

2.2 Proposed approach – Constructivist Methodology

In order to achieve a model that suits the complexity of FMS, a constructivist methodology was followed. This section presents two models that were conceived and implemented, in order to gradually validate results with added complexity. Additionally, exposes how agents interact in the system throughout the blackboard protocol adapted to our model.

Table 3 summarizes the main information of Model 1. The main purpose for this model was to explore and analyze how was the interaction between one AGV and other entities in the system, with a main distinctiveness reflected in static priorities for every supply. Hence, no results are presented from this first approach because of its main objective.

<table>
<tr><th colspan="2">MODEL 1</th></tr>
<tr><td>Objective:
Design a simple FMS with one AGV and two machines seeking to understand how is the interaction process between them based on the blackboard paradigm</td><td>Agents
AGV: responsible of transporting material in the system. One of its functions is transport raw material, in process and finished products. Besides, reports updated data to the system for making decisions.
System: is the agent responsible of assigning an unchanging priority to every supply.</td></tr>
</table>

Parameters:	**Subject to:**

• All AGV have homogeneous attributes as: speed, battery and load capacity.
• Quantity of AGV: 1
• Quantity of server 1:1
• Quantity of server 2:1
• Product A sequence:
  ○ Warehouse
  ○ Server 1
  ○ Server 2
  ○ Rack
• Product B sequence:
  ○ Warehouse
  ○ Server 2
  ○ Rack
	• Priorities are static and assigned by the system following LPT scheduling rule.
	• Distance is 0 in the warehouse.
	• AGVs can only transport material if the next node resource in the sequence process is available.
	• AGVs decides the task according to the priority.
	• An AGV cannot be delegated by the system if it has a not finished current task.
	• Always the AGV picks up the entity with the largest priority.
	• Product queues cannot be formed on any machine servers.

Table 3. Summary of Model 1

However, unlike model 1, priorities in model 2 are dynamic. Therefore, additional processes and decisions are held by agents in the system as updates in different stages of the production system are executed. Figure 6 exemplified a flowchart of task assignation in the system, aiming to have a first understanding of model 2 logic.

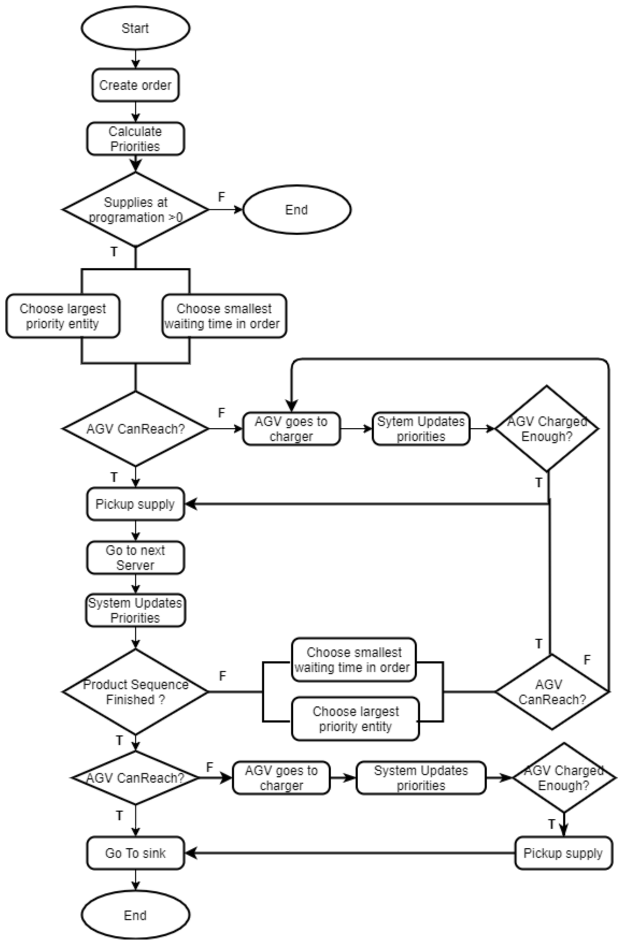

Figure 6. Flowchart of model 2

This model (see Table 4) was designed applying different scheduling heuristics seeking the best heuristic for task assignment based on the makespan of the system as the performance measure. As scheduling rules were added to the model, complexity and reactiveness increased in the manufacturing system. The following heuristics were used in the system:

- Earliest due date (EDD): Jobs are processed according to the due dates; earliest due date go first.
- EDD/Total processing time (EDDT): This heuristic refers to the division of the earliest due date and the total processing time remaining.
- Less waiting time (LWT): Jobs that have the least waiting time go first.
- Less total time (LTT): Jobs that have less work remaining (including the job to be processed) are the priority.
- Most Significant Move (MSM): The most complete of the five heuristics, as it combines in a single equation processing time, due date and transportation time, in order to select the most significant move.

  $P_{k,i} = \frac{1}{L_{k,i,j}} * e^{-\frac{S_{k,i}}{\bar{T}_{i,j}}} \quad , \forall\, i,j\, \in Ports, \forall k\, \in Tasks$ where:

$$L_{k,i,j} = Transport\ time\ of\ task\ k\ from\ port\ i\ to\ j$$
$$S_{k,i} = Slack\ time\ of\ task\ k\ at\ port\ i$$
$$\bar{T}_{i,j} = Average\ transport\ time\ of\ one\ unit\ from\ port\ i\ to\ j$$

<table>
<tr><th colspan="2">MODEL 2</th></tr>
<tr><td>Objectives:
• Apply different scheduling heuristics in a dynamic system, seeking to minimize makespan.</td><td>Agents
• AGVs: According to its position, allocation and idle status.
• System: is the agent responsible of assigning a dynamic priority to every supply, which is updated as it goes through the system.</td></tr>
<tr><td rowspan="2">Parameters:
• All AGV have homogeneous attributes as: speed, battery and load capacity.
• Quantity of AGV: 3
• Quantity of server 1:2
• Quantity of server 2:1
• Quantity of server 3:1
• Product A sequence:
○ Warehouse
○ Server 1
○ Server 2
○ Server3
○ Rack
• Product B sequence:
○ Warehouse
○ Server 2
○ Server 3
○ Rack</td><td>Subject to:
• Priorities are dynamic and assigned by the system following each heuristic algorithm
• The priority of all tasks is updated upon arrival at the warehouse, upon arrival at a server, after an AGV is loaded, and whenever an AGV enters the charger
• Tiebreaker rule used for server 1 and server 1_2 is less waiting time.
• Servers and AGVs fail randomly and are unavailable until repair is completed</td></tr>
<tr><td>Heuristics: the following heuristic were applied under the same parameters, restrictions and agents, with the same objective. An annex of the best heuristic is presented.
• Earliest due date (EDD)
• EDD/Total processing time
• Less waiting time
• Less total time
• Most significant move</td></tr>
</table>

Table 4. Summary of Model 2

As well as authors in [9] classified agents in two groups: logical and physical, we do so in our work. System agent is a logical agent in the system in charge of distributing tasks, assign priorities to supplies, and keep track of the status of each task. While AGVs are hybrid agents composed by a logical agent which communicates with the System agent, and a physical agent that transports material in the system. Besides, there are also physical stations in the system, such as programation, servers, charger dock, and sink, with different tasks. However, it should be noted that communication in the system via blackboard protocol is only given between agents. The blackboard protocol proposed for our model characterizes by the following:

- The system does not have a global database for every agent to access and obtain information about the entire system.
- Direct and indirect communication is used for the following scenario:
  The implemented blackboard method applies direct communication when the System Agent needs to give information to every AGV. Likewise, indirect communication between AGVs is used every time an AGV needs to communicate to the rest of the fleet that it already has an assigned task, to avoid a double transport bid by AGVs, this throughout the System Agent.

2.2.1 Communication between agents

Whenever a new supply arrives at Programation, the System Agent notifies throughout the "Nueva Orden" function, the arrival of new supplies to every AGV in the system. As new supplies arrived, also does production sequence information of every type of supply, so System Agent assigns a priority by the "ActualizoPrioridades" process. Once priorities are assigned to every supply, the System Agent assigns a task to one AGV. Forthwith, the AGV agents starts the logical "IrPorOrden" process, in which it decides whether to accept or decline the transportation request received. Acceptance occurs when the AGV is idle and the battery units of the AGVs are enough to make the following travel: charger station-initial node-destination node-charger station. Conversely, the AGV waits in the charger station until it is completely charged to accept the task. Upon the task acceptance, System Agent notifies the other agents that AGV "x" has already been assigned task "y". By the "Siguiente Maquina" process, the system informs the "InitialNode" and "DestinationNode" parameters to the AGV to fulfill the task. Important to mention that whenever an AGV arrives to the initial node of the task, the highest priority supply notified by the System agent is picked-up to be transported to its destination node. This, because it may happen that there are more than one supply waiting to be transported in the same initial node. When the AGV leaves the supply in its destination node, System agent updates priorities of the entire system. Promptly, AGV returns to the charger station as its permanent position when is idle, and priorities are updated again to assign new tasks to the AGV.

A summary of the processes above mentioned is illustrated in Figure 7, in which the achievement of a sequence of acts since a new supply arrives, is responsibility of an agent, i.e System agent is responsible of calculating priorities, and AGV agent of calculating and deciding whether it can reach a task or not.

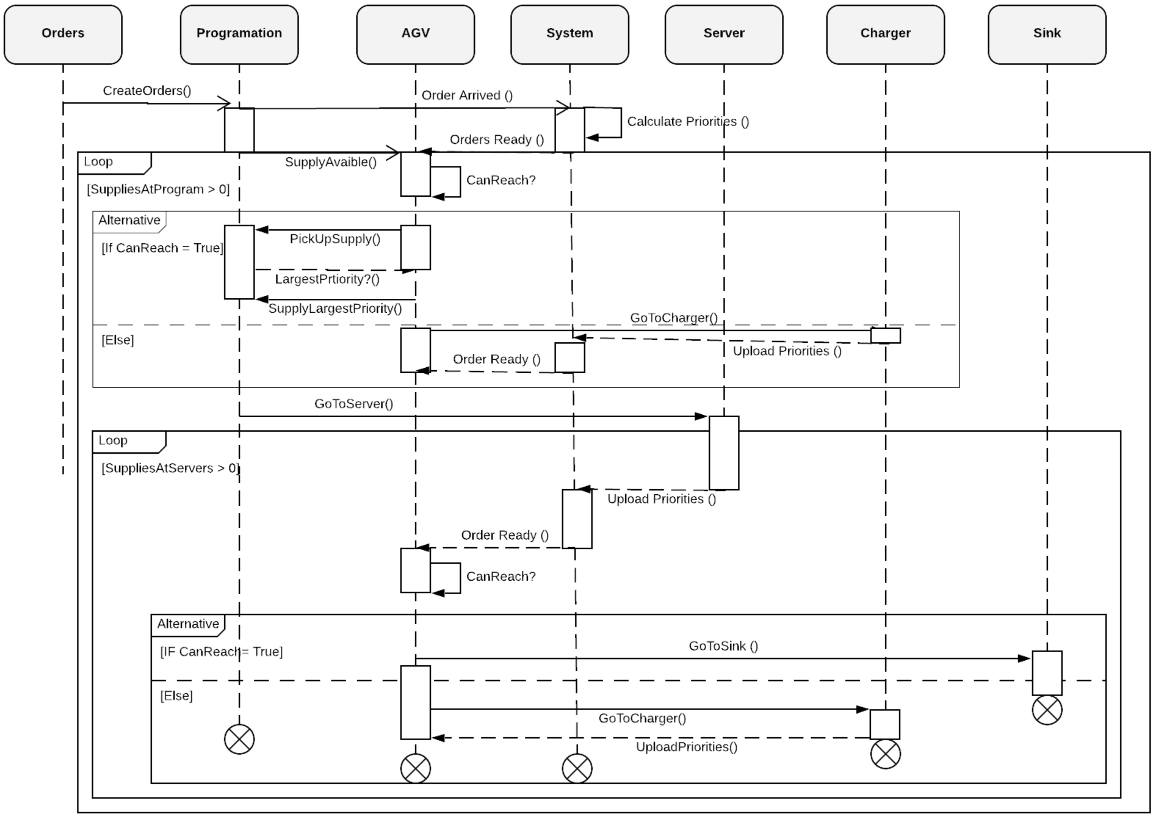

Figure 7. Sequence Diagram of the model 2

Once having the designed model, we run the simulation as shown in the next section.

Objective 3: Implement the proposed approach using a manufacturing simulation software.

3.1 Simio: Chosen Program

As described in [20] Simio is a discrete simulation modeling framework which is based on Intelligent Objects. Simio is a tool for building and executing dynamic models of systems so that the user can observe how they perform.

Although simulation and animation have been around for several years, Simio has an easy to use simulation environment in which there is no need to write programming code to create new objects as it has an object-based library, which contains the key elements to develop a certain model. One of the benefits of animating objects is the representation of the changing state of the object in a given process. An object might be a machine, robot, airplane, doctor, tank, bus, ship, or any other thing that you might encounter in your system. When an object is instantiated into a model, properties of the given object may be specified. These properties include setup, processing, and teardown times, maximum capacity, materials needed to execute the process, bills that indicate the cost of an operation, a required operator, among others. Simio is built with the latest dotnet technology that permits advanced users to customize a specific object using different languages including Visual Basic, Java, C++ among others in the model. This particular feature is crucial for simulating the exact theme of a warehouse case to be studied for AGV uses, as Simio is the only simulation software that has access to google tremble warehouse. This software allows data to be correlated among multiple tables in datasets, as well as importing external data using Excel. The most important hallmark is that Simio grants the possibility to create intelligent objects that function together in synergy in order to solve a drawback or hitch.

3.2 Agents

Simio is an advantageous program because in terms of agents in the system, it uses intelligent transport agents that communicate with the System agent directly within the program. Meaning that, intelligent agents within simio make contact with each other using Backend communication. Therefore, the events that take place within the program (which are explained immediately after figure 9) cannot be perceived by the user when executing the model simulation.

3.3 Models in Simio

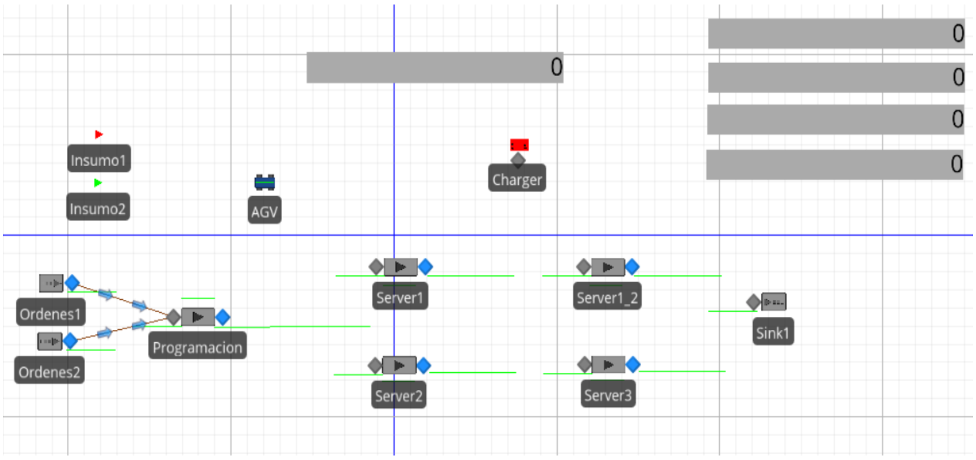

Figure 8. Final model in Simio before start

The logic behind the final model is that there is an interaction AGV-System, in which three decisions are made. The first decision consists of the system assigning the priorities for each job according to the most significant move heuristic. There are two distinct jobs, each job has its custom sequence and processing time. These dynamic priorities are updated every time the following moments take place.

1. When items arrive at “Programacion”.
2. When the AGV collects a unit and therefore is loaded.
3. When a unit arrives at any given machine (server).
4. When the AGV travels to the charging station.

The second decision is taken by the AGV when entities in the system have items in queue. Therefore, messages are sent to the AGV. The AGV analyzes if the tasks to be performed are viable according to its battery life. For example, if server1 requires collecting an item by the AGV, but the AGV can´t complete the task of moving from its current position to server1 and from server1 to the destination, then the task will be discarded until the battery requirements are met. In the opposite scenario, the AGV will have green light to collect the item, which leads to the final decision of what viable task to do. The larger priority (designated by the system) task will be executed first. The four gray bars at the top right of figure 8 indicate the total waiting time of each server as a result of the number of items in queue in each machine. The AGV takes the item to the server that has the least waiting time in order to achieve optimization.

All entities within Simio are correlated in order to successfully complete the flow of information throughout the system. Figure 9 exposes the relationship between AGVs and the supplies they carry, as well as the stations the apparatus visit. Each box illustrates the parameters that compose each entity with its corresponding type of variable (float, integer, etc). Fig. 9 shows how in a class diagram all entities presented in model 1 and 2 interact with each other, besides of including intrinsic attributes and functions necessary for a flawless performing. The purpose of this diagram is to represent both the main elements, interactions in the application, and the classes to be programmed. This class diagram shows the static structure for AGV scheduling. The top compartment of each box is the name of the class. The middle compartment represents the attributes and parameters of the class. The bottom compartment represents the operations, said class can execute.

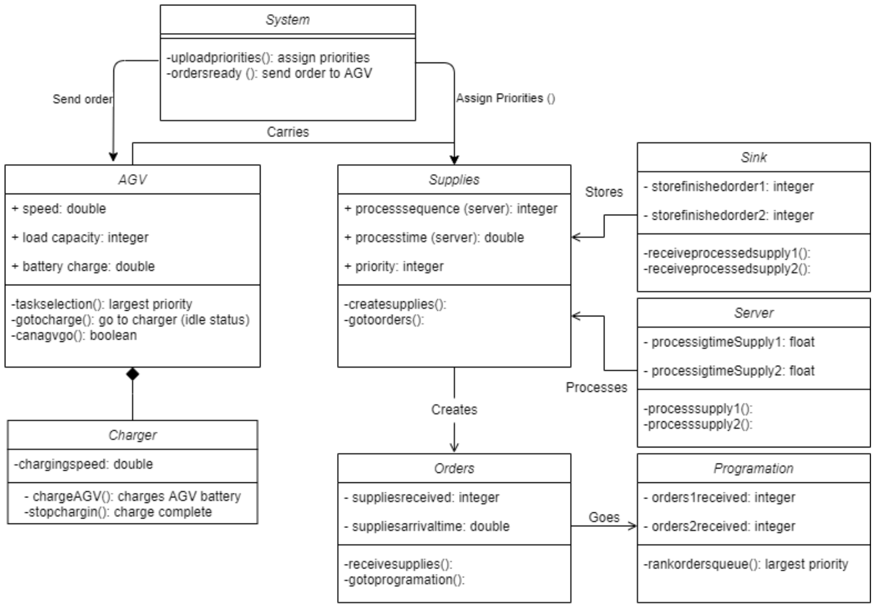

Figure 9. Class Diagram of model in Simio

Objective 4: Simulation Results and discussion

Using the New Experiment tool provided by the SIMIO software, a hundred simulation runs were made using the same parameters, only the heuristics used were changed and so different values were obtained for the target variable, makespan. The main objective of the analysis is to be able to observe in detail the performance of the current model to draw conclusions and thus be supported to forecast future applications where a greater or lesser amount of resources are used, guaranteeing scalability according to the results obtained. Likewise, this tool provides different types of results such as computational time related to intrinsic variables of the model for their analysis.

This section presents some analysis focused on the heuristic that minimized the makespan. To validate the information obtained statistically, different methods were used to assure that the chosen heuristic provides the best results. To achieve this, we elaborated an experimental design, first using an Anova, to then apply different tests such as Tukey and LSD (See annex 10) seeking to validate the significance in the mean difference between the resulting heuristic and others. Additionally, multiple charts are displayed using the results of the tool mentioned above. The results of the simulation are analyzed based on the makespan, as the performance measure of each heuristic implemented. Every result is discussed separately according to the graphic. Results are consequent when comparing with a static model (heuristic and metaheuristic) for the same facility developed in VBA (See annex 8 and 9).

To support which heuristic has the best makespan, an experimental design was made to reject or not the following initial hypothesis (H_0):

$$H_0\text{: } \mu_1 = \mu_2 = \mu_3 \ldots = \mu_k \qquad H_1\text{: Means are not all equal.}$$

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.006000753	4	0.001500188	18.6816589	4.12411E-09	2.578739
Within Groups	0.003613623	45	8.03027E-05			
Total	0.009614376	49				

Table 5. Results of the Anova

The results of the Anova shown in Table 5, evidence that $F_0 > F_{Crit}$ and Pvalue $< 0{,}05$ (α) therefore H_0 is rejected and is cautious to say at least one heuristic is significant different to others. Thus, Tukey test was applied to determine which treatment is not equal. Hence, it exists enough statistical evidence to confirm which heuristic has a shorter average time of makespan.

Results of the Tukey test in Table 6 shows numerical differences between each pair of means compared to the Tukey statistic. The absolute difference between means that are greater than the Tukey outcome, connotes that the pair is significantly different to the other sets.

Mean Yi.	Mean Yj.	|Yi.-Yj.|	Test
EDD	EDDT	0.001919	Insignificant
EDD	LTT	0.003958	Insignificant
EDD	LWT	0.005832	Insignificant
EDD	MSM	0.022915	**Significant**
EDDT	LTT	0.005878	Insignificant
EDDT	LWT	0.003913	Insignificant
EDDT	MSM	0.024835	**Significant**
LTT	LWT	0.009791	Insignificant
LTT	MSM	0.018957	**Significant**
LWT	MSM	0.028747	**Significant**

Table 6. Results of Tukey test

As shown in the test column, every time the MSM heuristic is involved in each pair of means, the test result is significant. Meaning that MSM is significantly different from the other heuristics and in general terms for this simulation, the best heuristic.

4.1 SIMIO results

- ***Computational time vs. the amount of jobs in the system***

Observing the behavior of the curve in Fig 10, it is denoted that as the number of jobs increases, the computational time increases. This is considered a combinatorial optimization problem because the order of the sequences needs to be taken into account by the system, which could result in an np-hard complexity problem, reinforcing the intention to use metaheuristics to solve the problem of scheduling in static environment at first. However, understanding that having a static environment in practical terms can only be applied in an ideal circumstance. Using a dynamic and reactive approach, although it does not assure optimality, it does offer the advantage of being applied in a real flexible manufacturing system, where the components can have failures or delays due to uncontrolled variables.

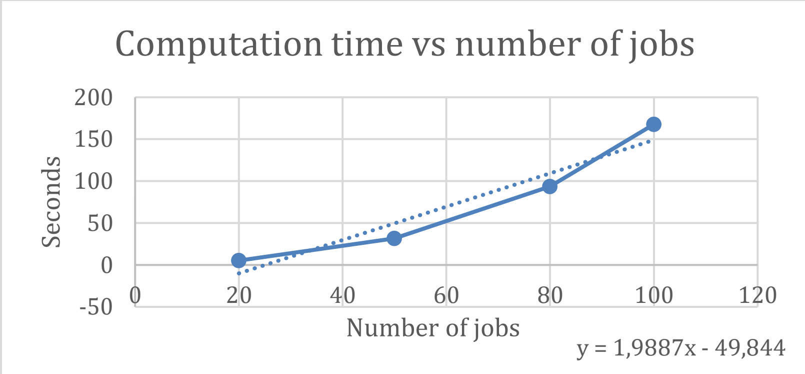

Figure 10. Graph of computation time vs number of jobs

To contrast the results obtained, a comparison of computational times was made with results obtained by Ullrich in [24]. Unlike the approach of this document, Ullrich uses genetic algorithms to generate solutions that are closer to optimal. Comparing the graphs obtained by both parties, similarities were found in the curve. However, given the fact that Ulrich´s approach is more complex, its slope in the linear equation is much greater than that obtained in this document, even doubling it. Consequently, it is evidenced that when making designs focused on the reactivity of a flexible manufacturing environment, significant reduction in computational time makes possible to handle more jobs in the whole system.

- ***Computational time vs. number of AGVs in the system***

Although the computational time and the number of jobs are correlated variables, Fig 11 proves that the computational time is not greatly affected by the number of AGVs. When contrasting computational times for 20 AGVs and 100 AGVs, we observe that the usage of computational resources increases approximately 40%, despite the number of AGVs increased 400%. Moreover, to validate this previous analysis, when comparing results with Ulrich in [24], it is denoted that both slopes of the linear equation are near zero, even though Ulrich applied a genetic algorithm to reach optimality, while this work used the MSM heuristic. This to say that results obtain in the simulation make sense when comparing with other works.

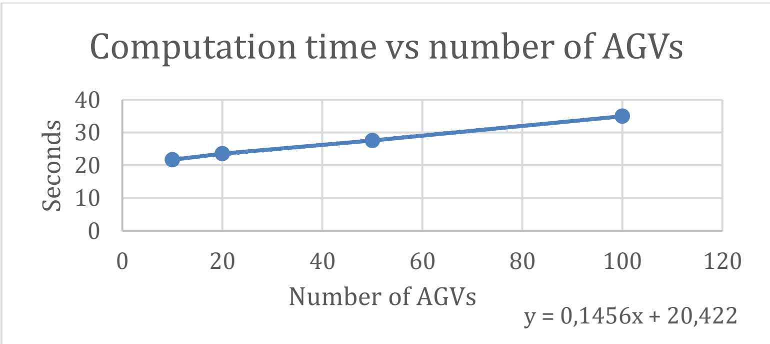

Figure 11. Graph of computation time vs number of AGVs

- ***Makespan performance based on AGV failures***

Analysis of makespan vs AGV failures was carried out comparing two scenarios in 5 situations, each with different number of tasks involved but with the same number for each type of job, and ANOVA with the situation as blocking factor was done to validate results (see annex 10). The following assumptions were considered for the experiment:

- Scenario 1 does not consider any type of failures
- Scenario 2 considers that all AGVs fail at a random exponential time with mean of 30 minutes and repair time of 15 minutes

The 5 situations were:

1. 20 tasks
2. 40 tasks
3. 60 tasks
4. 100 tasks
5. 150 tasks

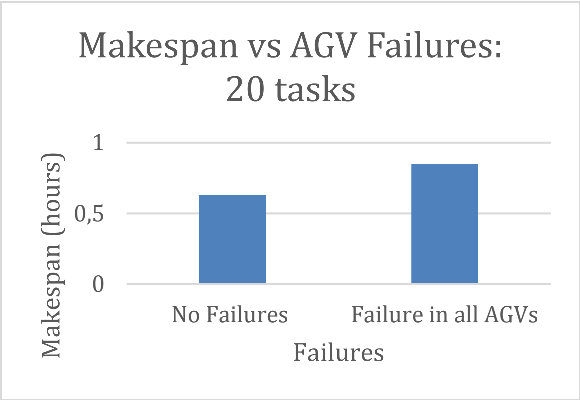

Figure 12. Graph of makespan vs AGV failures: 1

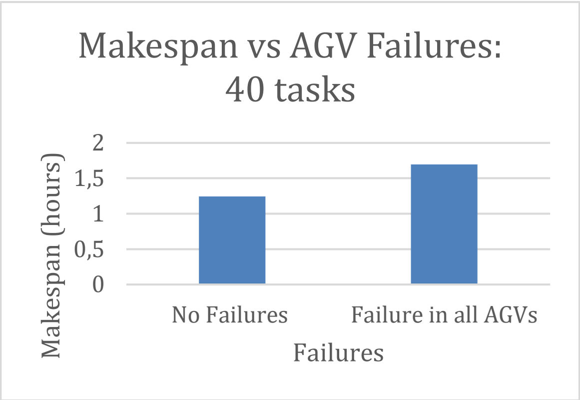

Figure 13. Graph of makespan vs AGV failures: 2

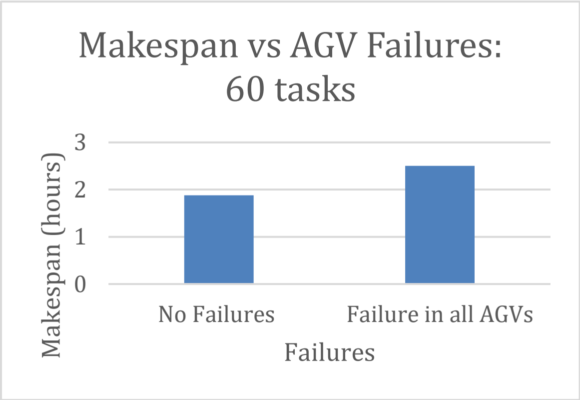

Figure 13. Graph of makespan vs AGV failures: 3

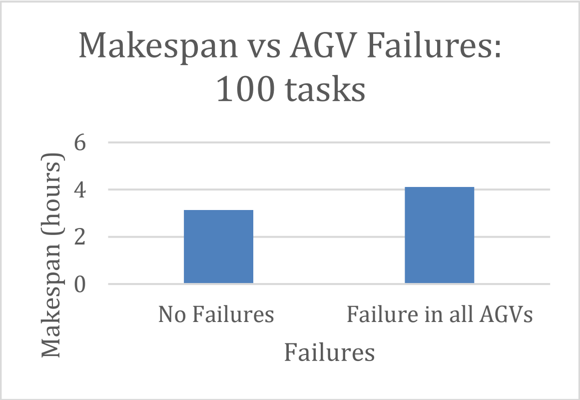

Figure 14. Graph of makespan vs AGV failures: 4

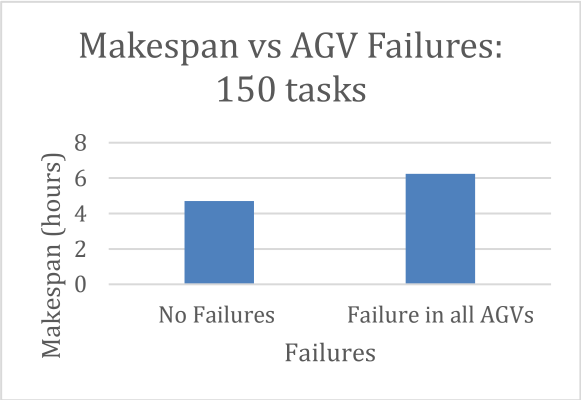

Figure 15. Graph of makespan vs AGV failures: 5

The results show that the makespan of scenario 2 increased 35% on average in comparison with the makespan of scenario 1 for all five situations. This proves that the distributed approach of AGV scheduling has great reactivity when it comes to unpredicted failure of transporting agents. As tasks are distributed among AGVs in almost real time, when one fails the others re-distribute remaining tasks very fast and with no need of human intervention, thus generating a seamless experience in the manufacture environment and saving re-scheduling process times.

- ***Makespan performance based on Servers failures***

Analysis of makespan vs machine failures displayed in Figs 13-17, was carried out comparing four scenarios in the previous 5 situations. Important to mention that the number of tasks is always equally distributed among the two job types and that the four scenarios simulated in each situation are as follows:

- Scenario 1 does not consider any type of failures
- Scenario 2 considers failures in server 1 every 30 minutes and repair time of 15 minutes
- Scenario 3 considers failures in server 3 every 30 minutes and repair time of 15 minute
- Scenario 4 considers failures in servers 1, 2 and 3 every 30 minutes and repair time of 15 minutes

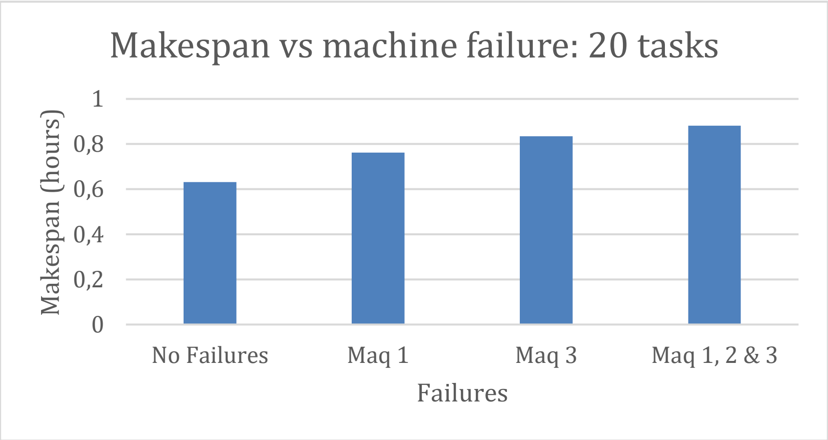

Figure 16. Graph of makespan vs machine failures: 1

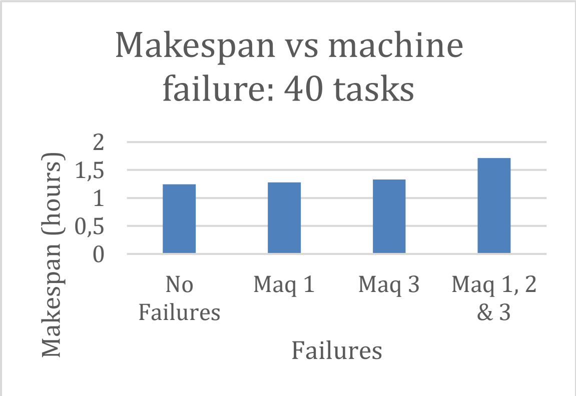

Figure 17. Graph of makespan vs machine failures: 2

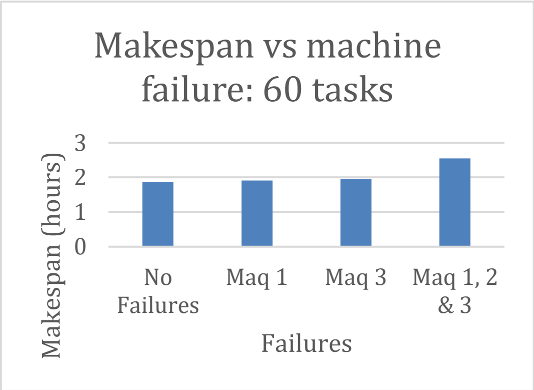

Figure 18. Graph of makespan vs machine failures: 3

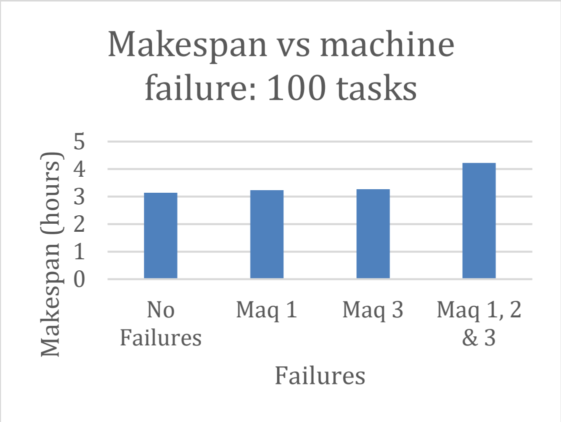

Figure 19. Graph of makespan vs machine failures: 4

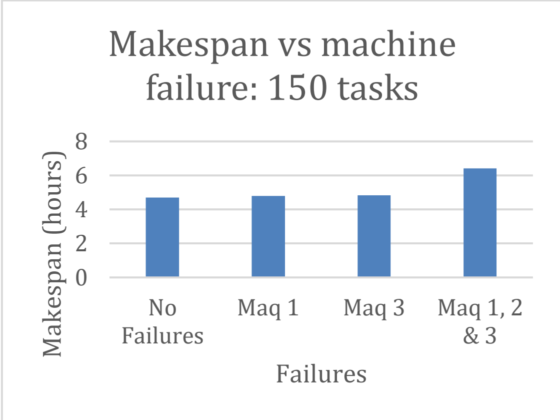

Figure 20. Graph of makespan vs machine failures: 5

The results show that the makespan for scenarios 1, 2 and 3 doesn´t change significantly (see Table 8), and that for the worst-case scenario (scenario 4) the increase in makespan compared to the first scenario is on average 37% for all five situations. This proves that the proposed distributed model has a high reactivity towards machine failure which is a major issue when it comes to real life manufacturing facilities. This reactivity is given by the distribution of tasks among AGVs which means that at any moment the robots will choose to do the task that has the highest priority at that time. Since priorities are dynamic and updated in almost real time the AGVs can adapt very quickly to changes in production conditions.

To statistically validate the significance of the experiments the following ANOVA (See table 7) was carried out with 4 treatments (Failures in the machines), 5 situations used as blocking factor and 100 observations for each factor. Considering the direct correlation between the number of jobs and the performance of the makespan in the system, the above-mentioned blocking factor was used in our experiment. Through this factor we seek to obtain an adequate and effective comparison of the simulation results against the failure scenarios, avoiding bias. The following hypothesis were considered:

H_0: $\mu_{No\ Failures} = \mu_{m1} = \mu_{m3} = \mu_{m1,2,3}$

H_a: $\mu_i \neq \mu_j$ for i ≠ j = No Failures, maq1, maq3, maq1,2,3

ANOVA						
SOURCE OF VARIATION	SS	DF	MS	Fo	Fcrit	P-value
Failures	2,263130833	3	0,754376944	8,897117927	3,490294819	0,002235686
No. of Works	50,06147265	4	12,51536816	147,6061898	3,259166727	4,29823E-10
Error	1,017466938	12	0,084788911			
Total	53,34207042	19				

Table 7. Results of the Anova for test instances

The results of the Anova displayed in Table 7 show that $F_0 > F_{Crit}$ and Pvalue $< 0{,}05\ (\alpha)$ therefore H_0 is rejected and is cautious to say at least one instance of failures is significant different to others. Thus, LSD test was applied to the treatments to determine which treatment is not equal.

Results of the LSD test in Table 8 show numerical differences between each pair of means compared to the LSD statistic. The absolute difference between means that are greater than the LSD outcome, connotes that the pair is significantly different to the other sets.

LSD FOR TREATMENTS					
Pairs		Mean Yi.	Mean Yj.	|Yi.-Yj.|	
No Failures	Maq1	2,318442	2,397648	0,079206	NO Significant
No Failures	Maq 3	2,318442	2,446016	0,127574	NO Significant
No Failures	Maq 1,2 y 3	2,318442	3,15707	0,838628	Significant
Maq 1	Maq 3	2,397648	2,446016	0,048368	NO Significant
Maq 1	Maq 1,2 y 3	2,397648	3,15707	0,759422	Significant
Maq 3	Maq 1,2 y 3	2,446016	3,15707	0,711054	Significant

Table 8. LSD for treatments

As shown in the test column, every time the instance in which maq 1,2,3 is involved in each pair of means, the test result is significant. Therefore, there is enough statistical evidence to state with 95% confidence that the makespan is affected when machines 1, 2 and 3 fail. For more detail on the statistical validations see annex 10

- ***Makespan performance based on numbers of AGVs***

In this experiment multiple scenarios with different number of AGVs were simulated to compare the different makespans, as shown in Figure 14. All scenarios were run under the same conditions (facility, heuristic rule, parameters, etc.) with only the number of agents being variable.

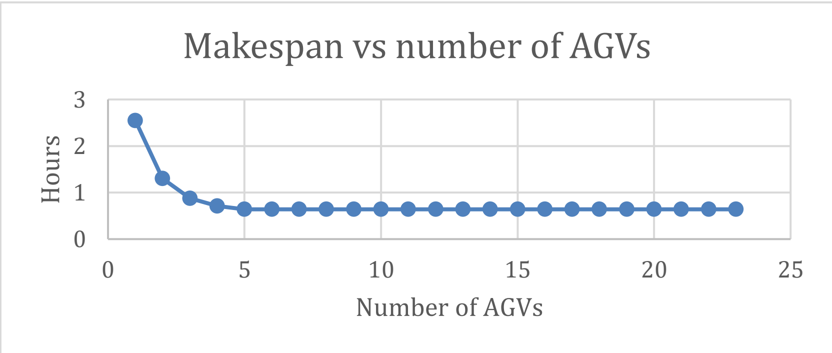

Figure 21. Graph of makespan vs number of AGVs

The results suggest that for the proposed model the optimum number of AGVs is 5. Beyond this threshold the makespan does not get any better as all extra agents are idle given the CONWIP system of the model. This proves that the chosen model is at a stable state.

6. *Conclusions:*

Industry 4.0 has overtaken the lead in the uprising of industrial evolution in order to address manufacture problems using the ultimate technology artifacts and discoveries that enhance simpler solutions at a lower budget. Among the advances the new industrial revolution is offering, Automated guided vehicles will be and are being employed in warehouse floors in the upcoming decades and years all over the most important companies around the whole world. Throughout the length of this document, a distributed approach for AGV scheduling was studied, defined, measured, executed and analyzed geared to foresee its viability and reactivity to possible changes that a real-life situation might present. The following are crucial conclusions of this project:

- Agent-based modelling is widely spreading as newer and more robust simulation software for production planning and scheduling is available, potentially improving manufacturing operations and reducing costs.
- Distributed scheduling provides great reactivity towards changing conditions in the manufacturing environment and help reduce scheduling reprocessing times.
- Agents do not require complex direct communications to coordinate task distribution. Indirect communication systems such as blackboard allow agents to effectively communicate through a third-party system that acts as moderator without taking away agent autonomy.
- Computation time is negligible as information processing is distributed among independent agents and third-party system, while it represents a big problem for static scheduling models
- Flexible job shop scheduling needs to take into account server queue time to reduce makespan and increase efficiency.
- Dynamic task priority is fundamental in order to react to real life changes in a flexible manufacturing facility.
- The number of AGVs in the system does not affects significantly the computational time of the simulation. While the number of jobs in the environment does affect significantly the usage of computational resources.

7. References

[1] ANDI, A. N. de I. de C. (2017). Encuesta de Transformación Digital 2017, 15. Retrieved from http://www.andi.com.co/Home/Pagina/19-transformacion-digital%0Ahttp://www.andi.com.co//Uploads/Encuesta Transformación Digital ANDI.pdf
[2] IFR, 2018. Executive Summary World Robotics 2018 Industrial Robots.
[3] Wang, S., Wan, J., Li, D., & Zhang, C. (2016). Implementing Smart Factory of Industrie 4.0: An Outlook. International Journal of Distributed Sensor Networks, 2016. https://doi.org/10.1155/2016/3159805.
[4] Zhou, K., Liu, T., & Zhou, L. (2016). Industry 4.0: Towards future industrial opportunities and challenges. In 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2015 (pp. 2147–2152). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/FSKD.2015.7382284
[5] Rahman, H. F., & Nielsen, I. (2019). Scheduling automated transport vehicles for material distribution systems. Applied Soft Computing Journal, 82. https://doi.org/10.1016/j.asoc.2019.105552
[6] Gu, W., Li, Y., Li, Z., & Qian, Y. (2019). An intelligent approach for dynamic AGV scheduling problem in the discrete manufacturing system. In Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2019 (pp. 1736–1739). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ITNEC.2019.8729029.
[7] Qiu, L., Hsu, W., Huang, S., Wang, H., (2010). Scheduling and routing algorithms for AGVs: A survey.
[8] Interact Analysis, 2018. Mobile robots 2018.
[9] Erol, R., Sahin, C., Baykasoglu, A., & Kaplanoglu, V. (2012). A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems. Applied Soft Computing Journal, 12(6), 1720–1732. https://doi.org/10.1016/j.asoc.2012.02.001
[10] Ghavamzadeh, M. (2006). Learning to Cooperate using Hierarchical Reinforcement Learning. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.112.2029&rep=rep1&type=pdf
[11] Pu, P., & Hughes, J. (1994). Integrating AGV schedules in a scheduling system for a flexible manufacturing environment. In Proceedings - IEEE International Conference on Robotics and Automation (pp. 3149–3154). Publ by IEEE
[12] Nouiri, M., Bekrar, A., Jemai, A. et al. An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. J Intell Manuf 29, 603–615 (2018). https://doi.org/10.1007/s10845-015-1039-3

[13] Pach, C., Berger, T., Sallez, Y., Bonte, T., Adam, E. and Trentesaux, D., (2013). Reactive And Energy-Aware Scheduling Of Flexible Manufacturing Systems Using Potential Fields. [online] https://www.elsevier.com/. Available at: <https://www.sciencedirect.com/science/article/abs/pii/S0166361513002388> [Accessed 23 May 2020].
[14] Silva J.R. (1996) A communication architecture for multi-agent applications in FMS/FAS. In: Camarinha-Matos L.M., Afsarmanesh H. (eds) Balanced Automation Systems II. BASYS 1996. IFIP Advances in Information and Communication Technology. Springer, Boston, MA
[15] Berger, T., Sallez, Y., Valli, B. et al. Semi-heterarchical Allocation and Routing Processes in FMS Control: A Stigmergic Approach. J Intell Robot Syst 58, 17–45 (2010). https://doi-org.ezproxy.javeriana.edu.co/10.1007/s10846-009-9343-9
[16] Vokřínek J., Bíba J., Hodík J., Vybíhal J., Pěchouček M. (2007) Competitive Contract Net Protocol. In: van Leeuwen J., Italiano G.F., van der Hoek W., Meinel C., Sack H., Plášil F. (eds) SOFSEM 2007: Theory and Practice of Computer Science. SOFSEM 2007. Lecture Notes in Computer Science, vol 4362. Springer, Berlin, Heidelberg
[17] Y. Indrayadi, H. P. Valckenaers and H. Van Brussel, "Dynamic multi-agent dispatching control for flexible manufacturing systems," Proceedings. 13th International Workshop on Database and Expert Systems Applications, Aix-en-Provence, France, 2002, pp. 578-582, doi: 10.1109/DEXA.2002.1045960.
[18] Valmiki, P., Reddya, A. S., Panchakarlaa, G., Kumara, K., Purohitb, R., & Suhanec , A. (2018). A Study on Simulation Methods for AGV Fleet Size Estimation in a Flexible Manufacturing System. Retrieved from https://www.sciencedirect.com/science/article/pii/S2214785317329310
[19] Carlos Ángel Iglesias Fernández (1998). DEFINICIÓN DE UNA METODOLOGÍA PARA EL DESARROLLO DE SISTEMAS MULTIAGENTE. Retrieved from http://www.upv.es/sma/teoria/agentes/tesiscif.pdf
[20] Ibrahim Amin, Muhammad Adil, Safi Ur Rehman and Ishaq Ahmad (2017). Simulation of Truck-Shovel Operations Using Simio
[21] L. v. Moergestel, E. Puik, D. Telgen and J. Meyer, "Implementation of an Agent-Based Manufacturing Grid," 2015 IEEE Twelfth International Symposium on Autonomous Decentralized Systems, Taichung, 2015, pp. 101-108, doi: 10.1109/ISADS.2015.43.
[22] Erol, Rizvan & Sahin, Cenk & Baykasoglu, Adil & Kaplanoğlu, Vahit. (2012). A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems. Applied Soft Computing. 12. 1720–1732. 10.1016/j.asoc.2012.02.001.
[23] Corkill, D., 1991. Publication - Blackboard Systems. [online] Mas.cs.umass.edu. Available at: <http://mas.cs.umass.edu/paper/218> [Accessed 3 June 2020].
[24] Ulrich, C. (2013). Integrated machine scheduling and vehicle routing with time windows. Retrieved 7 July 2020, from https://pdf.sciencedirectassets.com