Self-Organisation of Mobile Robots in Large Structure Assembly Using Multi-Agent Systems

by

Spartak Ljasenko

A doctoral thesis submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University

25th February 2019

When the war of the beasts
Brings about the world's end
The goddess descends from the sky
Wings of light and dark spread afar
She guides us to bliss, her gift everlasting

Infinite in mystery
Is the gift of the Goddess
We seek it thus, and take to the sky
Ripples form on the water's surface
The wandering soul knows no rest.

There is no hate, only joy
For you are beloved by the goddess
Hero of the dawn, Healer of worlds
Dreams of the morrow
Hath the shattered soul
Pride is lost Wings stripped away,
The end is nigh

My friend, do you fly away now?
To a world that abhors you and I?
All that awaits you is a somber morrow
No matter where the winds may blow

My friend,
Your desire is the bringer of life,
The gift of the Goddess
Even if the morrow is barren of promises
Nothing shall forestall my return

My friend, the fates are cruel
There are no dreams, no honour remains
The arrow has left the bow of the goddess
My soul, corrupted by vengeance
Hath endured torment,
To find the end of the journey

In my own salvation
And your eternal slumber
Legend shall speak
Of sacrifice at world's end
The wind sails over the water's surface
Quietly, but surely

Even if the morrow is barren of promises
Nothing shall forestall my return
To become the dew that quenches the land
To spare the sands, the seas, the skies
I offer thee this silent sacrifice

—*Genesis Rhapsodos*[1]

[1] Rhapsodos G. “Loveless,” *Crisis Core: Final Fantasy VII* [PSP]. Japan. (2007)

Publications

1) S. Ljasenko, N. Lohse, and L. Justham, “A Comparison of the Manufacturing Resilience between Fixed Automation Systems and Mobile Robots in Large Structure Assembly,” in *49th CIRP Conference on Manufacturing Systems*, 2016, vol. 0.

2) S. Ljasenko, N. Lohse, L. Justham, and I. Pereira, “A Self-Organisation Model for Mobile Robots in Large Structure Assembly using Multi-Agent Systems,” in *SOHOMA’16 Workshop*, 2016, pp. 1–8.

3) S. Ljasenko, N. Lohse, L. Justham , “Dynamic vs Dedicated Automation Systems - a Study in Large Structure Assembly” [journal, submitted]

4) Ljasenko, S., Ferreira, P., Justham, L., & Lohse, N. (2018). Decentralised vs partially centralised self-organisation model for mobile robots in large structure assembly. *Computers in Industry*. http://doi.org/10.1016/j.compind.2018.09.002

Acknowledgements

First and foremost, I would like to thank Doctor Niels Lohse for being an exceptionally professional mentor and for his tireless efforts on the path to making this thesis what it is today. The learning curve has been very rough and spiky at times, but it has certainly achieved the goal of developing me into a much better professional.

Secondly, I would like to express my gratitude to Doctor Laura Justham for taking the time to repeatedly read through the chapters of this thesis and give me constructive feedback.

Finally, sincere words of gratitude go to my academic colleagues who supported me on this journey: Doctor Karthick Dharmaraj, Doctor Angel Alberto Sanchez Salas, Doctor Eugene Benoit Florent Kalt, Doctor Ivo André Soares Pereira, Doctor Prasad Manorathna, Doctor Balaji Ilangovan, Doctor Paul Danny Anandan, Doctor Ranveer Matharu, Doctor Istvan Biro, Doctor John Raymond Hodgson and even Doctor Simon Micheler.

Abstract

Competition between manufacturers in large structure assembly (LSA) is driven by the need to improve the adaptability and versatility of their manufacturing systems. The lack of these qualities in the currently used systems is caused by the dedicated nature of their fixtures and jigs. This has led to their underutilisation and costly changeover procedures. In addition to that, modern automation systems tend to be dedicated to very specific tasks. This means that such systems are highly specialised and can reach obsolescence once there is a substantial change in production requirements. In this doctoral thesis, a dynamic system consisting of mobile robots is proposed to overcome those limitations.

As a first knowledge contribution in this doctoral thesis, it is investigated under which conditions using mobile robots instead of the traditional, fixed automation systems in LSA can be advantageous. In this context, dynamic systems are expected to be more versatile and adaptive than fixed systems. Unlike traditional, dedicated automation systems, they are not constrained to gantry rails or fixed to the floor. This results in an expanded working envelope and consequently the ability to reach more workstations. Furthermore, if a product is large enough, the manufacturer can choose how many mobile robots to deploy around it. Accordingly, it was shown that the ability to balance work rates on products and consequently meet their due times is improved.

For the second knowledge contribution, two fundamentally different decision-making models for controlling mobile agents in the complex scheduling problem are investigated. This is done to investigate ways of taking full advantage from the potential benefits of applying mobile robots. It is found that existing models from related academic literature are not suited for the given problem. Therefore, two new models had to be proposed for this purpose. It was plausible to use an agent-based approach for self-organisation. This is because similarly to agents, mobile robots can perform independently of one-another; and have limited perception and communication abilities.

Finally, through a comparison study, scenarios are identified where either model is better to use. In agreement with much of the established literature in the field, the models are shown to exhibit the common advantages and disadvantages of their respective architecture types.

Considering that the enabling technologies are nearing sufficient maturity for deploying mobile robots in LSA, it is concluded that this approach can have several advantages. Firstly, the granularity and freedom of movement enables much more control over product completion times. Secondly, the increased working envelope enables higher utilisation of manufacturing resources. In the context of LSA, this is a considerable challenge because products take a very long time to get loaded and unloaded from workstations. However, if the product flow is steady, there are rare disruptions and rare production changes, fixed automation systems have an advantage due to requiring much less time (if any) for moving and localising. Therefore, mobile systems become more preferred to fixed systems in environments where there is an increasing frequency of disruptions and changes in production requirements.

The validation of agent-based self-organisation models for mobile robots in LSA confirms the expectations based on existing literature. Also, it reveals that with relatively low amounts of spare capacity (5%) in the manufacturing systems, there is little need for sophisticated models. The value of optimised models becomes apparent when spare capacity approaches 0% (or even negative values) and there is less room for inefficiencies in scheduling.

Table of Contents

Table of Figures

Notations

C_{dj} – closeness of due times

C_j – completion time of job J_j

$C_{j,\ min}$ – earliest possible completion time of job J_j

$C_{t,\ j}$ – tardiness cost of job J_j

$C_{t,m}$ – the maximum tardiness cost for all given jobs J

d_j - due time of job J_j

d – distance between given mobile resource agent and product agent that it negotiates with

f - number of deployed resources of the dedicated system

G_b – bid gap of a PA's credit offering strategy

i – a workstation *WS*' column index

IF - interference factor for the mobile system

$J = \{J_1, J_2, \ldots., J_n\}$ - a set of *n* jobs to be processed

L_j - time required to load a job J_j to a workstation

l_j – start of loading of a job J_j to workstation

l_j' – end of loading of a job J_j to workstation

m - number of deployed mobile resources

m_j – combined capacity of a manufacturing system's resources at job J_j

M_o – maximum credit offering

$MD = \{MD_1, MD_2, \ldots., MD_f\}$ – a set of *f* dedicated resources (machines) that process jobs for the dedicated automation system

$MM = \{MM_1, MM_2, \ldots., MM_m\}$ - a set of *m* mobile resources (machines) that process jobs for the mobile automation system

$P(t)_j$ – priority value of job J_j as a function of time *t*

p_j – processing time of job J_j

PTF - sum of resource utilisations of the dedicated system

PTM - sum of resource utilisations of the mobile system

R – ratio between the sum of required loading and unloading times of a job J_j and its working content W_j

r_j - release time of job J_j

$R_p(t)_j$ – priority ranking of job J_j as a function of time *t*

R_{rj} – resources required at job J_j

S_j – starting time of a job J_j

t - time

T_j - tardiness of job J_j

$t_{l, j}$ – launch time of job J_j into the manufacturing system

u_j – start time of removal of job J_j from workstation

u_j’ – end time of removal of job J_j from workstation

U_j – removal (unloading from workstation) time requirement of a job J_j

U - utilisation of a system. $U = PTF / (t * f)$ for the dedicated system, and $U = PTM / (t * m)$ for the mobile system

w_j – priority of a job J_j

W_j – working content of job J_j. This is a measure of how much work must be processed in job J_j by mobile robots. Each working RA processes one unit of WC_j per time step

WD_j – work done on job J_j

WR_j – work rate at job J_j

$WS_{i\ldots1,2}$ – a workstation with column index *i* and a row index of 1 or 2

$\mu_j \leq M$ – a set of machines allocated to a job J_j

Glossary

Adaptability – the ability to change the system’s capabilities over time

Agility – the speed of implementing adaptations in a manufacturing system

Complexity of planning/decision-making – the measure of how much computational overhead is expected to be required for achieving a solution

Control – the act of distributing manufacturing resources to products in the production process. In this thesis, mobile robots are the resources that are controlled in a distributed manner. It is used interchangeably with **planning**, **decision-making** and **scheduling**

Flexibility – the ability to switch quickly between products and/or schedules by having short setup times and the capability to process a variety of products

Redundancy – the duplication of manufacturing equipment to enhance reliability

Resilience – the ability to mitigate or absorb the impact of a disruption and quickly return to normal operating conditions

Scalability – the ability to incorporate additional resources to workstations. In the context of this thesis, scalability is product-centred. This means that it is the ability to add machines to products so as to increase their individual work rates, as opposed to the more common meaning where machines are added to a larger part of the whole system

Versatility – the ability to process several different types of tasks

Abbreviations

ARC – ratio of available and required capacity

CNP – contract net protocol

FIPA – foundation for intelligent physical agents

HMS – holonic manufacturing systems

LSA – large structure assembly

MAS – multi-agent systems

PAP – priority aging policy

ROI – return of investment

TWT – total weighted tardiness

Chapter 1 - Introduction

Manufacturing automation technology has been rapidly advancing to meet the needs of the modern manufacturing domain. The drive to reduce production costs, improve quality and increase productivity has encouraged heavy investments from manufacturing companies. Large structure assembly (LSA) is no exception. For example, the already heavily overbooked order backlogs of the currently two largest aerospace manufacturers, Airbus [1] and Boeing [2], seem to be relentlessly growing further. The commonly used manufacturing systems in LSA are usually very large and rely on dedicated jigs and systems as well as specialised equipment and highly skilled workforce. It has been convenient to use such systems because until recently there was no frequent need to make changes in the production processes. The current trends are increasingly suggesting that this could be changing in the near future. The inherent disadvantage of such systems is that they are often not fully utilised due to the difficulty of effectively supplying them with enough work or setting up to meet new requirements. Moreover, developing a dedicated system for each process is very expensive and is increasingly less viable due to them reaching obsolescence faster than before. With the recent trends of increasing customisation, shortening time between changeovers and increasing need for flexibility, two core issues have risen: Firstly, it is challenging to increase productivity by simply upgrading production technology or reorganising production layouts. Secondly, considering the various capacity-related disruptions such as rush orders, cancellations, rework, scrap and requests for prototypes; the traditional systems are increasingly becoming less feasible. Therefore, it is reasonable that manufacturers have acknowledged the shortcomings of such systems and are heavily investing in the research and development of more agile, versatile, flexible, resilient and adaptable systems.

While manual labour is generally accepted to be very flexible, it is less economically viable for large scale production, particularly in high wage countries [3]. In addition to being less productive, it usually accounts for high losses due to absence, injuries and defects. Furthermore, in LSA, machine drilling and riveting (which account for a large proportion of working content) save several steps and issues compared to manual methods (pilot holes, removing metal chips from gaps, burrs and part deflections) [4]. Thus, pushing the limits of automation has become necessary for staying competitive in production environments.

Manufacturers see the value in solutions that are not permanently fixed to the floor nor require additional facility investments [5]. With such solutions, any change in the shop floor layout would become reversible and there would be no need for additional infrastructure. Modern automated assembly systems like the E6000 [6], E7000 [7], HAWDE [8] and GRAWDE [9] are used for drilling and filling tasks of aircraft assembly parts. The E6000 is used for drilling, countersinking and riveting holes in aircraft wing panels. The E7000 does the same processes but on fuselages. HAWDE is used for drilling holes in the wing box of Airbus A380 aircraft, whilst GRAWDE is used for the same in gear rib areas. All of the mentioned systems use gantry rails for locomotion and accurate localisation in order to carry out their tasks to a high standard. Whilst performing to an excellent standard in a static environment (rare disruptions, mass-production) they have a poor response in relation to any changes in production requirements (i.e. new product specifications, changes in schedules, fluctuations in the required throughput).

In LSA, very large, heavy and often difficult to handle products are assembled. Manufacturers of ships, trains, aircraft and other industries are often tasked with solving challenges that have not been addressed in any other industry. Some of the toughest challenges that the manufacturers are currently facing are caused by a rapidly growing demand and ever-increasing need to frequently change setups and schedules. It has been outlined that current systems face serious issues when responding to demand fluctuations, ramping up production, introducing new variants of products and experiencing disruptions on the shop floor [10]. The future needs in such environments include agile response to changes in production and disruptions; efficient and cost-effective ramp-up of production rates in response to fluctuations in demand; scalable and convertible fixtures that can support all variants of the same product type or sub-assembly [4]. Therefore, there is a clear need to smoothen the process of handling any production changes that may occur.

Much like manufacturers in other industries, manufacturers in LSA are facing an increase in demand for varieties of products as opposed to large batches of identical ones [11]. From the perspective of manufacturing tasks, this means that the currently common systems are becoming less practical for use because they tend to be designed without much consideration for being moved, reused or altered. When responding to changes in requirements, it is often necessary to install additional fixtures [12]. Such fixtures can take up to 24 months to design, manufacture and install [13]. Thus, it is clear that more adaptable and versatile solutions are necessary to maintain competitiveness in the coming years.

One solution for that is the development of reconfigurable manufacturing systems for such purposes [4], [8]–[10]. In [13], it is extensively explained how using bespoke, permanent tooling for aerospace wing assembly is costly for both: the design and manufacturing stages. When demand increases, then additional one-off tools must be produced to achieve the required work rates. Whereas, when the demand decreases, the expensive equipment is under-utilised [14]. In the light of increasing demand and decreasing time to market, the currently used systems are strained further [13].

Despite the additional initial capital investment, the proposed reconfigurable systems should quickly adapt to changing requirements without the need to halt production or investing additional capital. Until very recently, the development of such systems has been neglected mainly due to the inherent complexity, cost and infancy of enabling technologies [4]. A case study on Airbus [15] has shown that the implementation of an appropriate methodology (which targeted creating cost-effective reconfigurable cells) for such systems can ramp up their capacity from 40 to 100 aircraft per month. This shows how under-utilised their current systems are. Furthermore, Müller and Esser compiled a list of scenarios where reconfigurability and standardisation can bring additional benefits [12]:

- They can support the planning process and reduce engineering cost in the design phase because capabilities can be adjusted at later stages

- They leave the option of changing specifications at short notice, which is particularly beneficial in the ramp-up phase
- They support new variants that are commonly introduced during the life-cycles of aircraft models
- An increased demand for outsourcing of aircraft assembly tasks, prototypes and test units can be adequately addressed with reconfigurable manufacturing systems at a reasonable cost

From the perspective of job flow, it can be very challenging to supply all manufacturing machines with work to maintain high utilisation and consequently a shorter return on investment (ROI). Because it is not feasible to handle such products by means of conveyor belts [16], the most common method has been the use of crane systems [17]. Handling products with crane systems is not only slow in LSA but also challenging to avoid crane interference when using multiple cranes [18]. In particular, this can be challenging in shipbuilding where assemblies are composed of many levels of sub-assemblies [19]. Therefore, even without considering any disruptions or changes, there seem to be benefits from reducing the need to transport products between workstations.

One way of reducing the load on the crane system would be by using appropriate mobile platforms. Currently, such machines can carry payloads of up to 100 tonnes and open up the possibility of reducing or completely removing the load on crane systems. This approach would inevitably require additional space for manoeuvring on the shop floor; and appropriate planning for routing and collision-avoidance. Moreover, it would not enable product-centric scalability as well as a system of mobile robots. Thus, this approach has some potential; however, there is very little literature investigating the topic due to the enabling technology still being new.

As is discussed in the literature review (Chapter 2), some work has been done in the direction of mitigating the existing limitations. Promising approaches include extending the working envelope to accommodate additional workstations for products [17] and using modular systems with standardised parts [4], [10], [13], [15]. Whilst having a high potential for the short to medium term, these approaches still exhibit their inherent limitations and may not be feasible in the long term.

As a result of the observations above, the benchmarked fixed automation system’s shop floor layout in LSA is as represented in Figure 1-1. As shown there, the system is constrained to its own gantry rails and has a large enough working envelope to work on one workstation while a product can be loaded on the other one.

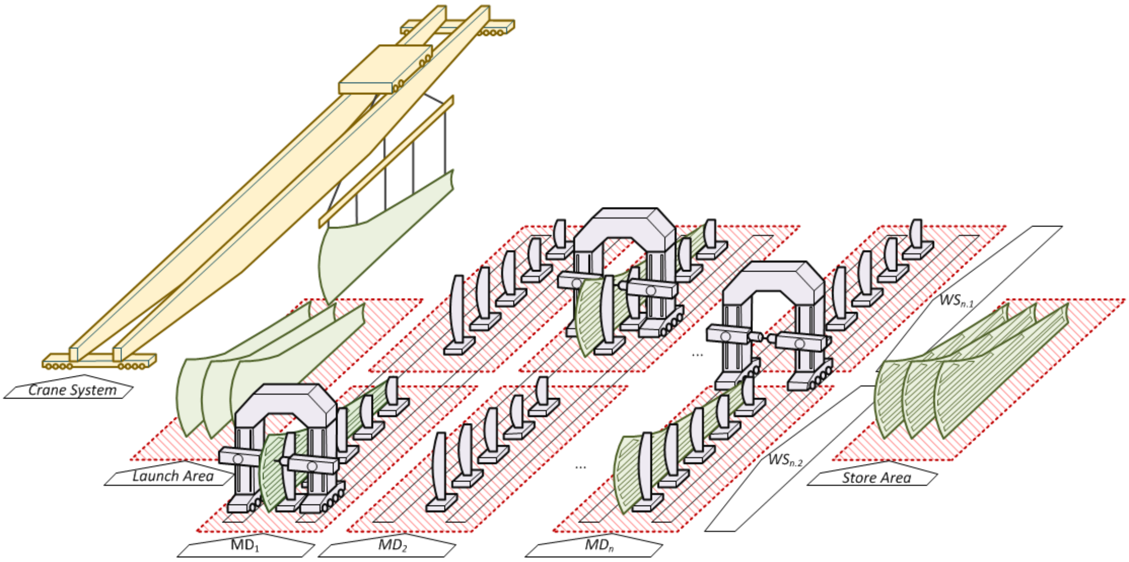


Figure 1-1: The principle shop floor layout of the fixed automation system

In this doctoral thesis, it is envisioned that mobile robots take over the mentioned assembly tasks. As opposed to moving products to manufacturing resources, with mobile robots, the manufacturing resources move to products instead. This is expected to dramatically increase machine utilisation because they will not need to wait for products to be loaded on their workstations. Instead, they can move to available products. This is particularly useful in LSA because the products often require a long time for loading and unloading from workstations. Furthermore, considering the size of the products, it is possible to vary the number of mobile robots allocated to each product to control the individual work rates and consequently the completion times of jobs. Thus, many limitations of traditional manufacturing systems in LSA could be naturally resolved this way.

A mobile robot can be thought of as a standard 6-DOF robotic arm but installed on a mobile platform that adds the ability to transport itself anywhere on a shop floor [20]. In many cases, it is simply a robotic arm with the required capabilities mounted on a mobile platform that can carry the desired payloads. Until recently, such mobile robots have mostly been used for pick-and-place jobs. Lately, some application-specific mobile robots, like [21], have been developed for tasks that demand higher accuracy and structural stiffness.

The envisioned mobile system's principle shop floor layout in LSA is represented in Figure 1-2. It has much more movement freedom than the dedicated system: any mobile robot can move to any workstation at any moment in time. Furthermore, several pairs of mobile robots can fit around one product to increase the work rate. The latter ability is called "product-centric scalability" throughout this thesis.

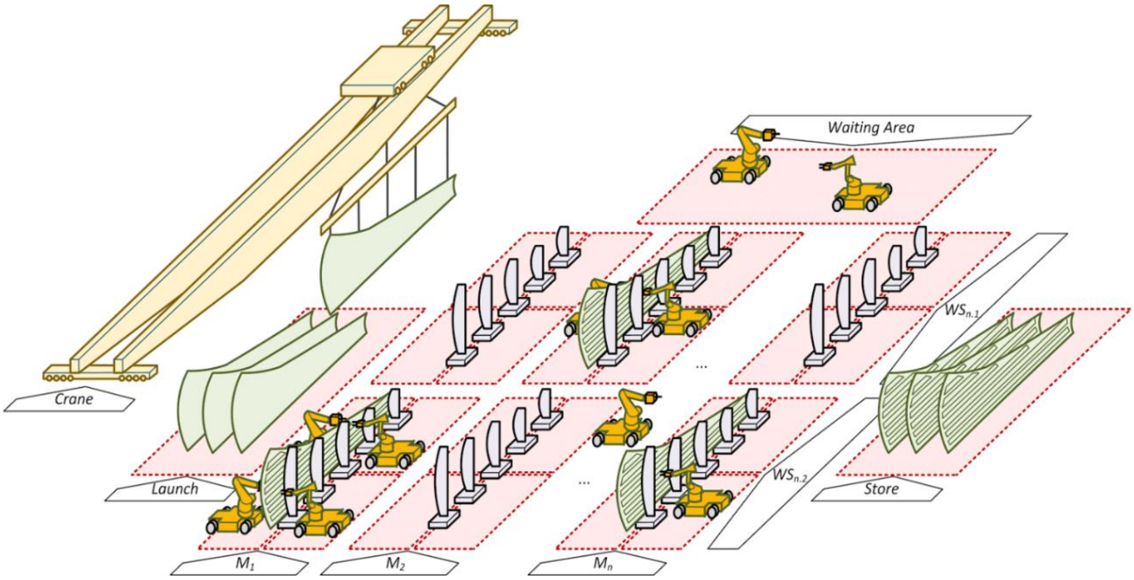


Figure 1-2: The principle shop floor layout of the mobile automation system

Further to the benefits of deploying mobile robots and executing tasks, it is important to examine the economic considerations. Owing to the novelty of the given technology with respect to the aerospace assembly tasks, it is evident that mobile robots can be too expensive to implement at this time. However, there are numerous arguments in favour of mobile robots in the long term.

Hypothetically, considering a surge in the technological uses cases for mobile robots in the future, the demand for them would increase. Mobile robots would usually be mass-produced because they would be used for many different tasks. This would lead to lower costs per unit because the engineering effort in developing them would be spread out. Also, their mass-production would mean that their components would be standardised and reconfigurable, receiving the abovementioned benefits. Scaling production rates would also become smoother to facilitate because general-purpose mobile robots would be available off-the-shelf. Or in the more specialised cases, they would have already been designed and the lead time for their production would be considerably shorter. Therefore, there is a strong economic incentive in overcoming the remaining engineering challenges and considering mobile robots for manufacturing tasks in LSA.

1.1. Aims and Objectives

In addition to determining in which situations mobile systems can be preferred to static ones, the proposed approach inevitably leads to questions with regards to decision-making or planning of the given tasks. Because the fixed automation system is responsible for very few workstations, there is little logic necessary to decide which product to process. Whereas, with mobile robots, the freedom of movement and product-centric scalability mean that there is access to any workstation even if there are already mobile robots working there. This adds complexity to deciding which way is best to distribute the available manufacturing capacity. The interest is in maximising the utilisation whilst taking into account the individual due times of products. Therefore, planning models that address those aspects and have an objective to minimise total weighted tardiness (TWT) must be developed and compared as well.

Therefore, this doctoral thesis has two aims: The first aim is **to assess in which types of scenarios in LSA, a mobile automation system should be preferred to a dedicated automation system.** Meeting this aim not only presents a comparison between the systems but also provides a baseline for what would be expected from a good self-organisation model. Following from that, the second aim of this thesis is to **investigate self-organisation behaviour models for of mobile robots in LSA.** The models should allocate manufacturing resources to products autonomously. The allocation should consider the due times and tardiness costs in such a way that would lead to the lowest possible TWT.

In order to meet the set aims, three specific objectives are created:

1) Identify and evaluate in which scenarios a mobile system should be preferred to the fixed automation system
2) Investigate different behaviour models for self-organising mobile robots in LSA
3) Compare and validate the investigated self-organisation models to determine which one performs better in any considered environment

As is common with any set of objectives, it is necessary to validate or verify that they have been met. Objectives 1 and 3 require no validation because they are comparisons. Therefore, they have to be verified. Objective 2 requires a validation because it involves newly developed models and it is necessary to show that it functions as expected. To be able to verify the results of the comparisons and validate the self-organisation models, a set of requirements is set for this thesis in section 3.1.

The key performance indicators for the utilisation are the time proportions spent in utilisation, moving and waiting. Those for resilience are the production loss (PL) and total underproduction time (TUT). The control over product delivery times and optimality are characterised by how well either system or behaviour model can minimise TWT in the considered scenarios. The computational efficiency is measured by how much time is required by either model to proceed after a disruption.

1.2. Research Scope

The main interest of this doctoral thesis is linked to self-organising mobile robots in LSA. The focus is specifically on the operational characteristics of the mobile robot system rather than the specific challenges of the assembly process itself. Nevertheless, it is recognised that there are still many technical challenges, however, those have been left outside of the scope in order to have a suitable focus for the doctoral thesis. It is argued that it is difficult to justify addressing the technical challenges unless the operational advantages are known.

Thus, this work is in operations research aspects, such as utilisation of manufacturing resources, meeting due times and responding to disruptions. This means that the physical execution of manufacturing tasks, localisation, path planning and other similar activities are out of scope. Instead, estimated work rates of machines are established and used throughout the thesis as benchmarks. Therefore, the scope of this work is bound to the use of simulation models to firstly investigate the differences in performance between mobile and fixed automation systems; and secondly, to the investigation of ways of autonomously self-organising mobile robots in LSA.

In all the following work, a mobile resource is a set of mobile robots with specified manufacturing capacity. Due to there not being any benchmarks or physical comparisons, it is unknown exactly how many mobile robots could match the capacity of a dedicated automation system. Clearly, this depends on the individual characteristics of mobile robots and benchmarked fixed systems. According to [4], machining operations like drilling and riveting in aircraft assembly require the components to be pressed together from both sides. Therefore, a group of mobile robots that matches a unit of a dedicated automation system can be considered a multiple of two. Using the relative performance factors of both system types ensures the transferability of the attained results from this work.

1.3. Thesis Structure

The thesis is structured as shown in Figure 1-3:

- In Chapter 2, the state-of-the-art systems and the relevant academic literature are discussed
- The research problem and hypotheses are outlined in Chapter 3.
- In Chapter 4, the two manufacturing system types are compared to one-another in like-for-like scenarios.
- The development of two self-organisation models follows in Chapter 5, Chapter 6 and Chapter 7
- The work of this thesis is validated in Chapter 8 and brought to conclusions in Chapter 9

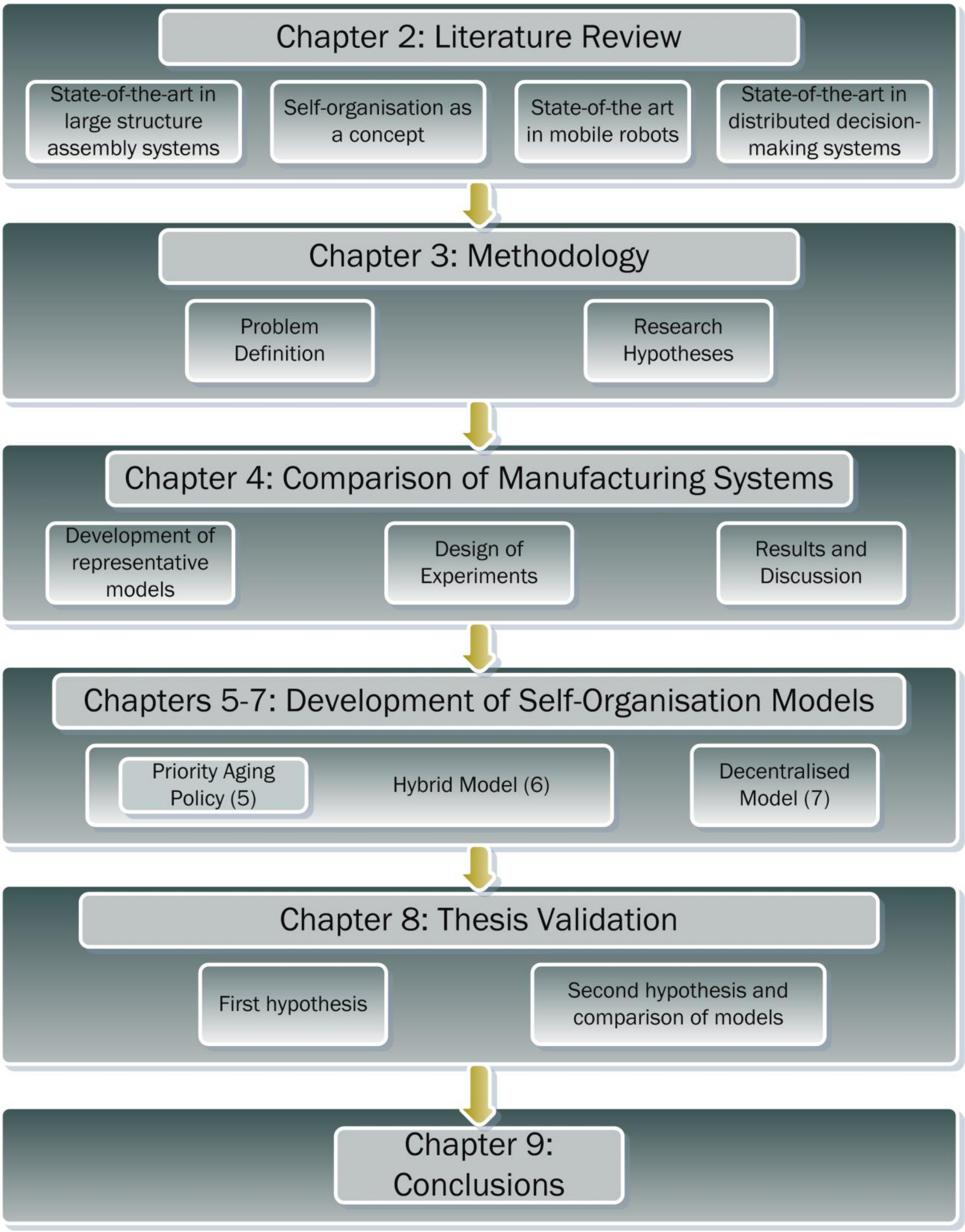

Chapter 2: Literature Review
State-of-the-art in large structure assembly systems
Self-organisation as a concept
State-of-the art in mobile robots
State-of-the-art in distributed decision-making systems
Chapter 3: Methodology
Problem Definition
Research Hypotheses
Chapter 4: Comparison of Manufacturing Systems
Development of representative models
Design of Experiments
Results and Discussion
Chapters 5-7: Development of Self-Organisation Models
Priority Aging Policy (5)
Hybrid Model (6)
Decentralised Model (7)
Chapter 8: Thesis Validation
First hypothesis
Second hypothesis and comparison of models
Chapter 9: Conclusions

Figure 1-3: Thesis structure

Chapter 2 - Literature Review

In this chapter, the relevant literature for this thesis is described. Essentially, the literature review is intended for bringing together some of the most recent and relevant work that can be used as a foundation for the underlying work. The structure in this chapter is such that the review starts with more generic topics and then brings the focus to the most-relating literature in each section. The review firstly examines what work has been done with mobile vehicles and robots in the manufacturing context. It then provides a brief overview of what is self-organisation. Then, a large proportion of the review is dedicated to the behaviour models that could potentially be applied to mobile robots for autonomous operation. The literature review concludes with a summary and an identification of knowledge gaps for this doctoral thesis.

As discussed in [22]–[24] and an overwhelming amount of academic literature articles mentioned below, the manufacturing industry has seen a shift from mass production to mass customisation. This means that manufacturers must accommodate for greater varieties of products, reduced time to market, reduced cycle times and reduced order lead times in their design [25] and often even make changes in their shop floor layouts [26], [27], [28]. Whilst changes in the physical layout and product requirements do not apply to the modern LSA environments on a frequent basis, there is no reason to believe that it will not do so in the coming decades.

2.1. Mobile Robots in Academic Literature

Increased interest has been shown towards mobile robots in the past few decades. The enabling technologies have matured enough to embed logic on a mobile robot to autonomously perform various manufacturing tasks. To date, some of the greatest challenges for mobile robots have been the accurate estimation of their location [29] and appropriate navigation [30]. Thus, the most popular applications have been the kinds where high accuracy of motion is not a necessity. However, the newest technology is pushing the limits towards enabling high-accuracy tasks, as shown in [21]. Thus, the capability of mobile robots has been steadily developing and there is much to benefit from further improvement. In the following sections, some of the most closely related mobile robot research is discussed.

2.1.1. Mobile Robots in the Literature of Schedule Optimisation, AGV Planning and Swarm Theory

In this literature review, mobile robots that are used for pick-and-place tasks are considered analogous to autonomously guided vehicles (AGVs). This is because the former uses its own robotic arm for handling products, and the latter relies on static robots at depots to do it for them. Thus, the difference is only in where the manipulator is positioned.

Gen and Lin wrote a survey paper on scheduling problems in manufacturing [31]. They provided a brief history of the problems and present a number of algorithms that have been used in job shop scheduling, AGV scheduling, dispatching in flexible manufacturing systems and other applications. They (also, Wan and Yuan [32]) stress that many optimal solutions in manufacturing operational problems fall into the class of NP-hard combinatorial problems. This means that from an algorithmic perspective, the problems of this type would most likely result in a polynomial increase in computational overheads when the input size (i.e. the number of agents, planning horizon, etc.) is increased. One logical continuation was to develop more efficient optimisation algorithms. In the 1960s, researchers proposed evolutionary programming, evolution strategies and genetic algorithms. Since then, in addition to developing a vast variety of new algorithms, academics have extended the existing ones to suit the needs and purposes of specific problems.

Recently, Saidi-Mehrabad, et al. [33] developed a two-stage Ant Colony Algorithm to schedule jobs on a job shop with AGVs. The objective was to minimise the makespan. They showed that the algorithm can achieve good results (but not optimal) in a “reasonable amount of time”. In the thirteen different experiments that they carried out, the computational effort to achieve a solution in some of the simpler experiments took 142 seconds. Whereas, when increasing the difficulty (more jobs, machines, AGVs, larger shop floor), the time to find a solution rose to hundreds of thousands of seconds. Therefore, the feasibility of using centralised algorithms decreases when the difficulty increases.

Kousi, et al. [34] investigated the effect of adding Mobile Assistant Units (MAUs) in an automotive manufacturing case study. A MAU is a vehicle with multiple shelves for carrying parts between locations. They proposed a centralised algorithm for short-term planning and tested it at different settings. They concluded that despite achieving interesting results in their work, the algorithm is not suitable for larger instances of the problem due to computational overheads.

Optimisation algorithms tend to be naturally tailored to achieving very good results in relation to their objective functions. However, as shown and concluded in the works above and other related literature [35], optimisation algorithms lose their feasibility once the considered scheduling problem increases in complexity (i.e. state space becomes too large). The increase can come due to an increased planning horizon, the number of controlled entities or their skill sets on the shop floor. A promising method of overcoming that complexity is by distributing the load of decision-making among the involved entities. For example, in swarm theory, large groups of entities are coordinated through the use of local rules. For robotics, this means that each robot has its own logic to follow in every condition that it may come across. The inspiration in swarm robotics comes from observing how societies of insects can perform tasks in a group that are beyond the capabilities of any individual in the group [36]. Applications with potential for use of swarm robotics include exploration, reconnaissance, search missions, the division of labour, collective transportation of objects and collective mapping [36].

For example, Liang and Lee [37] use elite individuals to preserve good evolution of collision-free path planning based on an artificial bee colony; Cheng, et al. [38] use simple consensus algorithms to cover unknown areas by mobile robots using sweep coverage; Arcaute, et al. [39] developed a mathematical model of attractive fields based on pheromones (these are commonly used by ants to help guide one-another in colonies). Examples like these distribute the decision-making load among the involved entities so that no single entity has to be responsible for the whole process.

The mentioned examples of distributed decision-making are a few out of many that provide a good insight into the field. It is accepted that learning from nature can give great ideas on how to approach engineering problems [36],[40],[41]. Leitao, et al. [41] add that the bio-inspired paradigms are best-suited for unpredictable environments. Due to the limited knowledge of the environment by each entity, there is an inability to plan forward or predict any future events. Therefore, in order to effectively enable these qualities in manufacturing, their suggestion is to combine these methods with those that have global knowledge. This way, the systems should be reactive in unpredictable circumstances and optimised in steady, predictable ones.

This section can be concluded by stating that the rigid, centralised algorithms are losing relevance in the modern manufacturing industry. The ever-increasing unpredictability in product demand and shortening product lifecycles are driving innovation in the field. On the computational level, this is analogous to the physical level: the currently common manufacturing systems exhibit similarly poor responsiveness to changes in production.

2.1.2. Mobile Robots in Manufacturing Literature

Buschhaus, et al. [42] state that there is great market potential in using standard 6-DOF industrial robots for manufacturing applications. This is mainly due to the universality of performing any type of movement in space at a lower price than CNC machines. The reasons for their limited use are the limitations of lower static stiffness of the serial kinematic chain than CNC machines. By mounting such an arm on a mobile platform, the stiffness and resultant accuracy decrease even further [43]. It can, therefore, be deduced that additional structural and accuracy features must be applied to mobile robots in order for them to be applicable in high-accuracy manufacturing tasks.

Chryssolouris, et al. have published several academic papers regarding the application of mobile robots in the automotive industry. In [44] they assess the advantages of mobile robots' ability to relocate on the shop floor and carry out a plethora of production processes with the 6-DOF manipulator. They find that mixed-model assembly systems (including both static and mobile robots) exhibit increased responsiveness both to planned system reconfigurations and unpredictable manufacturing resource breakdowns. This strongly increases the manufacturing system's ability to handle a variety of products. They stress that the ability to do so is of paramount importance in the shift from mass production to mass customisation [24]. In [45], Michalos, et al. point out the challenges of replacing or adding a static robot in a workstation. They then use an automotive case study to design and simulate the action plan for autonomously replacing a robotic unit. As a result of that, the reconfiguration time of the given workstation was considerably shortened and human intervention was not necessary either. In [46], they developed a planning model for a mixed-model assembly system. They presented an approach for task allocation to both static and mobile robots in their automotive case study. They conclude that is necessary to develop standardisation of hardware and software interfaces to enable seamless 'Plug & Produce' (PnP) [26] behaviour. The PnP concept is a drive towards easier installation of assembly devices, analogous to the "Plug and Play" concept in computing. It has mostly been proposed for static robots [47] because those have been more widely adopted in the industry. However, the concept suits mobile robots even better because they can autonomously change workstations.

The key point of interest in the automotive applications seems to be the reconfiguration of workstations by means of adding, removing or replacing robots. As a result, a workstation configuration with the necessary capabilities can be achieved and the production of a new product may start. In the case of replacing a broken down or inefficient robot, an appropriately functioning one can be deployed.

Some cooperative assembly tasks in LSA have been addressed by researchers over the last two decades. For example, Simmons, et al. proposed a method for coordinating three robots with completely different configurations in [48]. They used an overhead crane, a mobile manipulator and a roving eye to perform a high-precision docking task of a metal beam that none of them would be able to achieve individually. Pott, et al. take advantage of the dramatically expanded working envelope of mobile robots in comparison to fixed robots in [49]. They presented conceptual work for parallel cable-driven robots to perform the final assembly of solar power plants on-site. They argue that similar approaches can be used for building, maintenance and material handling in big warehouses. Dogar, et al. coordinated teams of mobile robots to carry objects that are too heavy for a single robot; co-localise holes; and insert fasteners in them in [50]. These publications target the applications where single machines or those constrained to a fixed point are impractical or ineffective to use.

Some of the closest work to this thesis' was found in [51]. There, Giordani, et al. proposed a "two-layered decentralised multi-agent system" to self-organise mobile robots on a manufacturing shop floor. On the first layer of the system, the products request the central coordinator agent for the right to invite resources. The coordinator agent then assigns the products a certain number of resources based on the product requirements and production costs. On the second level, mobile robots are assigned to specific products with the objective of minimising the total movement cost. The mobile robots are considered homogeneous and the interest is purely in shortening the movement distances. As a result, the proposed model effectively minimises the make span of product batches.

The given paper also provides some good reasons for why using mobile robots can be beneficial. The key point is the fact that there is no need to set the shop floor layout in the planning stages, as is the norm with static machines. The less constrained layout of dynamic systems consisting of mobile robots allows those decisions to be postponed to the operative level with the option to reverse them whenever required.

Furthermore, they point out that controlling mobile robots by means of embedded agents is suitable due to the physical nature of mobile robots themselves. Firstly, this is due to mobile robots having limited perception and communication capabilities. Secondly, they are not constrained to any single location on the shop floor and can autonomously relocate themselves. Thirdly, their proposed multi-agent approach claims to overcome the issues of centralised approaches associated with a single critical point of failure, scaling, fault tolerance and others [52].

The point of criticism is that the paper claims to present a decentralised multi-agent system, however it has a single central coordinator agent. Whilst much of the planning and allocating load is indeed distributed among the manufacturing products and resources, the coordinating agent is the single point of failure in the system. It also processes a large amount of information. This means that if there is some form of error with that agent, then the whole system may come to a halt, much like traditional, centralised systems. Nevertheless, there is very little that could go wrong with the coordinator agent per se and it would be a trivial option to have a backup ready to take its position in case of issues. Therefore, it is fair to consider this model as a hybrid, because it consists of both centralised and decentralised elements.

2.1.3. Section Summary

In this section, models and algorithms to govern mobile robot behaviour and their general directions in academic literature have been presented. The section covered both: the applications where mobile robots have been used and the algorithms/models used to govern them.

It can be concluded that so far mobile robots have mostly been used for tasks that do not require high positioning accuracy. This can be explained by the fact that the enabling technology for more demanding tasks (like those in automotive and aerospace manufacturing) is still maturing to the necessary levels for LSA.

2.2. Self-Organisation

Self-organisation as a term has been used in many research fields. Examples can be brought from urban traffic control [53], stem-cell growing [54], nanofibres [55] and even pedestrians on pavement [56]. In all of them, researchers study how small, independent entities spontaneously act to achieve an orderly arrangement of a large group. In manufacturing, self-organisation means that the system actively participates in its design at creation time and manages itself during operation [57]. This means that when a product's requirements become known, the entities of the manufacturing system autonomously find suitable partners and positions in the shop floor layout. During operation, entities adapt their behaviours or change the shop floor layout if a disruption of any kind has been encountered.

Self-organisation as such is a broad term and can include a range of self-* properties, such as self-optimisation, self-healing, self-configuration, self-diagnosis, self-adaptation and others [41], [58]. However, for the purposes of this doctoral thesis, self-organisation is limited to the autonomous distribution of manufacturing resources to products. The interest is in two aspects – the efficient allocation of resources to meet product due times and the mitigation of negative impacts when disruptions occur. The ability to mitigate or absorb the impact of adverse events and return to a steady state is called resilience [27]. It is accepted that most engineered systems are designed with little to no consideration for disturbances and can, therefore, become unreliable under unpredictable circumstances [28]. Gu, et al. [59] defined that for manufacturing, the resilience parameters are the total underproduction time (TUT), total settling time (TST) and production loss (PL). These are illustrated on a throughput over time plot in Figure 2-1. TUT is a measure of how long a system has spent in underproduction or behind schedule. TST is an indicator of how long it takes a system to reach a steady state after a disruption ends. This is generally important in cases where buffers have been set up to avoid creating an instant bottleneck from a single failure. The PL is the exact quantity by how much working content the system is behind schedule. Thus, a well-designed self-organised assembly system (SOAS) must be efficient with respect to the general objectives of the production system as well as naturally exhibit a high degree of resilience.

Clearly, it is beneficial to recover from disruptions as quickly as possible. Scheduling models have traditionally relied on centralised architectures. However, they have not coped well with the increasingly dynamic character of modern market demands. As is discussed in the next section, distributed planning models have been developed to overcome that.

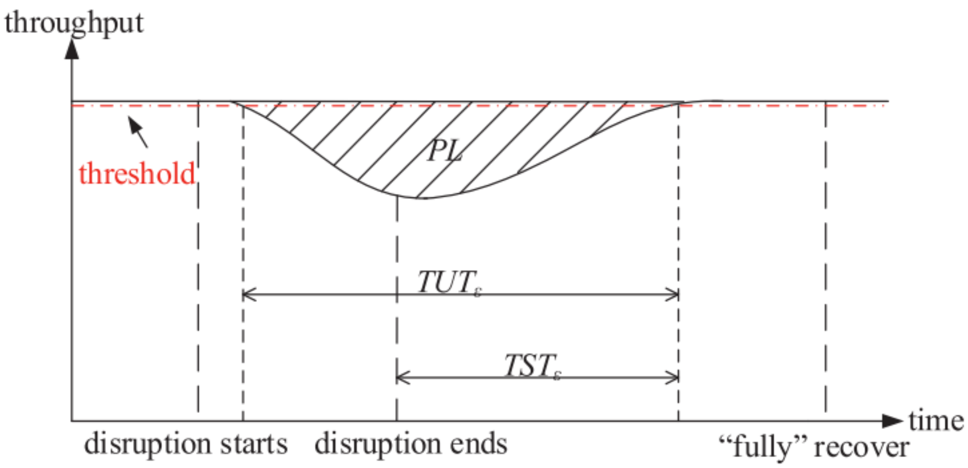


Figure 2-1: An illustration of resilience measures [55]

2.3. Scheduling/Planning/Decision-Making Systems

In this section, various decision-making/planning/scheduling systems for manufacturing are presented. As was shown in section 2.1, several algorithms and behaviour models have been developed for controlling mobile robots. None of them were found to directly address the needs of self-organising mobile robots in LSA (product-centric scalability and meeting due times). This section provides an in-depth review of the kinds of approaches used in manufacturing. This enables narrowing down the options when identifying suitable architectures for self-organising mobile robots in this work.

In early research, centralised scheduling has dominated in literature. However, in the past few decades, there has been increasing pressure to either make the centralised approaches more efficient or shift over to decentralised ones. This is mainly due to the traditional methods' slow response to any disruptions, increasing complexity or changes in production requirements.

Perturbations often lead to the need to change the scheduling code. With some larger centralised applications running on millions of lines of code, these changes are very costly to implement in terms of time and effort. Whereas, in distributed systems, the code is often only hundreds of lines long [60]. It is, therefore, substantially easier to change. Furthermore, distributed systems have been shown to respond to disturbances better than centralised ones [61], [62]. This is because, in a distributed system, a disturbance has a local effect, whilst in a centralised system, it has a global effect [63]. Minor updates in information or disruptions on the shop floor may impact the schedules of only a few entities on the shop floor. In decentralised systems, this is handled by only the affected entities, as opposed to rebuilding the whole schedule by a central entity. The requirement for a great amount of information and the processing of it becomes particularly hindering for centralised systems. Unexpected events can be, for example, power blackouts, machine breakdowns, late supply arrivals or rush orders [61]. Whilst the former three examples can be controlled or mitigated to a reasonable extent by the manufacturer, rush orders depend on customers. A rush order is an order that is not considered in the main production schedule at the time of its arrival. According to [64], rush orders for prototypes, replacement orders or specific customer demands have become a regular part of companies' daily business. For the reasons above and additionally due to mobile robots enabling quick changes on the shop floor, fully centralised decision-making algorithms are not considered for self-organisation in this thesis.

2.3.1. Distributed Decision-Making

It is generally accepted that the planning of tasks will become increasingly automated in the future [65]. With the new technology paradigms, the IoT (Internet of Things) [66] and Industrie 4.0 [67], large networks of interconnected cyber-physical systems are envisioned. In these networks, entities autonomously interact with one-another to trigger certain actions which result in achieving their goals. This is in line with the vision of controlling large numbers of mobile robots in an assembly environment. As discussed in the section above, centralised coordination of these is not sensible due to high computational overheads and consequently poor responsiveness.

Much of the early distributed decision-making in related fields was carried out in the form of swarm robotics. The resultant reactive behaviour is excellent for highly unpredictable environments and if there is little interest in efficient performance. Manufacturing environments, however, require more intelligent approaches in order to achieve higher utilisation of machines and consequently better ROI. For such purposes, a very promising paradigm has been proposed and developed over the past several decades: multi-agent systems (MAS) [52]. This paradigm is described in section 2.3.1.1.

2.3.1.1. Multi-Agent Systems

As the name suggests, MAS is a system that is comprised of multiple agents. This subfield of computer science is considered relatively new because it has been studied since the latter half of the 20th century and only received wider recognition in the 1990s [52]. There have been many definitions to agents. For example, Wooldridge and Jennings define that "an agent is a computer system situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives" [68]. Leitao [69] defines an agent as "an autonomous component that represents physical or logical objects in the system, capable to act in order to achieve its goals, and being able to interact with other agents when it does not possess knowledge and skills to reach alone its objectives." Guo and Zhang [70] defined it in a more straightforward way: "Agent is the object which could finish the given task independently without people's interference". Whilst there are many more definitions (see [71], [72]), there is a consensus that agents are computational entities that have the ability to communicate with one-another and aim to work towards their objectives autonomously. Such characteristics in the digital environment form a good parallel to mobile robots in the physical environment.

2.3.1.1.1. Product Intelligence

Generally, in manufacturing MAS, product agents are responsible for their respective products’ manufacturing phase. However, recently the paradigm of product intelligence has gained interest. This paradigm links the physical order or product to information and rules that govern the way it is intended to be made, transported or stored [73]. This way, the basic concept of a product agent in MAS is expanded. According to [74], an intelligent product has its own unique identification, can communicate with its environment, store information about itself, share that information and is capable of participating in decision-making that influences its lifecycle. This intelligence can either be embedded in a product or in a remote location (i.e. PC or cloud computing service). The intelligence can also be expanded from only the product itself to the range of sub-assembly parts that it is made of [75]. Also, in some cases, even the life-cycle of the product may be considered [58]. This approach makes the production more product-based and suits the decentralisation narrative very well. In the context of developing MAS, this can be seen as a considerable step forward from the initial agents with basic rules for negotiations. In LSA, many products require not only a high volume of work to be processed but also a high number of different tasks. Applying this concept to products in this context could enable a fluid transition from raw material to finished assemblies. That would be followed by self-monitoring for maintenance purposes.

2.3.1.2. MAS Architecture Types

This section describes the architecture types that MAS have been based on throughout literature. Three common types have been outlined: The hierarchical (functional), blackboard and heterarchical types [76]. Illustrative examples of these architectures are shown in Figure 2-2.

In the hierarchical (or functional) architecture, each function of a whole process is a single agent. The agents are given certain pathways to communicate (customer-supplier or master-slave) and thus the system lacks distribution [77]. This architecture is most frequently applied to e-commerce, because it is well-suited for one-on-one trading purposes as shown in [86]. However, it is too constraining for manufacturing purposes.

The blackboard architecture relies on a central blackboard agent for information exchange. The blackboard agent (BA) can be used as a central database where all agents can request information from. It may also use its information to predict and notify of any upcoming issues that the other agents have not been able to plan for. This does not mean that the BA controls the rest of the system. Such a system can be considered partially centralised and partially decentralised because it has centralised and decentralised elements in it. Despite being helped by a central source of information, the decision-making logic of other agents in this architecture is still inherently complex [78]. Due to using a combination of centralised and decentralised entities, this architecture can be considered a hybrid one.

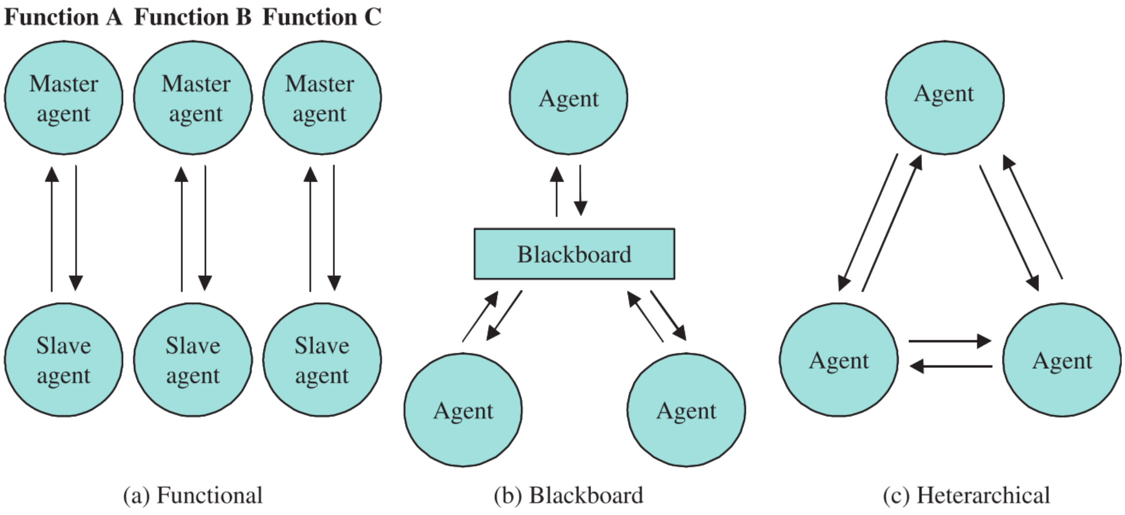


Figure 2-2: The three general MAS architecture types [85]

There are arguments both in favour of and against using blackboard architectures. Proponents are interested in combining the benefits of centralised optimisation and decentralised responsiveness [41], [73], [79]. The adversaries are challenging these claims by saying that it is incompatible to combine the two opposites. The incompatibility is due to the blackboard architecture promoting a collaborative environment, while the general direction and purpose of the decentralised approach have been to create a competitive environment [80]. Furthermore, it has been shown that in addition to the gathering and processing of large amounts of information, it can be challenging for a central agent to handle the communication overheads [81]. Nevertheless, it can be argued that if a MAS with a blackboard architecture can be designed to meet its requirements, then there is no sense in arguing about the initial intentions of the paradigm. Instead, it can be seen as an alteration to achieve a more efficient solution.

An analogy can be seen between the blackboard architecture and the HMS approach. Holonic Manufacturing Systems (HMS) is a specifically structured case of recursive MAS [82]. By recursion, it is meant that each level of operation can theoretically be broken down to smaller levels and be part of a greater system at the same time. The functions of a BA can be very similar to those of the supervisor holon in HMS. The only major difference is the recursion that the blackboard architectures usually do not exhibit. By recursion, it is meant that each level of a system consists of lower levels at the same time as being part of higher levels.

The heterarchical architecture type is the classical architecture of a distributed system. This architecture type enables the greatest variety of process routes for any task. That is because each agent can talk to any other agent and normally no single agent has full knowledge of the environment. Jennings, et al. emphasized the power of this parallelism [71]. Any idling agents can request for work from other agents without posting to a blackboard or joining a queue. Furthermore, finding replacements or additional resources is straightforward and can only require a single exchange of messages. Therefore, this architecture type enables the full benefits of distribution; however, it is more difficult to achieve efficient results due to each agent having limited knowledge of the environment. Furthermore, these can occasionally result in particularly undesirable emergent behaviour [83].

2.3.1.3. Example Architectures

In this section, some of the most known MAS/HMS architectures are presented. It is shown that they are mostly intended to act as generic architectures that can be specialised to address specific problems. The discussion in this section gives an overview of widely accepted architectures that could be used for self-organising mobile robots in this thesis.

A very early example of an HMS is the PROSA [84], which literally stands for Product-Resource-Order-Staff Architecture. It was presented in 1998 and consists of product, resource and order holons.

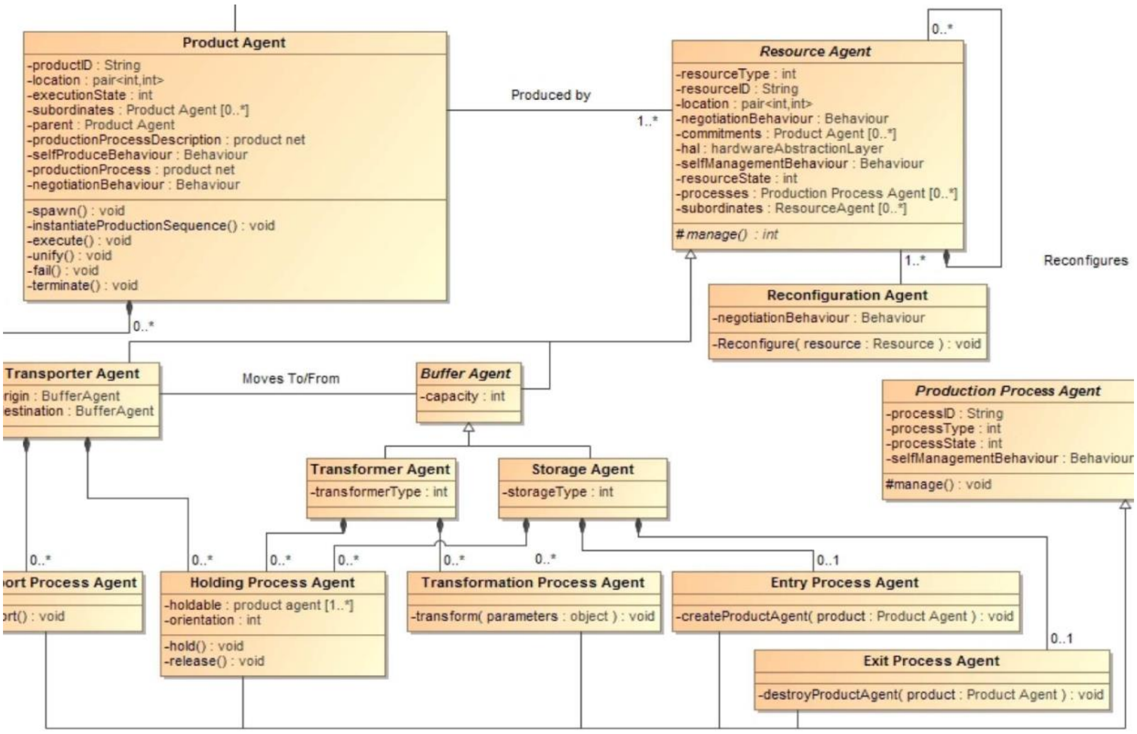
The Holonic Component-Based Approach (HCBA) [85] to reconfigurable manufacturing control architecture was presented in 2000. In this architecture, the holons are tied to the component level, not planning. Thus, each whole robot can be a holon, with different sensors, manipulators and grippers acting as sub-holons within it.

The ADAptive holonic COntrol aRchitecture (ADACOR) [86] was based on PROSA. It was presented by Leitao and Restivo in 2006. They kept the basic PROSA holons and added to them a supervisor holon to act as a coordinator and global optimiser within its holon. It also participates in the coordination and formation of holon groups. As such, the supervisor holon turns the architecture into a hybrid one. In 2015, Barbosa, et al. added self-organisation capabilities to the architecture and called it ADACOR^2 [87]. Self-organisation was enabled in two forms: behavioural adjustments for minor perturbations and structural adjustments for major changes. Then, in 2016, Leitao and Barbosa presented a dynamic switching mechanism for scheduling manufacturing operations [88]. The system was designed to respond to disturbances in an agile manner, freeze the early part of the obtained sub-optimal schedule and optimise the latter part of the schedule. This way, the system becomes responsive to changes and effective in building long-term schedules. The challenging part of this system may arise if the initially frozen part of the schedule happens to be poor and will negatively affect the part of the schedule that is being optimised. Nevertheless, it is a great attempt at bringing together the qualities of two fundamentally different control structures.

The MetaMorph structure [89] was an early paper (1995) where heterogeneous agents were clustered into groups and then coordinated by a central mediator agent. The architecture is very similar to those of HMS and it is possibly even a direct precursor.

Farid and Ribeiro claimed that one of the barriers to adoption of MAS and RMS by manufacturers was the absence of formal and quantitative MAS design methodologies. They proposed the Axiomatic Design of a Multi-Agent Reconfigurable Mechatronic System (ADMARMS) in 2014 [90]. The key points of interest in this architecture are the facts that they do not neglect the transportation factor, they root the system in established engineering methodologies and designed the agent-based architecture for maximum reconfigurability.

The architecture and data model of ADMARMS is shown in Figure 2-3. It is shown how there is a set of agents that support the traditional product and resource agents.



Product Agent
-productID : String
-location : pair<int,int>
-executionState : int
-subordinates : Product Agent [0..*]
-parent : Product Agent
-productionProcessDescription : product net
-selfProduceBehaviour : Behaviour
-productionProcess : product net
-negotiationBehaviour : Behaviour
-spawn() : void
-instantiateProductionSequence() : void
-execute() : void
-unify() : void
-fail() : void
-terminate() : void
Produced by
1..*
Resource Agent
-resourceType : int
-resourceID : String
-location : pair<int,int>
-negotiationBehaviour : Behaviour
-commitments : Product Agent [0..*]
-hal : hardwareAbstractionLayer
-selfManagementBehaviour : Behaviour
-resourceState : int
-processes : Production Process Agent [0..*]
-subordinates : ResourceAgent [0..*]
#manage() : int
0..*
1..*
Reconfigures
Reconfiguration Agent
-negotiationBehaviour : Behaviour
-Reconfigure(resource : Resource) : void
0..*
Transporter Agent
rigin : BufferAgent
estination : BufferAgent
Moves To/From
Buffer Agent
-capacity : int
Production Process Agent
-processID : String
-processType : int
-processState : int
-selfManagementBehaviour : Behaviour
#manage() : void
Transformer Agent
-transformerType : int
Storage Agent
-storageType : int
0..*
0..*
0..*
0..*
0..*
0..*
0..1
ort Process Agent
rt() : void
Holding Process Agent
-holdable : product agent [1..*]
-orientation : int
-hold() : void
-release() : void
Transformation Process Agent
-transform(parameters : object) : void
Entry Process Agent
-createProductAgent(product : Product Agent) : void
0..1
Exit Process Agent
-destroyProductAgent(product : Product Agent) : void

Figure 2-3: The ADMARMS architecture and data model [96]

The entry and exit process agents create and destroy agents as necessary. A reconfiguration agent is linked to the resource agent to trigger the reconfiguration process. Two agents are responsible for transportation – the transport process agent creates the contract which the transport agent can then choose to accept. Other agents either facilitate the storage; specification of production details or transformation of parameters within any of the agents due to reconfiguration. The resultant architecture is very responsive to reconfigurations on the shop floor, both due to the design modularity and due to applying a distributed decision-making model.

The inteGration of pRocess and quAlity Control using multi-agEnt technology (GRACE) [58] project was launched in the summer of 2013. Its aim was to develop a system to supervise quality control stations and coordinate operation through a network of collaborative individual agents. This project highlights a function of product intelligence, as discussed in section 2.3.1.1.1. In this case, the product agent can gather information about itself from other distributed entities and later participate in decision-making when its life-cycle is concerned.

It can be concluded that efforts have been focused to introduce distributed technologies at several levels in manufacturing. Component, planning/scheduling, reconfiguration, design and execution levels have been addressed by various research projects. While the shown architectures are by no means exhaustive, it is clear that the earlier and more basic architectures have been used as foundations for the more advanced architectures that have been proposed later. For the purposes of this work, two architecture types seem to be promising. One would include coordination by some central entity (blackboard) and the other option would not.

2.3.1.4. Negotiating in MAS

Negotiation is a fundamental interaction mechanism in any MAS. The ability of entities to influence one-another or come to a consensus through information exchange is one of the main things that make MAS more intelligent than swarm robotics, where communication is done by interacting with the environment instead [36].

The Contract Net Interaction Protocol (CNP) [91] has been commonly applied to describe agent interactions through communication. It has been developed by the Foundation for Intelligent Physical Agents (FIPA). An example of such a protocol is shown in Figure 2-. By definition, each agent can be an initiator, a participant or both at any instance in time.

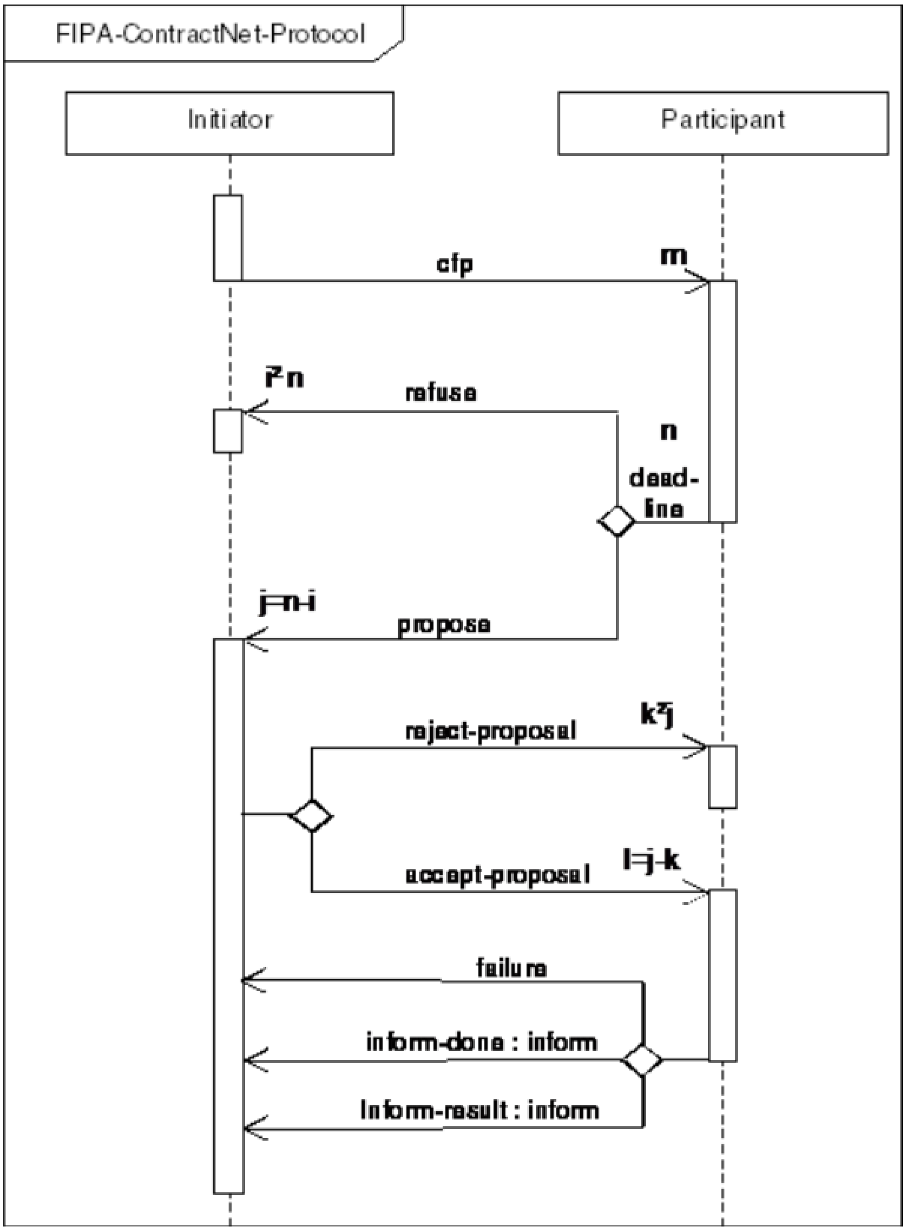


Figure 2-4: An example Contract Net Interaction Protocol [91]

A negotiation is triggered when an agent (initiator) identifies a need to request or offer a good or service. It searches for suitable addressees (participants) and messages them the offer. The offer can then be accepted, declined or negotiated further. Occasionally, negotiation conflicts may occur in task sharing [92]. These can be instances where either more than one or no participant agrees or to the request of the initiator. In these cases, conflict resolution strategies must be incorporated into the agent behaviour models to overcome them.

Much of the attention in this field relates to e-commerce (buyer-seller) where negotiation is the main purpose of the agents. Some of the early negotiation methods, like [93] and [94], were based on game theory. In [95], Gatti, et al. proposed a bargaining method where one of the negotiation sides is under a time-constraint. In [94], Fatima, et al. propose a framework for multi-issue negotiations between agents with their own agendas. However, they also admitted that existing game-theoretic models have two inherent limitations [98]: firstly, they commonly assume that the opponent has complete knowledge and is thus more informed than in real cases; and secondly, they assume that the agendas of all agents are fixed.

In a cooperative environment, it makes sense for agents to share their information; however, this is not the case in competitive environments [96]. A major challenge in competitive environments is the ability to predict the behaviour, preferences or interests of the opponent. In [97], Zhang, et al. propose a Bayesian-based approach to help an agent predict the opponent's preferences. Renne presented a model [98] where agents can "overhear" the messages that are sent between others and use them for justifying their logic in the prediction of future events. Thus, several methods can be used to acquire and make use of information.

Many different methods have been used for negotiations. For example, Wong and Fang designed ECNPro for multilateral agent-based negotiations [93]. The system treats multilateral negotiations as several concurrent bilateral ones and can adjust its negotiation strategies as required. Mansour and Kowalczyk provided a method [99] for coordinating the bidding strategy when there are multiple suppliers who offer multiple distinct services to the bidder. Sarvapali, et al. [100] stated that the interest in academia has grown towards Persuasive Negotiation (PN), where agents use threats, rewards or appeals in an attempt to convince the opponent to accept any given offer. Such models are effective in situations where there are repeated encounters. In [100], their proposed model uses rewards that can be given as an incentive to opponents to accept offers. In [101], An, et al. presented a model where negotiation agents need to attract multiple resources at the same time. There, the challenge lies in ensuring that the exact number of resources is attracted. They designed their negotiation agents to adjust the concessions that they are willing to make when conditions change and the number of tentative agreements for each resource. Ferreira, et al. [102] proposed a method where agents negotiate only in the near future. This enables a smooth transition when any immediately planned events occur, however it lacks vision further into the future and may not foresee problematic scenarios developing early enough.

Whilst there have been many negotiation methods developed, there seems to be a lack of systematic classifications and categorisations for manufacturing applications. Thus, each of the presented methods has been developed for a specific purpose and there remains a lack of standardisation. Bridging this gap will allow new MASs to be developed more easily and reliably by making use of previous experiences and a systematic approach.

2.3.1.5. Scheduling/Planning Problems and Models

In this section, some typical planning problems and models in MAS are presented. Agnetis, et al. have published a book on MAS [103] with comprehensive explanations on the field. The concept involves one set of agents (products) that have a set of jobs (tasks) that need doing and another set of agents (resources) that are capable of doing them. The tasks take some time to process and the system as a whole must find the best ways to do so. The agents can be set to be either greedy (competitive) or cooperative. In the greedy option, each agent is driven by its individual cost function and disregards interests of others. In the cooperative option, agents are driven by some common interests. Depending on the problem, the aim is to best utilise some limited resources (i.e. time or energy [104]).

A wide range of scheduling problems has been addressed by MAS. The problems include different numbers of machines and products; sequences of tasks, constant or variable processing times, pre-emptive jobs and different objective functions. Each of these can contribute to the complexity of the scheduled problem.

Some recently published literature includes: allocating tasks to parallel machines with the objective to minimise earliness and tardiness (just-in-time) [105], minimising total energy consumption and TWT in job shops [106], controlling AGVs in the dough making process [107], single-machine scheduling with sequence-dependent setup times and maintenance [108], two-machine scheduling to minimise makespan and individual working cost-functions [109] and scheduling two products on parallel machines that have different processing speeds [110]. Thus, very different problems in very different environments can be of interest.

Some of these papers do not have a direct application or case-study in manufacturing [108]–[110]. Their intention is to identify a generic problem that may arise in a manufacturing environment and provide a “one size fits all” solution to it. Contrarily, a clearly outlined case study and industrial application from Mes, et al. [107] presents case-specific work that may only be applicable for a very specific environment. Therefore, not only can there be a variety of problems but the scope of the solutions can be broad or specific as well.

Whilst not directly related, these works provide valuable insights into the architectural and model designing aspects for the work in this thesis. As a result of the literature discussed in this section and in section 2.1.2, it can be concluded that MAS is the most promising paradigm for controlling mobile robots in LSA. This is because similarly to agents, independently acting mobile robots have a limited perception of the environment and can access additional information through communication. Further similarities, such as responsiveness, adaptability and versatility are also observed due to their distributed nature.

The use of swarm robotics is not suitable, because they lack the level of intelligence for the tasks. Due to a lower level of coordination, this approach would require a higher amount of spare capacity to guarantee the same throughput. This means that manufacturers would be required to invest more capital into hardware.

Centralised methods of scheduling are not suitable either due to a single entity that processes all of the information. As discussed above, the responsiveness of such methods can become prohibitively poor.

Out of all of the found papers, none seem to be suitable for effectively self-organising mobile robots in LSA. This is because none of them takes into account product-centric scalability and meeting product due times at the same time. Because there are no competing models in this context, there is no need to analyse any of them in depth. Therefore, using the basic architecture types and relevant experiences from related papers, new self-organisation models had to be proposed.

The functional architecture type is not suitable for this problem, because it has a rigid master-slave relationship. The other two, blackboard (partially decentralised) and heterarchical (fully decentralised) were found to have potential.

The blackboard architecture possesses a central entity, which enables coordination in the system. Because the considered problem (product-centric scalability and meeting due times) was found to never have been addressed in academic literature, inspiration was taken from computer science. There, priorities of tasks are “aged” by increasing the priority values until eventually each task get processed. This approach, therefore, requires a suitable priority aging policy, as discussed in [111]. The priority aging process required to be modified in a suitable way to enable meeting product due times. These agents would then use their credits to give an incentive for the mobile agents to work on them.

The heterarchical architecture does not possess any central entities. Therefore, the activities must be driven by other means. Based on the experiences of researchers in MAS literature and in particular [93], [112], it makes sense to use an economic system similar to those used in the real world. The research approach for investigating behaviour models based on the mentioned architectures is described in detail in Chapter 3.

2.3.1.6. Industrial Adoption of MAS

Throughout this literature review, a plethora of reasons in favour of using agent-based systems has been presented. However, despite the field having been actively researched for several decades, the adoption has been very limited. Many reasons have been identified for this [69], [113]–[116]:

- **Capital investment.** Perhaps the main disadvantage is that adding redundant equipment (as spare capacity) to an existing system may incur prohibitive costs. This is a particularly important challenge because, at the design stage, it is difficult to economically measure the benefits. This also explains why MAS has been adopted much faster in the IT sector, where replication does not incur meaningful costs.
- **The education system and the available workforce.** Due to the limited acceptance and relatively recent introduction of the concepts, very few educational institutions include it in their curriculum and few professionals specialise in the field.
- **The emergent behaviour can be hard to predict.** In a distributed layout it is difficult to guarantee that the system will perform in the desired way.
- **Scalability.** When agent-based systems are expanded to thousands of agents and there is a need for frequent negotiations, in some cases the communication load in the systems can become challenging.
- **Standards.** The only known standard in the field was found to be the messaging standardisation developed by FIPA.
- **Industrial controllers.** Usually, controllers in industrial environments have not been designed to be controlled by agents.

- **Misapplication.** MAS is not a one-size-fits-all solution and the occasional misapplication has affected their reputation.

Alvarez Peixoto, et al. argued that conclusive research into the feasibility of MAS must be carried out in order to convince manufacturers [117]. The rest of the limitations would then start getting resolved due to wider acceptance.

2.4. Knowledge Gaps

This chapter has provided an overview of self-organisation, mobile robots, their applications and means for their control. It was shown that mobile robots have been successfully used in tasks like pick-and-place and area coverage. Some conceptual work was presented from the automotive manufacturing industry. No direct work was found to have been done specifically for the purposes of self-organising mobile robots in LSA. On the one hand, this is surprising because in LSA the mobile system's ability to scale up production rates on products could become a great advantage over fixed systems. On the other hand, this makes sense because the enabling technology has only recently made the industrial robots capable of the high accuracy tasks that are required in LSA. Thus, it was established that **it was unknown under which circumstances mobile automation systems have operational advantages over fixed systems in LSA**. This is the first knowledge gap that must be overcome to understand the feasibility of deploying mobile systems in LSA.

As any other automated manufacturing system that requires planning, the mobile system requires a decision-making model for functioning. Because self-organisation of mobile robots in LSA has not been researched, **it is unknown how the planning of resource allocations should be carried out most effectively in manufacturing environments of LSA**. It is predictable that there can be environments with different types and frequencies of disruptions and thus it may not be feasible to use a single solution for each of them. The literature review discussed the differences between fully centralised, partially centralised and fully decentralised systems. Despite optimisation, fully centralised systems were deemed incompatible for this work due to the inherent rigidity. The other two architecture types have some feasibility and their performance in various scenarios should be explored. Thus, **it is unknown in which manufacturing environments the mobile system should be controlled by partially centralised or fully decentralised decision-making models.**

2.5. Chapter Summary

In this chapter, the most relevant literature out of that found for this doctoral thesis has been reviewed. It was shown that most of the operational research on mobile vehicles has focused on pick-and-place tasks and topics related to swarm robotics. This can be explained by the fact that until very recently, mobile robots lacked the capabilities required for high-accuracy manufacturing tasks.

In manufacturing, tasks often require high structural stiffness in addition to accuracy itself. This is why CNC machines have been used so widely in the industry. Recent advances in robotics have enabled more accurate tasks (drilling and filling) to be done accurately enough by mobile robots for the automotive and aerospace manufacturing processes.

Some key conceptual work was presented and the potential benefits of mobile robots have been described. It was concluded that mobility enables applying changes on the shop floor much faster. Therefore, manufacturing environments with frequent changes or high schedule uncertainty should benefit from using mobile systems. The expected disadvantage of mobile systems is the accuracy. Firstly, mobile robots commonly rely on sensors to achieve locations estimates. Secondly, they have the additional challenge of structural stiffness. Both of these are engineering challenges and can be mitigated through additional capital investment.

The literature relating to mobile robots also uncovered the most common control approaches for mobile robots. Multiple sources claim that centralised scheduling algorithms are suitable only for small instances of such problems.

They commonly use decentralised models, and in particular, agent-based models for controlling mobile systems in related environments.

Out of the general MAS architectures, only blackboard and heterarchical architectures make sense for the problem in this doctoral thesis. The blackboard architecture possesses a central entity, which enables coordination in the system. The decentralised model required establishing a link between the economic considerations of the manufactured products and the agents that are responsible for them.

The methodology for comparing the fixed and mobile automation systems, developing the named self-organisation models and validating them is described in Chapter 3.

Chapter 3 - Research Methodology

To fulfil the objectives set in section 1.1, a research methodology is formalised. In this chapter, the research problem is defined, two hypotheses are stated and the research approach is described.

As discussed in the introduction, traditional manufacturing systems in LSA respond to changes in production very poorly and have a limited working envelope. Therefore, more versatile and flexible solutions than those must be explored in order to meet the needs of current and future manufacturing. An alternative to such systems is seen in the application of mobile robots. Due to not being physically constrained to any infrastructure on the manufacturing shop floor, mobile robots have the freedom of moving to any workstation at any time. As shown in the literature review (Chapter 2), some work had been done on the physical application of mobile robots. However, no direct work had been found to neither compare the system types in LSA nor develop the appropriate behaviour models for self-organisation in this context. In order to understand the aims and challenges of this doctoral project, different planning systems were described in Chapter 2.

The decision-making models for controlling manufacturing systems generally consider only one machine per product at any time. Throughout this thesis, each product is assumed to be product-centrically scalable, meaning that it is large enough for multiple mobile robots to fit around it in order to increase the work rate. The objective is to meet the due times of each individual product. This can be compared to scaling work rates with human work teams. Therefore, the self-organisation challenge in this thesis had to be achieved by proposing new self-organisation models.

The literature review highlights that shop floors can be controlled in centralised and decentralised ways. The expected general characteristics of these are illustrated in Figure 3-1. Centralised control is usually optimised because there is an entity that has complete knowledge and control of the environment. However, for the same reasons, an increase in entities on the shop floor leads to an exponential increase in processing time for each solution; thus, the problem is NP-hard. Therefore, if disruptions are frequent or the number of shop floor entities increases, the viability of using a centralised control algorithm decreases. This is because each rescheduling task means that the schedule must be reprocessed. In a properly designed decentralised self-organisation model, disruptions should not affect the whole system, instead, there should only be a local effect among involved entities. By solving the problems locally, the decision-making agents do not strain any central entity. This means that the manufacturing system should be scaled up without concerns about causing unreasonable computational overheads. The disadvantage of a decentralised approach is the fact that no entity knows everything about the environment and therefore there is no feedback on the resultant schedules. For this reason, it is impossible for the system to assess their optimality. An additional challenge in such a system may arise due to a high communication overload. This is because the system relies on constant negotiations between entities instead of a central decision maker that gives direct commands as and when necessary. For this reason, several different behaviour rules of the decentralised model must be tested at different settings to ensure the best possible outcome.

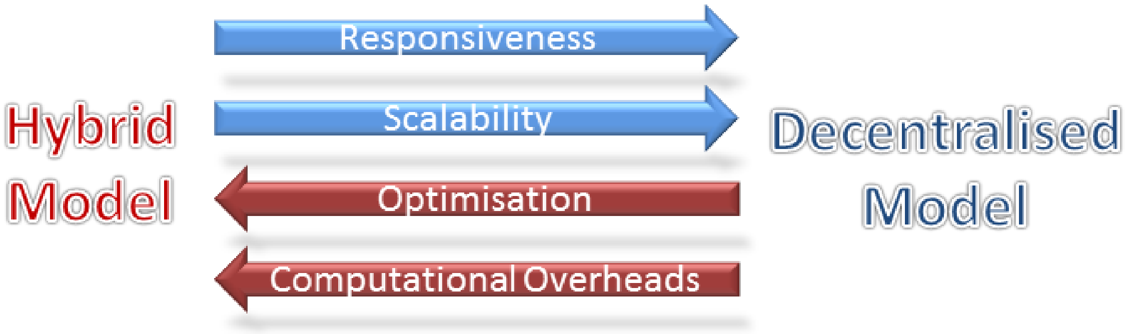


Figure 3-1: The expected differences between the models based on the literature review

3.1. Requirements for the Doctoral Thesis

The requirements for this thesis are listed as follows:

- The work in this thesis should be applicable to LSA
- The comparison of systems should highlight the differences in the systems' utilisation, control over product delivery times and resilience due to mobility
- The developed self-organisation behaviour models should be based on two fundamentally different multi-agent systems architectures
- The self-organisation behaviour models should autonomously arrange that the machines are allocated to jobs once products have been launched
- The models should work with the objective to minimise TWT
- The models should respond to any disruption automatically
- The comparison of models should show differences between the models' performance in relation to minimising TWT and computational efficiency.

3.2. Problem Definition

The problem definition for this doctoral thesis extends from the literature review. Large structure assembly generally requires the production of low quantities of products with high working contents. The products are usually very heavy, large and difficult to handle. From the manufacturing system's perspective, products' working contents must be processed by given due times. As each product is part of a larger assembly, it is important that no individual product is delayed. Tardy individual products may cause a delay in the release of full assemblies and consequently a monetary loss to the manufacturer.

Manufacturers aim to design their manufacturing system capacities to be able to process the working content of arriving products in a planned product flow. However, due to unpredictable events, it is highly likely that sometimes it will not be possible to meet the due times of each product. In such circumstances, the manufacturer aims to distribute the available manufacturing capacity in a way that minimises the negative impact of product tardiness. Therefore, the objective function of such a manufacturing system is minimising the total weighted tardiness (TWT). The weight in this function is the tardiness cost of products per unit of time and the tardiness is measured in units of time that a given product is tardy by.

A typical unit of a fixed automation system in LSA is shown in Figure 3-2. The machine is constrained to a gantry rail and, therefore, has a limited working envelope. Because it requires a substantial effort to load and unload products from workstations, the machines spend considerable time without utilisation. Consequently, manufacturers see a slower return of their capital investment, have little control over product delivery times and have a poor response to any required production changes.

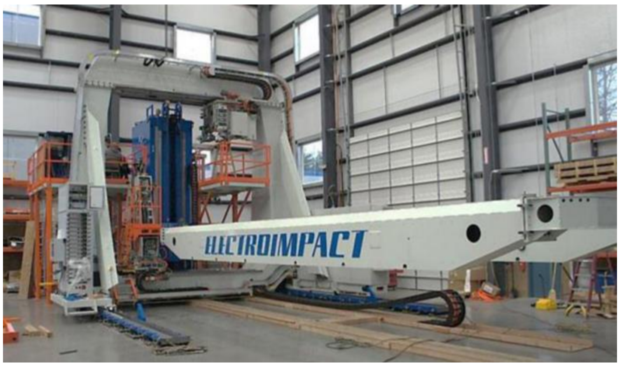


Figure 3-2: An example image of the ElectroImpact E7000 system [118]

A very important consideration for a manufacturing system is also its resilience. It was discussed in Chapter 2 how various disruptive events are increasingly becoming a part of manufacturers' daily business. Whilst disruptions are rarer in LSA than in most other industries, it is still important to evaluate and increase this performance characteristic. A point to argue is also the consideration that disruptions like accepted rush orders and changes in production requirements are rare due to the involved challenges of executing them. Hypothetically, if responding to such events became relatively effortless and of negligible cost, it would be much easier to accept them on the planning level.

With the physical considerations mentioned above, it is also important to establish the planning challenges. In this work, the performance of the mobile system is simulated in various manufacturing scenarios. These scenarios include processing different types of products, planning horizons and handling different frequencies of disruptions. The self-organisation behaviour of the mobile system should ideally react to any perturbations instantaneously and provide optimal planning with respect to minimising TWT. As was established in the literature review, scheduling models that include central sources of information tend to perform better with respect to the objective functions. However, the gathering and processing of information by the entity are very demanding in terms of computation and communication. This is particularly challenging when there is a large number of entities on the shop floor or the planning horizon is very long. An alternative to those are models that do not include any centralised entities. Therefore, a major part of the self-organisation problem is identifying which architecture type can provide the best compromise in terms of responsiveness and TWT minimisation.

3.2.1. Definition of Research Hypotheses

Until this section, it has been repeatedly discussed how and why the current state-of-the-art manufacturing systems in LSA are very constrained and irresponsive to any changes on the shop floor. It was proposed that dynamic systems consisting of mobile robots can overcome many of those challenges. Thus, the first hypothesis in this thesis is that **an appropriately controlled mobile system can be more utilised, resilient and has more control over product delivery times in dynamic scenarios of LSA than traditional, dedicated automation systems with identical working capacities.** By dynamic scenarios, it is meant that the shop floor encounters some frequency of predictable or unpredictable events that were not initially accounted for.

If the hypothesis holds true, then the next logical step will be to find out what kinds of behaviour models are best suited to facilitating the self-organisation of mobile robots in the given scenarios.

Much of the published literature ([65], [73], [86]) has shown that centrally controlled systems face many difficulties in situations relating to uncertainty and increasing numbers of controllable entities. According to [86] and [119], a well-balanced hybrid (partially centralised and partially decentralised) architecture should successfully combine the benefits of hierarchical (centralised) and heterarchical (decentralised) control architectures. Still, due to being only partially decentralised, it is predictable that in dynamic enough environments, such a model would become infeasible due to computational overheads. Particularly, considering that manufacturers are consistently pushed towards more customised products with shorter life-cycles, there is an increasing need for manufacturing systems to be responsive to it. Therefore, it is also necessary to develop a fully decentralised model for mobile robots in LSA to ensure self-organisation in more dynamic environments than the hybrid model can handle. This would enable to compare the models and determine in which environments either one should be used. Thus, the second hypothesis is: **The agent behaviour model for self-organising mobile robots in LSA based on the hybrid architecture will exhibit better self-organisation schedules but lower responsiveness than the model based on the decentralised architecture.** Considering the current level of dynamics in LSA, the hybrid model should be justifiable due to optimisation and responsiveness. The responsiveness is expected to be high because disruptions are not very frequent, and the number of deployed manufacturing resources and products is relatively small. However, in the light of a constantly developing domain and dynamic trends in manufacturing (and other industries), its responsiveness may become prohibitive and thus the need for the decentralised model may also arise in the future.

In order to prove or disprove these hypotheses, four knowledge contributions are presented in this doctoral thesis. The core contributions are:

1) A comparison of the manufacturing resilience and utilisation between mobile and fixed automation systems

2) A hybrid self-organisation model for mobile robots in large structure assembly
3) A decentralised self-organisation model for mobile robots in large structure assembly
4) A comparison of hybrid and decentralised self-organisation models for mobile robots in large structure assembly

3.3. Research Approach

The manufacturing systems are compared by running simulations on representative numerical models. By measuring the differences in the performances of like-for-like systems, it is possible to determine in which scenarios the mobile system will be preferred to the dedicated system. Hence, the need arises to develop means to self-organise mobile robots in an effective way.

The typical hybrid and decentralised architectures are illustrated in Figure 3-3 and Figure 3-4 respectively. Firstly, the hybrid self-organisation behaviour model must be developed and verified. In this model, a central blackboard agent receives information from all agents on the shop floor and processes it to determine how to proceed. Secondly, a decentralised self-organisation model must be developed and verified. In this model, there are no central coordinating entities and each agent follows their own interests. Here, the PAs only take immediate decisions and do not look into the future. They offer certain amounts of virtual credits to RAs which then decide whether to accept the offers or not.

The second hypothesis states that the two models will enable self-organisation in both static and dynamic environments of LSA. By static environments, it is meant that the initially set schedule gets very rare changes. Conversely, in a dynamic environment, changes occur frequently and the system must be able to respond to them quickly. As is shown in Chapter 2, the hybrid model should create near-optimal schedules at the expense of potentially high computational overheads. Where the overheads are prohibitively high, it makes sense to use the decentralised system due to its high responsiveness regardless of the environment. The validation in Chapter 8 assesses the performance of both models with respect to a) minimising TWT, and b) increasing the numbers of products and their working contents. The minimisation of TWT is a measure of the scheduling optimality. The responsiveness can become of critical importance when the environment is dynamic.

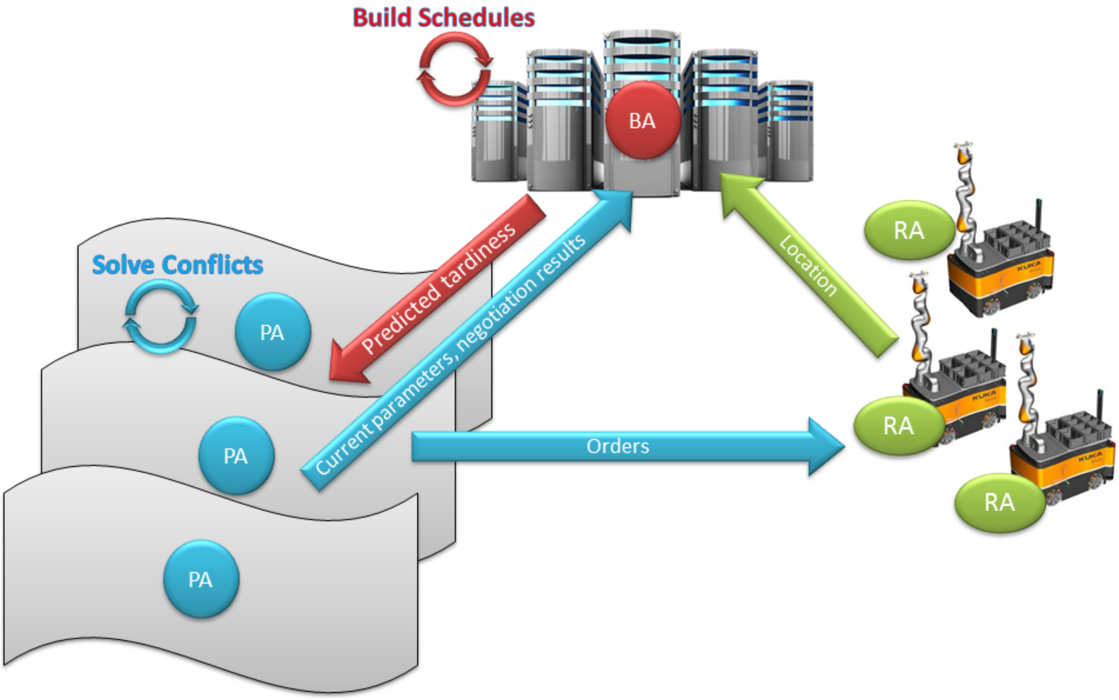


Figure 3-3: The illustration of a typical hybrid architecture in the research approach

For the purposes of this doctoral thesis, the architectures are treated separately. However, in a realistic manufacturing environment, they can be used interchangeably as and when preferred.

The hypotheses are divided into two: stating the expected differences between the system types; and the investigation and comparison of two fundamentally different agent-based self-organisation models.

The completed work consists of four core knowledge contributions. The first contribution identifies the benefits and challenges of using dynamic systems composed of mobile robots instead of fixed automation systems. The knowledge gained from the first contribution is used as a justification for the further work. The second contribution is the development of the hybrid behaviour model for use by mobile robots in LSA (Chapter 5 - Chapter 6). The third contribution is the development of the decentralised behaviour model for use by mobile robots in LSA (Chapter 7). The final contribution is the validation and comparison of the two developed models in Chapter 8.

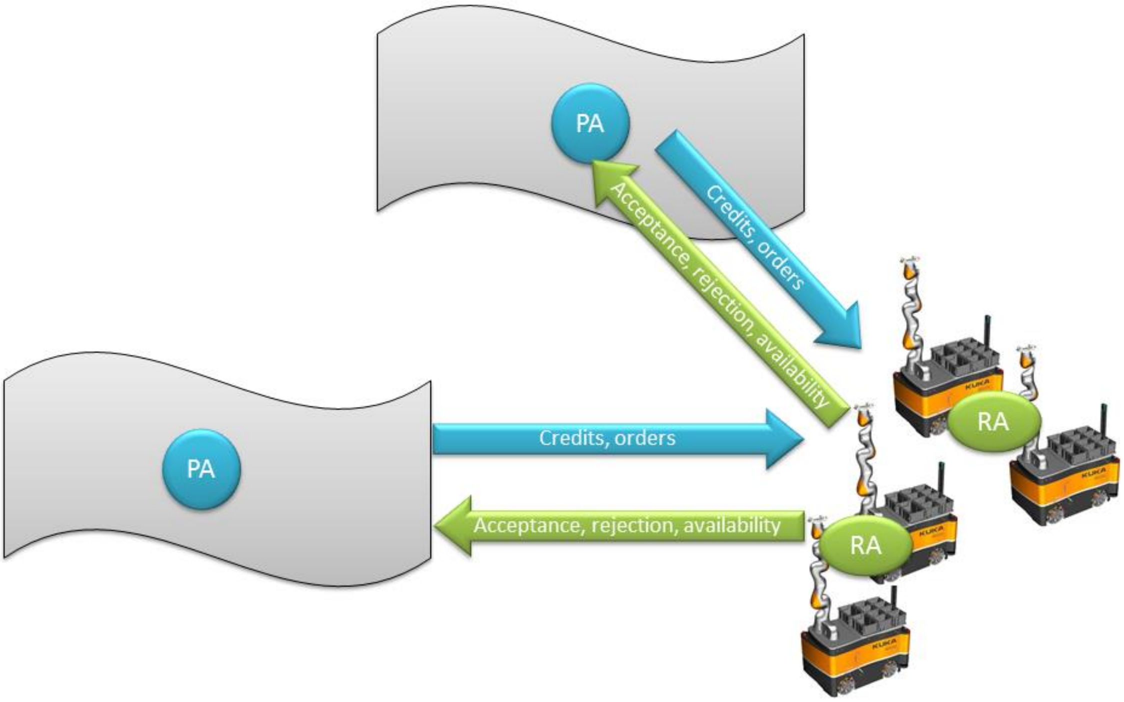

PA
Credits, orders
Acceptance, rejection, availability
PA
Credits, orders
Acceptance, rejection, availability
RA
RA

Figure 3-4: The illustration of communication links in the decentralised model in the research approach

The stated hypotheses allow defining the validation procedures for the knowledge contributions. As there is a very wide range of possible scenarios occurring in the given research problem, it is unwise to validate through them all. Instead, some of the typically arising types of problems were used.

The development of a hybrid behaviour model for mobile robots in static environments of LSA is split into two chapters in this thesis (Chapter 5 and Chapter 6). This is because the hybrid self-organisation model required a priority aging policy, as discussed in section 2.3.1.5. The reader is referred to those chapters for further details. The next sections in this chapter clarify the purpose and approach to the core knowledge contributions.

3.3.1. The Identification of Differences between the Application of Fixed and Mobile Automation Systems in Large Structure Assembly

The absence of relevant literature on this topic leaves too many open questions to be able to proceed with the development of the self-organisation models. What exactly are the differences with the more dynamic, mobile approach? What changes when resources move to products instead of the products to resources? How much benefit does product-centric scalability bring? How much availability is compromised due to moving? Michalos, et al. [45] have stated the advantages for the automotive manufacturing domain. However, no literature was found to either quantify these advantages or approach the questions with product-centric scalability in mind.

In this core contribution, a like-for-like comparison approach is adopted where the two systems with identical manufacturing capacities are compared to one another. The experiments include testing the systems' utilisation proportions and resilience parameters under various product sizes, product mixes and rush order arrival frequencies. The only difference between the systems is that the dedicated system has a limited working envelope (due to gantry rails) and the mobile system does not.

The comparison study firstly provides an understanding of when either type of system performs better and secondly provides a foundation for the development of the self-organisation behaviour models. The assessed characteristics between the systems are linked to the considerations that are likely to have an economic impact due to the manufacturing systems' performances: the utilisation, ability to control product delivery times and resilience. Utilisation is a measure of how well an investment is being used, and resilience and control over product delivery times enable a system to minimise TWT better.

3.3.2. The Investigation of a Hybrid Agent Behaviour Model for Mobile Robots in LSA

The investigation of a hybrid self-organisation behaviour model for mobile robots is the second core contribution. This model is intended to achieve the lower TWT out of the two. Due to having a central blackboard agent, this model is expected to achieve near-optimal results, but experience exponentially increasing computational overheads with increasing numbers of agents on the shop floor. Nevertheless, it was expected that this model should be sufficient for the self-organisation of mobile robots in the current scenarios of LSA. This is because there it is currently very uncommon to have large numbers of active products and machines at any one time on a shop floor. However, it is possible that similar applications will arise in the future which would require processing high numbers of agents and/or responding to disturbances more frequently than sensible for this model. Therefore, it was important that a decentralised self-organisation model is developed for that purpose as well.

It is stated in Chapter 2 that no academic literature was found to investigate problems with pre-emptive and scalable jobs with the objective of minimising TWT. For this purpose, it makes sense to consider approaches from related fields of science. A potentially suitable approach was found in computer science. There, processors with several cores must prioritise which tasks to process first. A very common problem with this approach occurs when higher-priority tasks get launched before low-priority tasks get processed. This results in some tasks never being processed. The solution to this is to gradually increase the priority values of all tasks until they get processed. This process is called “priority aging” and the policy that governs the aging is the “priority aging policy” (PAP). Because the computational approaches do not have strict due times to follow, new PAPs had to be proposed and investigated in Chapter 5. There, a PAP is determined that consistently achieves the best and often even optimal TWT values. To ensure that the sub-optimal schedules are reprocessed, a negotiation protocol and its triggering mechanism are developed in Chapter 6.

3.3.3. The Investigation of a Decentralised Agent Behaviour Model for Mobile Robots in LSA

Assuming that a decentralised behaviour model is more adaptable to changes in schedules and scaling up production, it is an important complement to the hybrid model. Thus, this model is the third knowledge contribution. Its disadvantage is that it does not guarantee as good results in terms of minimising TWT, because no single agent has full knowledge of the environment. In a real application, the mobile system should be able to shift between these two models based on immediate needs.

To govern the behaviour of agents in this decentralised architecture, an economic model has been applied. The PAs start with a bankroll of credits which is proportional to their tardiness cost. They offer credits to RAs for processing their tasks. The RAs seek to maximise their earnings. Therefore, they accept the highest-offering contracts. They also apply a movement penalty, because they cannot do value-adding work during moving. The full details of this model are presented in Chapter 7.

3.3.4. The Comparison of Self-Organisation Models

This knowledge contribution is presented in Chapter 8. The models are compared to one-another in identical scenarios to confirm the second hypothesis. It is expected that the hybrid model will achieve better results with respect to minimising TWT. However, it is also expected to exhibit lower responsiveness in comparison to the decentralised model.

The models are firstly tested in scenarios with varying kinds of scheduling challenges to assess their optimality with respect to minimising TWT. Secondly, they are tested under varying numbers of agents and product working contents. The interest is in investigating under which circumstances either model should be preferred for use.

3.4. Chapter Summary

In this chapter, the methodology that was undertaken to achieve the research objectives is presented.

The research problem is the fact that the usefulness of dynamic, mobile systems in LSA had not been investigated yet. In case if the usefulness is proven, it is unknown how to self-organise the mobile system in an applicable and effective way.

The research in this thesis is carried out by means of simulations. Firstly, the system types are compared to one another in order to determine the performance differences. And secondly, two agent-based self-organisation models are developed for autonomous task allocations in a range of scenarios. The models are compared in Chapter 8 as part of the thesis validation.

Chapter 4 - A Comparison of the Manufacturing Resilience and Utilisation between Mobile and Fixed Automation Systems

4.1. Introduction

Prior to developing self-organising models for mobile robots, it was necessary to understand the differences between the application of mobile and traditional, dedicated automation systems. In this chapter, an investigation is carried out to investigate under which circumstances mobile systems have operational advantages over fixed automation systems. This comparison study also determines the performance boundaries that can be achieved by controlling mobile robots with unsophisticated control models. The development of the more sophisticated self-organising models, in Chapter 5, Chapter 6 and Chapter 7 is based on the findings of this analysis.

The literature review showed that traditionally, manufacturing plants have been relying on dedicated machinery with fixed infrastructure and little ability for reconfiguration. In large structure assembly (LSA), the state-of-the-art systems are generally very well suited for their purposes. However, the ever-increasing fluctuations in customer demands and difficulty of reconfiguring these systems lead to their early obsolescence. Moreover, the transportation of products for these systems, which is done by cranes, is slow and requires highly skilled labour.

From an economic perspective, major problems with traditional, dedicated automation systems are their limited movement ability and the inability to scale up the work rate on any individual product. For this reason, in times of low demand, it may often occur that only a small proportion of the system's capacity is utilised.

An alternative approach which would overcome this problem is seen in the use of mobile robots in such environments. Firstly, mobile robots are much more reconfigurable than the dedicated automation systems and can stay in service longer. This is due to using standardised components and the inherent flexibility of the mobile manipulators. Secondly, by moving manufacturing resources to products instead of the vice versa, the need to transport products is reduced. Thirdly, with large enough products, it should be possible to scale up the work rate on any individual product by allocating several mobile resources to work on the same product at the same time.

As described in the literature review, one major challenge that currently separates this approach from becoming a reality is the inability of mobile robots to physically meet the working standards of the dedicated automation systems. The mobile platform reduces the structural stiffness of mobile robots, leading to difficulties in carrying out tasks accurately. Furthermore, the current localisation methods of mobile robots are far from being as accurate as carefully calibrated fixed robots. This further reduces the positioning accuracy of mobile systems.

Finally, there is the challenge of how to arrange the decision-making of the mobile system for the purposes of self-organisation. No such behaviour models were found to be published specifically for this problem. For this reason, it is important to firstly identify what the differences between the two considered systems are, assuming both are capable of working at the necessary specifications.

The problem considered in this work is an alteration of the common minimisation of total weighted tardiness (TWT). The alteration is the product-centric scalability. This means that several mobile robots may fit around the same product to scale up the individual work rates. An important specification is also the fact that the jobs are pre-emptive, meaning that any job can be paused and later resumed if necessary. This is justified, considering that a common job in aircraft wing assembly is the drilling and riveting of very large quantities of holes on the same products.

The key parameters of interest are tardiness ratios between the systems, rush order completion time ratios, production loss, the utilisation, moving and idling proportions of time. The tardiness ratio between the systems is a measure of how much time the systems spend in tardiness in relation to one another. In this context, tardiness is created by a disruption. The rush order completion time ratio is a measure of how quickly the systems can complete an identical rush order in relation to one another. The production loss quantifies how much working content has been lost due to a disruption. The proportions of time spent in any activity are self-explanatory and help compare the systems' utilisations.

4.2. Methodology

In order to establish the behavioural characteristics of the systems based on different parameters, Monte-Carlo simulations are used. The experiments were designed to be able to analyse how either system responds to rush orders, varying product arrival times, production mix variations in the production process and different levels of spare capacity. A rush order is an order that has been launched out of schedule and is often treated in literature as a form of disruption [60]. By variable arrival times, it is meant that there is an unsteady flow of products into the system. A production mix variation means that the launched products have different working contents.

4.2.1. Problem Formulation

Two kinds of products are launched to the systems in this chapter: regular and rush orders. A regular product is a product that arrives on schedule and does not have priority over others. The manufacturing systems consider working on this once they become available. A rush order is a high priority product that is launched out of schedule and is required to be completed as soon as possible. Thus, if the rush order is within the working envelope of any resource agents, they move to process the rush order until completion and then continue as before. The behaviour models of both systems are described in section 4.2.2.

Each product is launched with a working content at a given moment in time. Once the content has been processed, the product is considered ready and gets unloaded. The simulation model is subject to the following constraints:

$S_j \geq 0, l_j \geq 0, u_j \geq 0, l_j' \geq 0, u_j' \geq 0 \quad \forall J_j \in J$ (1)

$|r_j'_{min}| = L_j + p_j \quad \forall J_j \in J$ (2)

$S_j \geq l_j' \quad \forall J_j \in J$ (3)

$C_j = S_j + p_j \quad \forall J_j \in J$ (4)

$u_j \geq C_j \quad \forall J_j \in J$ (5)

$m_{j\ max} = 1 + Y * (m - 1)$ (6)

$m = f$ (7)

$(l_j,l_j') =]u_j,u_j'[, (l_j,l_j') =]l_{j+1},l_{j+1}'[\quad \forall J_j \in J$ (8)

$(u_j,u_j') =]l_j,l_j'[, (u_j,u_j') =]u_{j+1},u_{j+1}'[\quad \forall J_j \in J$ (9)

Where:

$$Y = \begin{cases} 0, & \text{for the dedicated system} \\ 1, & \text{for the mobile system} \end{cases}$$

The first constraint ensures that no activity can start before the simulation. Constraint (2) specifies that the minimum duration of every job $|r_j'_{min}|$ is the sum of the time taken to load (L_j), process (p_j) a job J_j. Constraint (3) defines that a job can only start being processed at time S_j once it has finished loading at time l_j'. The completion time C_j in constraint (4) is the sum of the starting time S_j added to the processing time p_j for each agent. Under constraint (5), for each job J_j, the unloading may be started at time u_j only as soon as the processing on that product has been finished at time C_j. Constraint (6) ensures that the maximum number of resources $r_{j,\ max}$ that can be allocated to processing any single job J_j is 1 for the dedicated system and *m* for the mobile system. The deployed number of resources (and hence the maximum values of MD_f and MM_m) have been equalled to 4 under constraint (7). The crane's availability is defined under constraint (8). It establishes that between the start l_j and finish l_j' of loading job J_j, there can be no unloading (u_j, u_j') or loading of other jobs (l_{j+1}, l_{j+1}') and vice-versa under constraint (9). The crane system is a shared resource on the shop floor that can be unavailable.

A large number of simulations were required to be carried out for this part of the thesis. Out of the available simulation platforms, NetLogo (version 5.3.1 [120]) was found to be the best choice due to the ease of setting up and repeatedly running a model [121]. Other than being able to carry out many simulations in a short space of time, this software package is also capable of measuring a wide range of parameters that are taking place in simulations. The disadvantage of using NetLogo is the difficulty in facilitating intelligent decision making. However, it was established that the control models in the simulations used for this initial research stage were not advanced enough to cause this challenge and therefore the desired results could still be extracted.

4.2.2. Decision-Making Policies

In this section, the three decision policies that are used by the systems in these simulations are described. They are all based on the First-In-First-Out [35] policy.

1) Dedicated system: As shown in Figure 4-1, the dedicated system applies the First-In-First-Out policy under normal circumstances. If a rush order arrives, then the system immediately switches over to that until completion. Only one dedicated system unit can be allocated to a rush order and therefore the remaining ones keep working on their regular products (if applicable).

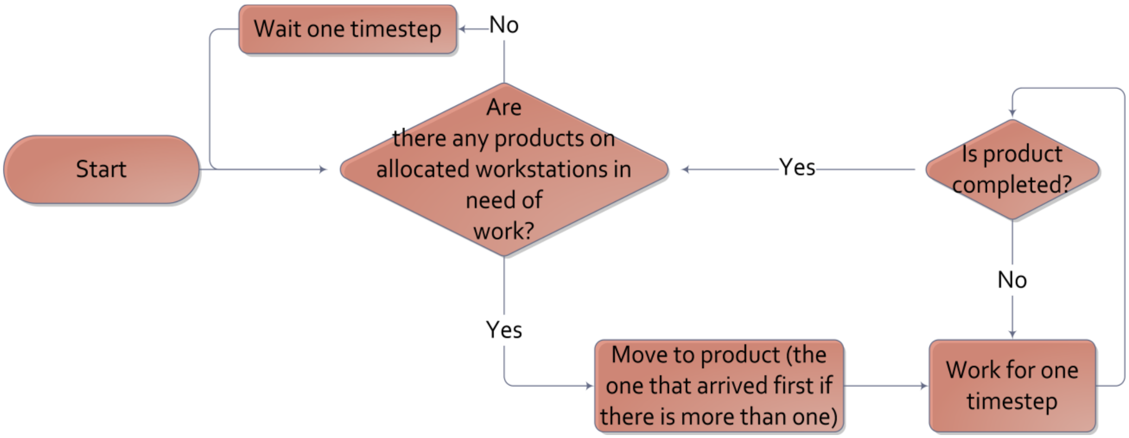

Wait one timestep
No
Start
Are there any products on allocated workstations in need of work?
Yes
Is product completed?
No
Yes
Move to product (the one that arrived first if there is more than one)
Work for one timestep

Figure 4-1: The flowchart for the dedicated system's decision policy

2) Mobile robots (non-cooperative): The flowchart for this behaviour is shown in Figure 4-2. Each mobile resource is assigned two adjacent workstations (for analogy with the dedicated system) in normal circumstances and applies the First-In-First-Out policy. If a rush order arrives then all the mobile resources leave their positions immediately in order to complete the rush order. If no job is available at either allocated workstation of a mobile resource, then it returns to the waiting area.

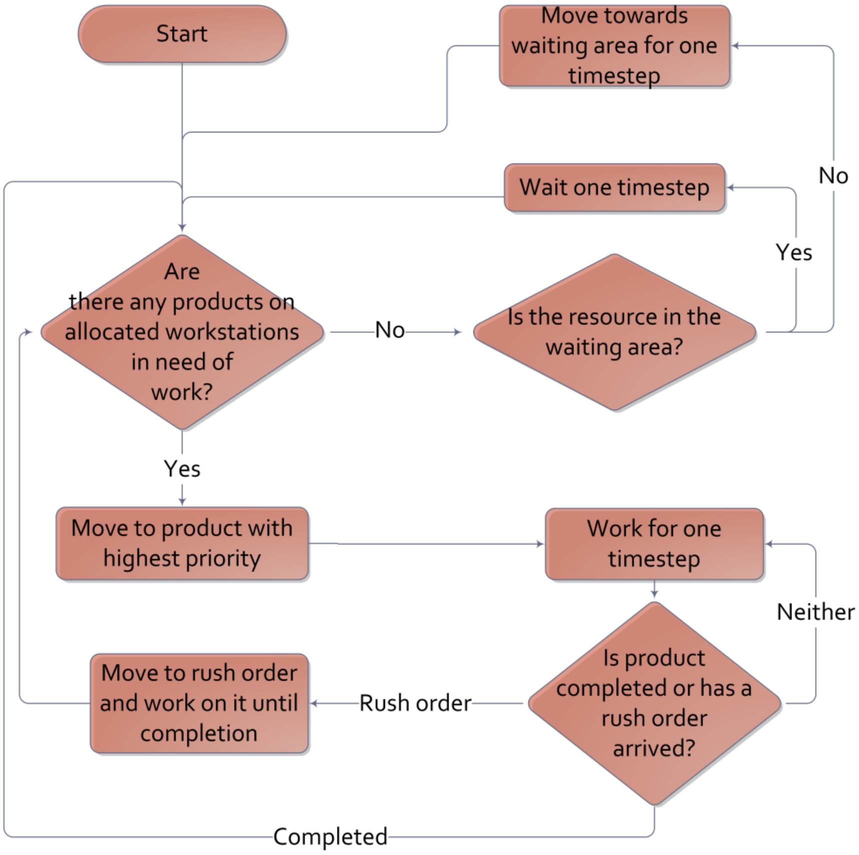
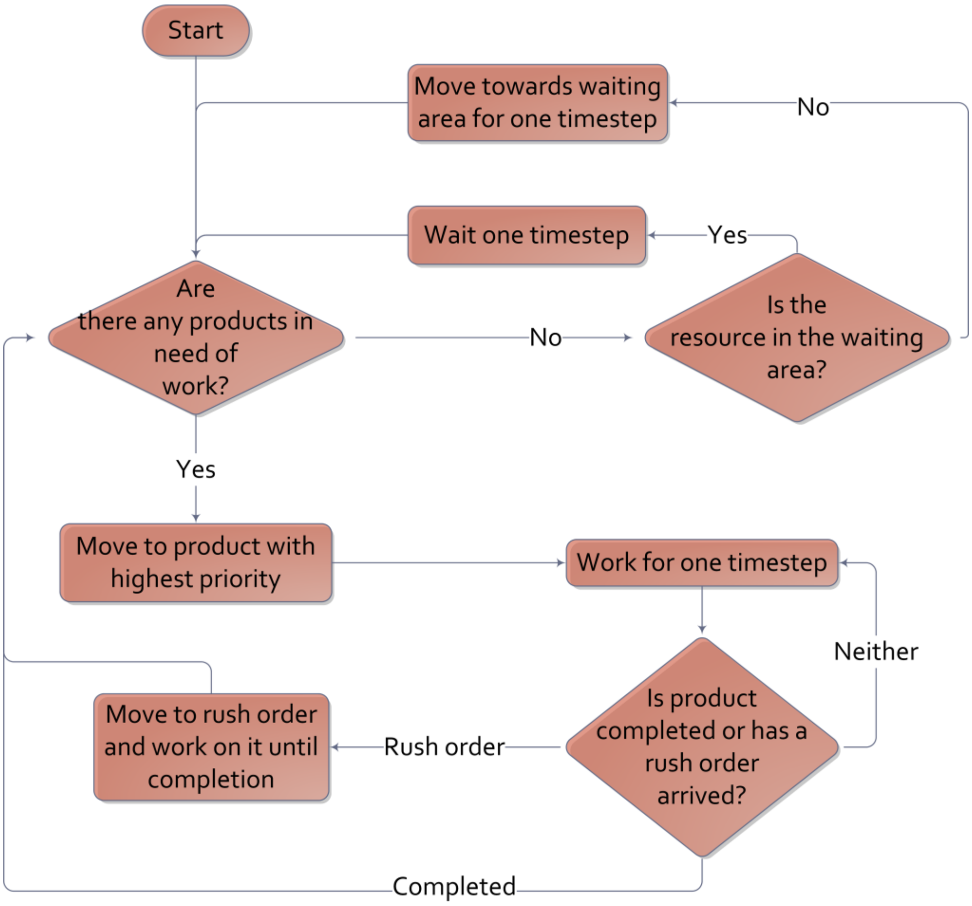

Start
Move towards waiting area for one timestep
Wait one timestep
No
Yes
Are there any products on allocated workstations in need of work?
No
Is the resource in the waiting area?
Yes
Move to product with highest priority
Work for one timestep
Neither
Move to rush order and work on it until completion
Rush order
Is product completed or has a rush order arrived?
Completed

Figure 4-2: The flowchart for the non-cooperative decision policy of the mobile system

3) Mobile robots (cooperative): As can be seen in Figure 4-3, this model is based on the non-cooperative policy, but instead of returning to base, each mobile resource looks for products on all of the shop floor's workstations, not only the ones allocated to them. This way, the mobile resources attempt to cooperate with one another, but may lose time due to excessive moving. They return to base only in the case where there is no available work at all.

The non-cooperative policy was designed to be passive and mimic the dedicated system as much as possible. This is to highlight the effect of scalability alone. On the contrary, the cooperative policy was designed to be very proactive. The two extremes of proactivity were expected to highlight the differences to the highest extent.



Start
Move towards waiting area for one timestep
No
Wait one timestep
Yes
Are there any products in need of work?
No
Is the resource in the waiting area?
Yes
Move to product with highest priority
Work for one timestep
Neither
Is product completed or has a rush order arrived?
Rush order
Move to rush order and work on it until completion
Completed

Figure 4-3: The flowchart for the cooperative decision policy of the mobile system

4.2.3. Simulation Model Specifications

The simulation model is governed by the flowchart shown in Figure 4-4. The simulations start with half of the workstations loaded with products. The model then considers the appropriate product launch policy (shown in Equations 4.1-4.4 below) to determine when a new product should be launched to the systems.

When a product is launched, it seeks to be loaded on an available workstation. In a situation when no workstations are available the launched regular products are sent to a queue. If the launched product is a rush order, it triggers the systems to unload a waiting product in order to free up a workstation and enable the loading of the rush order onto it.

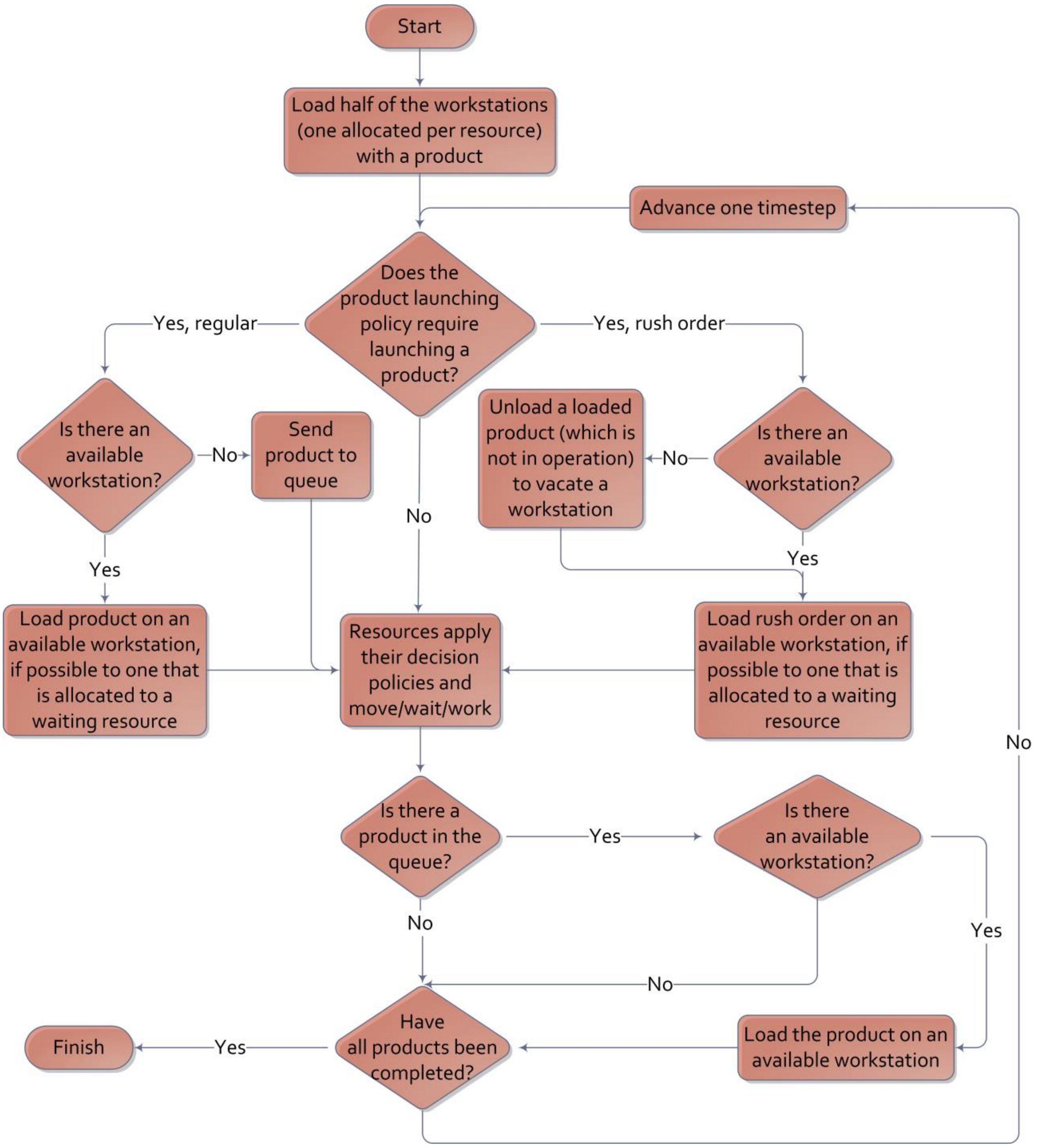

Start
Load half of the workstations (one allocated per resource) with a product
Advance one timestep
Does the product launching policy require launching a product?
Yes, regular
Yes, rush order
Is there an available workstation?
No
Send product to queue
No
Unload a loaded product (which is not in operation) to vacate a workstation
No
Is there an available workstation?
Yes
Yes
Load product on an available workstation, if possible to one that is allocated to a waiting resource
Resources apply their decision policies and move/wait/work
Load rush order on an available workstation, if possible to one that is allocated to a waiting resource
No
Is there a product in the queue?
Yes
Is there an available workstation?
No
Yes
No
Finish
Yes
Have all products been completed?
Load the product on an available workstation

Figure 4-4: The general flowchart for the simulation model

In the next step, the systems apply their decision policies, as described in section 4.2.2. After this, the simulation model checks whether there are any products in the queue. If so, the products are loaded on to the available workstations. If workstations are not available, then the queued products keep waiting.

In each simulation run, the number of deployed units is the same between the systems. For example, in a simulation run with 4 mobile resources (8 workstations), there are 4 units of the dedicated system (also 8 workstations) to match the working capacity. In the experiments, there are 2-4 deployed units. These amounts were seen as suitable to assess systems with multiple machines, but without causing unreasonable computational overheads (as determined from test runs).

By a deployed unit it is meant that a mobile resource with an identical working capacity is deployed per every unit of a dedicated system. A unit of a dedicated system can be considered a copy of a state-of-the-art system like the ElectroImpact E6000 [6]. Individual capabilities of each unit are considered equal. Each unit of the manufacturing systems works at a rate of one unit of work per time step (minute) of simulation, provided that it is located at the workstation. The mobile robots also have a base to return to in case if there were no tasks available for them. The time spent in it is registered as waiting time.

The steady-state of production is defined as the state when only regular products are launched into the systems. The launch of a rush order is considered a disruption that causes the systems to divert some of their working capacities away from the regular products in order to complete it. Consequently, the systems may then be in underproduction with respect to regular products, because the committed work capacity to them becomes lower than normal. If this occurs to a system, then it is considered to be in underproduction.

An important measure when responding to disruptions is the manufacturing resilience of the systems. It is a measure of a system’s ability to mitigate or absorb a disruption and return to normal operating conditions. Gu, et al [59] define three important parameters for it: the production loss, the total underproduction time and the total settling time. The production loss (PL) is a measure of how much work content the system has done less than what was initially planned for the given instance in time. The total underproduction time (TUT) is the time that a system has spent in tardiness due to a disruption. Even after completing a rush order, the system is still in a tardy state and must commit its full capacity to recover from it. The total settling time (TST) is measured in a system where there are buffers between stages of assembly. Due to considering only a single generic stage in this chapter, the TST is omitted.

The workstations are arranged in two rows and x columns for the mobile system, where x is the number of deployed units of resources. Each of the dedicated system's deployed units is allocated two workstations. Due to the product sizes in some extreme cases of large structure assembly, it was important to stress the moving distances of mobile robots. Each workstation is placed 60m away (according to the Airbus A380 model's design, an aircraft's wing panel can be over 40m long) from adjacent ones. In this layout, assuming 1m/s for mobile robot movement speed, it takes a minute for the shortest possible move between workstations for a mobile robot.

Two product launching policies are considered in this chapter: the regular and the fluctuating.

When the regular product launching policy is selected, the products are launched at regular intervals so that the launched working content is equal to the system's capacity minus spare capacity. Equation 4.1 shows the formula that governed the product launch intervals in this case. The spare capacity is used as a fraction and C is the working capacity in the simulation.

$$t = \frac{Average\ product\ work\ content}{(1 - spare\ capacity) * C} \qquad \text{(Equation 4.1)}$$

If the fluctuating launching policy is selected, then at each time step there is a given probability that a product will be launched. Similarly to the regular product launching, Equation 4.2 averages the incoming work content at the system's capacity minus spare capacity.

$$P = \frac{(1 - spare\ capacity) * C}{Average\ product\ working\ content} \qquad \text{(Equation 4.2)}$$

The frequency of launched rush orders is established as a rush order to normal product input ratio, denoted by “ж”. Its numerical value denotes per how many regular products is a rush order launched. From Equation 4.3, it might appear that the rush order is a regular occurrence and therefore should not be considered a random disturbance. However, the decision policies and capacities of both manufacturing systems were not designed to take it into account as such.

$$t_{ж} = t * ж \qquad \text{(Equation 4.3)}$$

The interval between rush order launches is the interval of a regular product multiplied by this ratio. For example, if a regular product is launched every 1000 seconds and ж is 10, then rush orders are launched every 10,000 seconds.

In the case of fluctuating product launches, the probability of regular product launch is divided by the ratio, as seen in Equation 4.4. For example, if the probability of regular product launching is 0.001 and ж is 10, then the probability of a rush order launch at each time step is 0.0001.

$$P_{ж} = \frac{P}{ж} \qquad \text{(Equation 4.4)}$$

Because the amounts of time required to transport and process different tasks can be very different depending on the specific nature of the production systems and used technology, this study introduces a ratio $\mathcal{R}$ (shown in Equation 4.5 and Equation 4.6) to express the relative time that it takes the crane system to load and unload a product in relation to the average working content W_j that the reference machine requires to complete all the operations on a job J_j.

$$\mathcal{R} = \frac{(\mathrm{Lj} + \mathrm{Uj}) * \mathrm{f}}{\mathrm{Wj}} \text{ for the fixed system, and} \qquad \text{(Equation 4.5)}$$

$$\mathcal{R} = \frac{(\mathrm{Lj} + \mathrm{Uj}) * \mathrm{m}}{\mathrm{Wj}} \text{ for the mobile system} \qquad \text{(Equation 4.6)}$$

Therefore, an increase in loading and unloading times or scaling up either system causes $\mathcal{R}$ to increase and an increase in product working contents cause $\mathcal{R}$ to decrease. It can be calculated that if supplying each resource with a job and unloading it afterwards takes more time than processing the average job J_j, then $\mathcal{R} > 1$ and the crane system *CS* is a bottleneck in the production process. Conversely, if L_j and U_j are low in comparison to the average working content of products W_j, the crane system *CS* is able to supply jobs J_j to machines MD_f and MM_m faster than they get processed. This occurs at $\mathcal{R} < 1$.

Finally, it is sensible to assume that increasing numbers of mobile robots around the same product cause them to interfere with one another and result in a loss of efficiency. Therefore, an interference factor *IF* is introduced for the mobile system as well. The reduction of the work rate on any specific job J_j is shown in Equation 4.7. This factor is a measure of the extent to which mobile resources slow each-other down due to spatial issues when processing the same product.

$$WR_j = (1 - IF)^{\mu_j - 1} \times \mu_j \qquad \text{(Equation 4.7)}$$

4.2.4. Assumptions

Manufacturing capacities of both systems are equal. In order to better highlight the differences in resilience, a like-for-like approach is employed where the mobile system's manufacturing capacity is equal to that of the dedicated system.

Resource reliability and quality are equal. The quality of the assembly processes of both systems is compliant with the requirements of the given tasks. Disruptions like maintenance, breakdowns, accidents, etc. have been excluded.

Both systems take negligible time to locally localise. For any automated manufacturing process, it is common for equipment to go through the local localisation process in order to be able to carry out work accurately. The local localisation is the process where the resource identifies its precise location and adjusts (if necessary) after roughly arriving (within 1m in this work for mobile robots) at its destination. This simulation incorporates that into the production capacities as described above. This should not be confused with the movement penalty that exists for the mobile system. Whilst the fixed system moves from one product to another instantaneously, mobile resources can only move at 1m/s.

The mobile system is scalable. Other than mobility, this is the second key difference between the two system types. By scalability, it is meant that several mobile resources can concentrate around one product in order to combine the work rate at it. This is convenient in situations where a rush order has been launched or when there are more mobile resources than products currently available. While it is clear that only a finite number of mobile robots can fit around a physical object; in this model, any number of mobile robots is permitted to attend any product for simplicity.

The rush order is due immediately. Both systems aim to complete the rush order as soon as possible in order to minimise its completion time. If more than one rush order has been launched, the mobile system interprets the one with the earlier launch time as the only rush order until completion.

Rush orders always trigger pre-emption. In reality, it may not always be wise to leave a product for a rush order. This is particularly true when there is a very small working content left to process on a single product. However, for the purposes of this work, pre-emption is always triggered by a rush order.

4.2.5. Experimental Objective and KPIs

The objective of the simulations is to assess how various scenarios in large structure assembly affect a number of Key Performance Indicators. These indicators as percentages of time are:

Utilisation. This is a percentage of how much of its time a manufacturing system spends processing products.

Waiting. This is a measure of how much time a manufacturing system is waiting for work.

Movement. This is a measure of how much time a system spends in movement. It applies only to the mobile system, as described under the assumptions. It should not be confused with the assumption of negligible localisation, which is the process of accurate positioning once the mobile robot has moved to the necessary location.

Time spent in tardiness. Rush orders tend to disrupt the manufacturing systems to such an extent that causes the regular products to be tardy. With the help of spare capacity, manufacturing systems can recover from a tardy state.

4.3. Design of Experiments and Results

In this section, the design and purposes of the experiments within this chapter and their results are explained. There were 6 experiments designed. Experiments 1-4 assume instantaneous product loading and unloading to analyse how the systems perform without that bottleneck in the supply. Experiment 1 was designed to analyse how the mobile system utilises its mobile resources at both proposed behaviour models. Experiment 2 is carried out to compare the abilities of the systems' to control product delivery times when processing a product mix. Experiments 3 and 4 were designed to be able to analyse the effects of rush order launching. For experiments 5 and 6, a crane system (CS) is introduced. It can only load or unload a single product at a time. Experiment 5 is set up to evaluate how much utilisation of the dedicated automation system is reduced due to having the crane system bottleneck. Whereas, in experiment 6, the control over product delivery times with a crane system bottleneck and different product mixes is assessed.

4.3.1. Experiment 1

This is the only experiment that compares the selected decision policies and not the system types. It is carried out to analyse how increasing or decreasing the working content W_j affects the proportions of value-adding and non-value adding activities. The values for this experiment are shown in Table 4-1. The highest W_j for this was only 24,000s, which is 6 hours and 40 minutes of working content; however, it is known that some aerospace applications can have much a higher W_j value than that. For example, a wing panel with a requirement for 100,000 holes can be completed in more than 4 days by the ElectroImpact E6000 (at 16 holes per minute) [6]. Due to the high computational cost of carrying out experiments at the top end of the given range, it was decided to limit the value of W_j to 24,000s.

The measured outputs for both decision policies are the time proportions of the mobile system in waiting, moving and utilisation. Because the decision policies represent opposite extremes in terms of proactivity, it is useful to consider these results as benchmarks. It was expected that the cooperative decision policy will spend much more time moving and less waiting than the non-cooperative policy. It was difficult to predict the differences in utilisation proportions.

Factor	Value	Levels
m, f	2, 3, 4	3
Decision-making policies	Non-cooperative, cooperative	2
Product launching	Regular, fluctuating	2
ж	∞	1
W_j	400; 600; 800… 1800; 2000; 3000; 4000; 5000; 6000; 8000; 10,000; 13,000; 16,000; 20,000; 24,000s	19

Table 4-1: The set of values for the first experiment

The results as percentages of waiting, moving and utilisation for the given product working contents using the non-cooperative policy are shown in Figure 4-5. At W_j = 400s, the mobile robots spend 16% of their time moving. This affects the amount of time they can be actually utilised and prevents them from going to base. Gradually, the wait percentage increases and moving percentage decreases as the W_j value increases. The utilisation proportion rises from 84% at W_j = 400s to 90% at W_j = 800s. It then remains at over 90%

due to starting the setup with half of the workstations already loaded and receiving products at an average working content of 90% of the systems' capacity from there onwards.

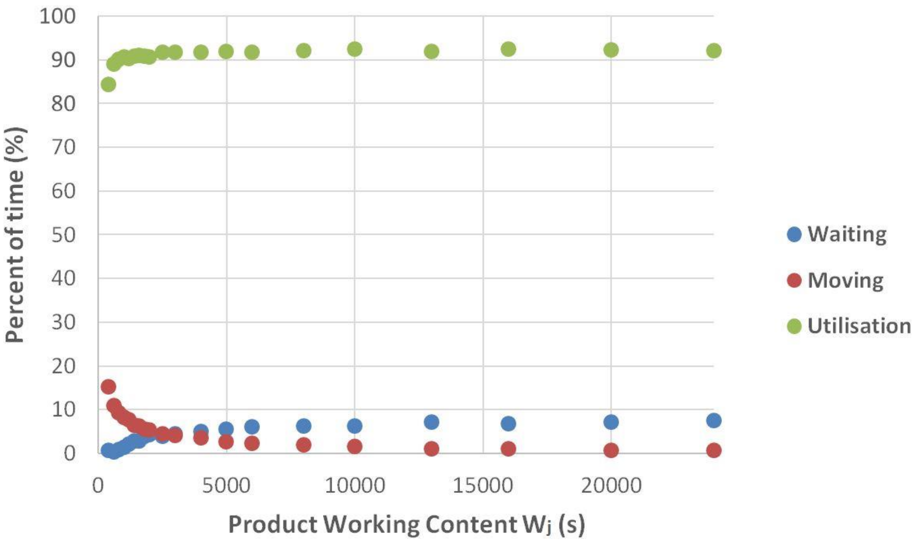


Figure 4-5: The percentages of wait, travel and utilisation of the mobile system (non-cooperative) at different product working contents

As the W_j value increases, the wait time and utilisation rise, and the moving time reduces. At W_j = 24,000s, the moving time was only 0.59%. Considering that in the aerospace industry some products can have a much higher W_j, this result can be considered not high at all. It is a small fraction of the 10% spare capacity and assuming that the trend continues, it will reduce in size at even higher working contents. Certainly, a counterargument is that 0.59% out of a very large amount of time can still be significant and that it would make sense to minimise it on an actual shop floor by optimising the agent behaviour models. However, for the purposes of the work presented in this chapter, it can be considered negligible.

The results with the cooperative decision policy are shown in Figure 4-6. The cooperative policy is much more proactive and movement-intense. No mobile robots returned to base at product W_j < 5,000s.

The key difference is that the more proactive model has led to higher utilisation at every W_j value. There is very little waiting time throughout the experimentation runs. Up to W_j = 5,000s, there is no waiting time at all and it reaches 3% at W_j = 24,000s. The moving time takes a much higher proportion in each simulation run, however, in this case, it is useful to do so. The higher utilisation is proof that moving more is not necessarily a negative factor for the mobile system.

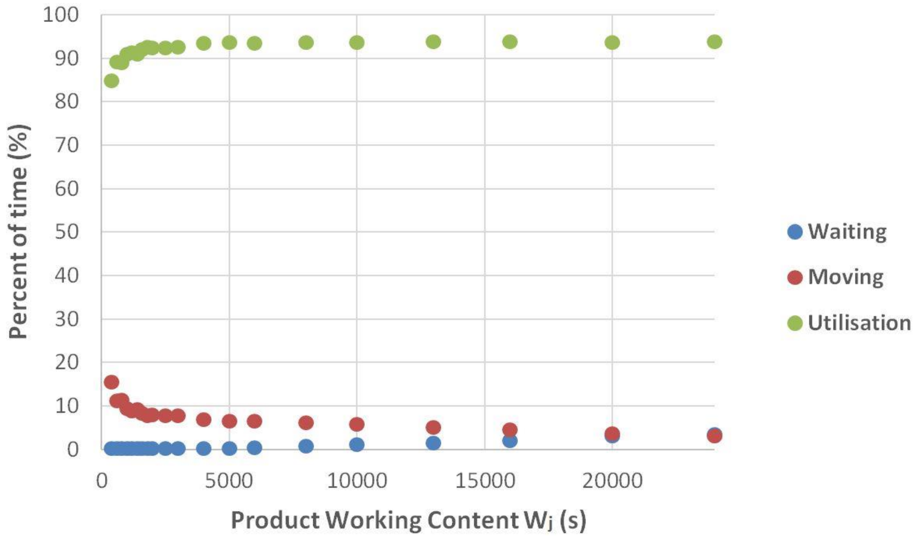


Figure 4-6: The percentages of wait, travel and utilisation of the mobile system (cooperative) at different product working contents

The required moving distances take a proportion of up to 16% of the whole production time at low W_j values. The fact that there is no waiting time and the utilisation is below 90% indicates that the proportion of moving has exhausted the spare capacity, causes tardiness and removes any flexibility from the system. This is because the necessary action of moving causes the inability to do value-adding work as much as required. In such a situation, the mobile system is not capable of recovering from additional tardiness if any disruption occurs. As the W_j value increases, the proportion of moving time reduces because the time to complete a product increases in relation to the moving time. This is because the distances between workstations remain the same. The argument in favour of the non-cooperative decision policy in realistic conditions is the mobile system's ability to occasionally return to base for maintenance purposes.

4.3.2. Experiment 2

The purpose of this experiment is to investigate the performance differences due to product mix variations (but without rush orders). The W_j values of products launched to the systems in this experiment are 10,000s and 40,000s. As there are different W_j values, both systems were expected to complete the smaller one first and then the larger one. The difference in release times r_j was expected with the cooperative decision policy because after completion of the smaller product, work rate would be increased on the larger one. The reason for not using the non-cooperative policy's behaviour is that it mimics the behaviour of the dedicated system in the absence of rush orders and would, therefore, achieve very similar results.

Factor	Value	Levels
m, f	4	1
Decision-making policies	Cooperative	1
Product launching	Regular	1
ж	∞	1
W_j (s)	10,000; 40,000	2

Table 4-2: The set of values for experiment 2

The work progress in time for both systems is presented in Figure 4-7. In this experiment, the cooperative decision policy is selected for the mobile system. At every instance in time, the mobile robots have done at least as much work as the dedicated system. At times when the dedicated system completes work on the smaller product, the work rate (represented by the gradient) reduces because one deployed unit of the dedicated system is not utilised any more. This is where the mobile system is able to scale up the production rate which results in completing the larger product sooner than the dedicated system. The horizontal line for mobile robots represents the waiting time after all available products have been completed. In a real environment, mobile robots could use this time to undergo maintenance tasks or work elsewhere.

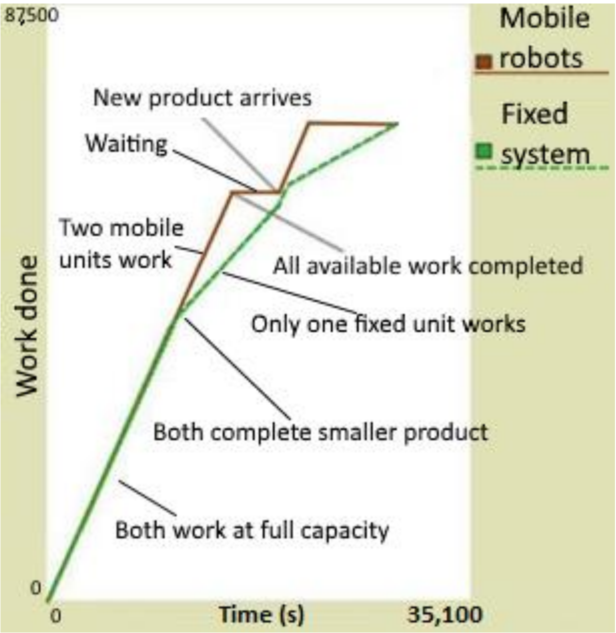


Figure 4-7: The work progress data of the cooperative strategy of mobile robots against the dedicated system with a mix of products

Other than assisting each-other, this also enables the mobile resources to consider temporarily moving to a side project (if it exists) and be utilised there. It was observed that the effect of this increases with an increase in the ratios of product working content. If used appropriately, this advantage of mobile systems should greatly simplify the challenge of releasing products effectively, as described by Thürer et al [122].

4.3.3. Experiment 3

The purpose of this experiment is to investigate the effect of rush orders on the tardiness of both systems at different numbers of deployed units and how quickly they can be completed. A plot was set up to monitor the tardiness of each system in time due to the rush order with deployed resources for both system types $m = f = \{2, 3, 4\}$. The monitored values are proportional, e.g. a 20% PL for $m = f = 2$ was the same as a 20% PL for $m = f = 3$ and $m = f = 4$. The non-cooperative decision policy is applied by the mobile robots in this experiment and only a single rush order's effect in steady-state conditions is examined. The cooperative policy is not used because its' behaviour during a rush order is identical to that of the non-cooperative decision policy and thus it is not sensible to use both.

Factor	Value	Levels
m, f	2, 3, 4	3
Decision-making policies	Non-cooperative	1
Spare Capacity (%)	10, 20, 30	3
W_j	10,000	1

Table 4-3: The set of values for experiment 3

The results of this experiment are shown in Figure 4-8. At *t* = 0s, the system is not experiencing any tardiness and a rush order is launched. In each case (2, 3 and 4 deployed units), the tardiness increases until the rush order is completed. Each maximum represents the completion of the rush order, from that point onwards the systems start recovering towards normal operating conditions. The mobile system always completes the rush order sooner with the same number of deployed units (which is represented by the maxima of each line). However, they also cause a larger proportion of tardiness for the regular products. This is due to allocating all available resources to the high priority rush order and neglecting the regular products. The dedicated system is only able to allocate one unit per system to the rush order and therefore the rest of its units continue working on the regular (low priority) products. For this reason, the gradient of adding production loss is different for the dedicated system, because the proportional loss is different at each number of deployed units. However, for the mobile system, the rate of gained extent of production loss is gained evenly because it loses 90% of the working capacity at every instance in time.

Product-centric scalability makes a clear difference in the control of product delivery times. It enables the mobile system to dynamically relocate the workforce to wherever it is required most. Essentially, the time spent in tardiness (that is when PL exists in the system) is nearly equal for both systems. However, the production loss is more time intensive for the mobile system due to diverting more working capacity away from the regular products for the duration of the time spent working on the rush order. In a realistic scenario with actual due times and more sophisticated self-organisation models, this means that the mobile system has a much better ability to balance its workload in the desired way. On the contrary, there is no chance that workstations of the mobile system are freed during that time. Because the dedicated system keeps working on the low priority products, it is able to free some workstations and allow for new products to get launched. However, this is not an issue for the mobile system as such, because the model chosen for this experiment was not optimised.

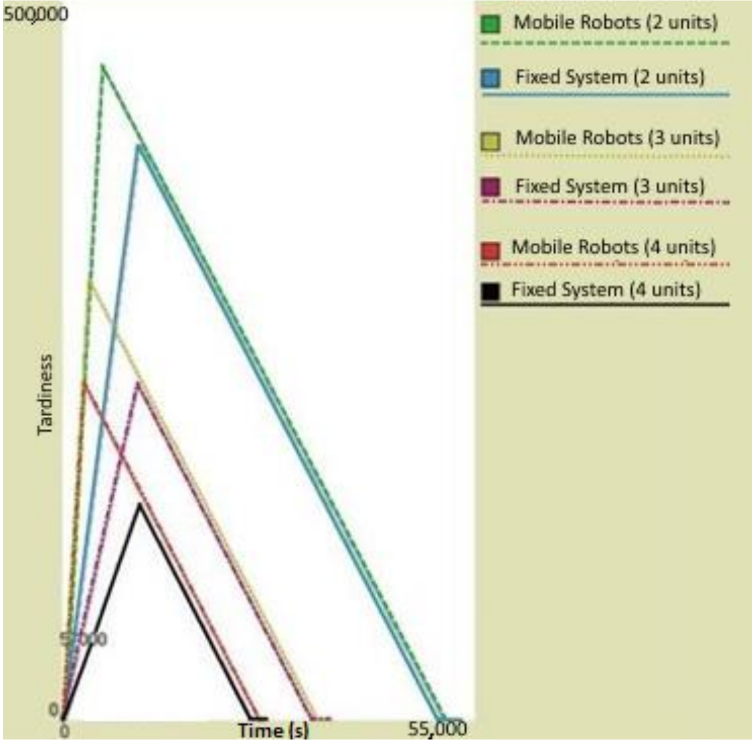


Figure 4-8: The results for experiment 3

Before the rush order launch, both systems were working at an average of 90% efficiency. Once the rush order is been launched, the mobile system diverts 90% of the workload away from the regular schedule due to committing full capacity to the rush order. This is expressed by the upward gradient on the graph. However, the dedicated system loses only 15% of its workload in relation to the regular products at *f* = 4. Instead of committing 90% of all available capacity to regular products, it is now committing 75% (one unit out of four is 25%, this is how much was temporarily allocated to the rush order). For three units, the loss is 23.3% (66. 7% instead of 90%) and for two units it is 40% (50% instead of 90%). The downward gradient on each number of deployed units shows how both systems recover at a uniform rate of 10% due to committing 100% of available manufacturing capacity to regular products while receiving 90% of its value in regular working content. This shows how the mobile system can complete the prioritised jobs earlier than the fixed system.

4.3.4. Experiment 4

This experiment is carried out to analyse the resilience of both systems under different spare capacities. The set of values for this experiment is shown in Table 4-4. As the spare capacity increases, it was expected that rush orders and the consequential tardiness in the system would be completed sooner.

Factor	Value	Levels
Deployed units	4	1
Product launching	Regular, fluctuating	2
Spare capacities	10%, 20%, 30%	3
ж	15	1
Product working contents	10,000s	1

Table 4-4: The set of values for experiment 4

The aim of this experiment is to analyse the resilience of both systems. An Illustration of the resilience parameters (PL and TUT) in a simulation with 10% spare capacity is presented in Figure 4-9. It shows what regularly occurs when a rush order is launched into a steadily working system. On the vertical axis, the designed utilisation is pointed out. In the first part of this experiment, this is 100% - 10% = <u>90%</u>. The dashed areas represent the mobile (shown in green colour) and fixed (shown in red colour) systems' contribution of workload towards regular products. Both systems start processing the rush order at the starting instance when *t* = 0min on the graph. This leads to gaining PL with respect to the regular products due to diverting some manufacturing resources to the rush order. The mobile system completes the rush order sooner due to scaling up the work rate at it. They then return to their original products and work at full capacity to restore the PL.

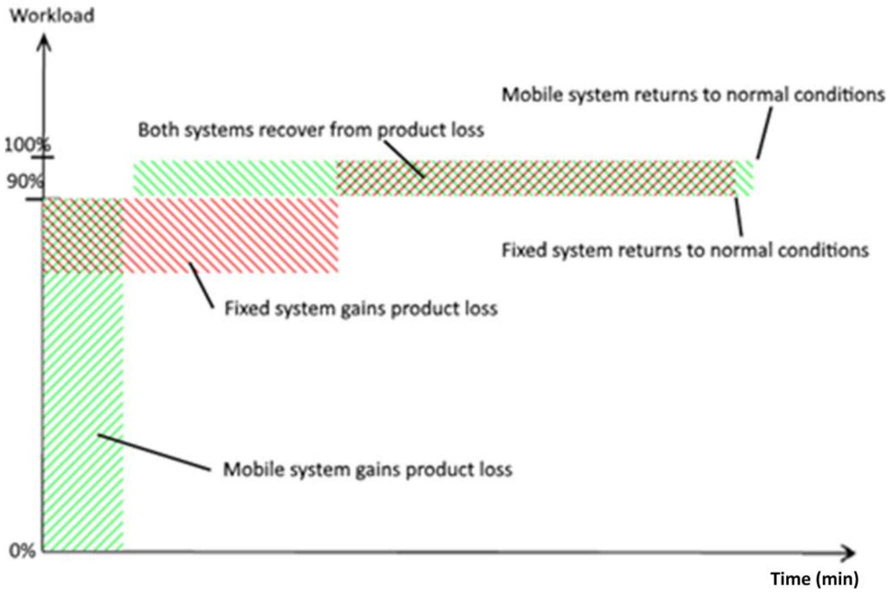


Figure 4-9: The illustration of resilience parameters at 10% spare capacity

Meanwhile, the dedicated system completes the rush order and returns to its regular products. The dedicated system gains PL at a much smaller rate due to only committing 25% of its manufacturing capacity to the rush order, however for the same reason it completes it later too. The major disadvantage for the dedicated system, in this case, is that this is not done by choice. Instead, it is constrained to committing only a single machine to any product. The return to regular production is slightly faster for the dedicated system because less moving must occur for this. The dedicated system then recovers from the tardiness caused by the rush order slightly sooner than the mobile system. The reason for this is the time loss due to travel (all other parameters were equal).

The same situation, but with a 20% spare capacity is illustrated in Figure 4-10. This time, the PL is gained at a much lower rate. The difference with

Figure 4-9 is particularly clear for the dedicated system. In both cases, the dedicated system works at a workload of 75% for the regular products. Therefore, the deficit for the mobile system in the first part of the experiment case is 15%, but in this one only 5% (80% - 75% = 5%).

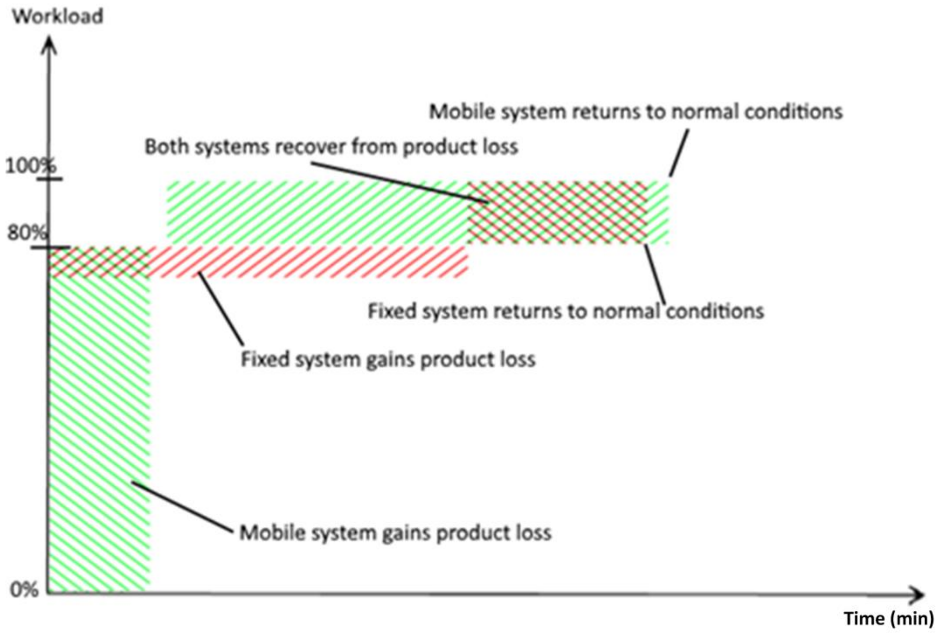

Workload
Mobile system returns to normal conditions
Both systems recover from product loss
100%
80%
Fixed system returns to normal conditions
Fixed system gains product loss
Mobile system gains product loss
0%
Time (min)

Figure 4-10: The illustration of resilience parameters at 20% spare capacity

As a result of this, there is much less PL to restore after the completion of the rush order and the recovery at full workload is also 20% instead of 10%, greatly shortening the time to recover (in comparison to Figure 4-9). Similarly, the mobile system gains PL at a lower rate, but still completes the rush order sooner. This allows it to start recovering earlier but return to normal operating conditions slightly after the dedicated system (again, due to moving).

In the final example, the spare capacity is larger than a unit of deployed resource capacity. This is illustrated in Figure 4-11. It is a special case because the dedicated system never gains any PL. That is because it allocates fewer resources to a rush order than there is spare capacity in the system. It works at 95% capacity because there is no need to make up for any PL at 100%. This is a hypothetical case where there is simply one unit always available for rush orders.

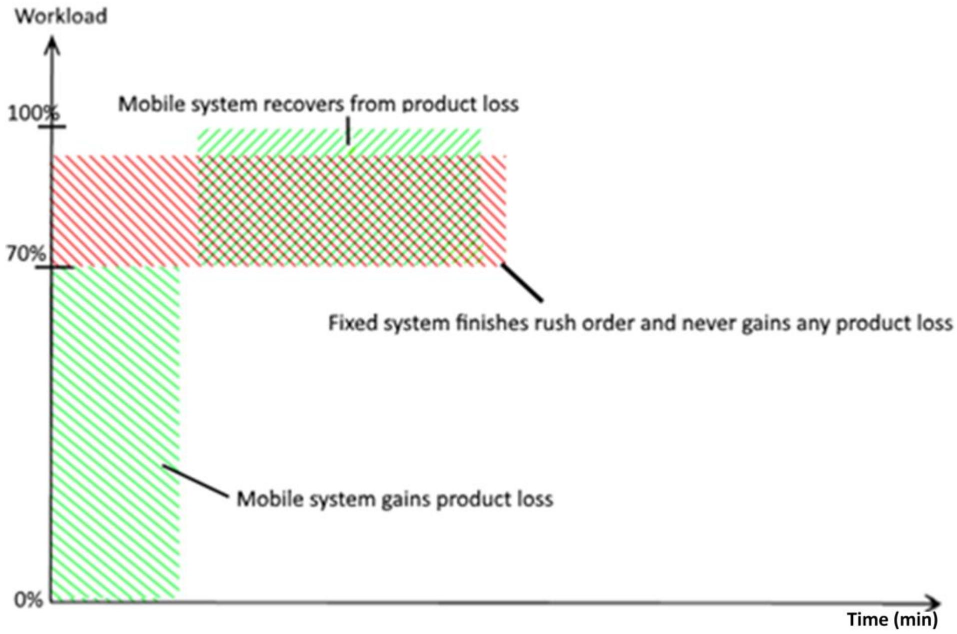


Figure 4-11: The illustration of resilience parameters at 30% spare capacity

The mobile system completes the rush order much earlier, as expected, and then recovers the PL at a workload of 100%. The PL is then eliminated before the dedicated system even completes the rush order. The earlier recovery time is mainly because the mobile system works at a higher workload during the rush order. Therefore, the non-scalability of the dedicated system causes it to lag behind the mobile system.

The first two illustrations in this experiment confirm the two key findings from previous experiments:

- Moving enables much greater control of product release times on the shop floor.
- It lowers the availability of mobile resources because they are unable to do useful work during moving between products.

The third illustration brings about a difficult question: is it better to have no diversion of busy resources at all and complete the rush order later or is it better to complete it sooner and have production loss to some extent? Various published literature on this topic lead to the conclusion that a rush order should only be accepted under conditions where it is considered beneficial for the manufacturer [64], [123]. Therefore, the rush order acceptance criteria are dependent on the manufacturer’s estimations, which are in turn dependent on the manufacturing system’s capabilities. By exhibiting the advantages of product-centric scalability, the mobile system offers more options.

In general, the TUT due to a rush order can be split into three main terms:

$$Total\ time = \frac{Rush\ order\ working\ content}{Manufacturing\ capacity\ at\ rush\ order} + Moving\ time + \frac{Production\ loss}{Spare\ Capacity} \qquad \text{(Equation 4.8)}$$

In the first term, the working content determines how long the regular products will spend in underproduction. The mobile system can influence this time by allocating different numbers of mobile resources to the task. Adding manufacturing capacity to a rush order shortens its completion time. However, it also proportionally increases the rate at which PL builds up. In the second term, the moving time is the inevitable delay that is spent to change the locations of manufacturing resources on the shop floor. It occurs both when resources are moving to the rush order and returning from there. In the third term, the PL is restored to return to normal operating conditions at full workload. The rate at which it is restored is dependent on the spare capacity of the manufacturing system.

4.3.5. Experiment 5

The experimental settings for the fifth experiment are shown in Table 4-5. The varied factors are the ratio $\mathcal{R}$ and interference factor *IF*. The ratio $\mathcal{R}$ is varied from 0.4 until 4. Knowing that the *CS* becomes a bottleneck when $\mathcal{R} > 1$, it is important to test across a range of values several iterations at both sides from 1 in addition to 1 itself. The *IF* is ranged from 0 (no interference at all) to 0.4 in 8 levels, where each consecutive mobile resource slows down the rest of the resources at that job by more than a third. Therefore, simulation runs of 100 products with working contents of over 5.5 hours are analysed in two types of scenarios. Firstly, in scenarios where the CS becomes a bottleneck; and secondly, where scaling up the production rate results in an effectiveness penalty.

Factor	Value	Levels
Working Content of Both Products	20,000s	1
Quantity of Products	100	1
$\mathcal{R}$	0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2, 2.4, 3, 4	12
Interference Factors	0, 0.03, 0.07, 0.1, 0.15, 0.2, 0.3, 0.4	8
Number of Resources per System	4	1
Sample Size	1	1

Table 4-5: The setup for experiment 5

The results of this experiment are shown in Figure 4-12. The plot contains the utilisation percentages for both systems and how much the mobile system's manufacturing capacity can be reduced to maintain an equal throughput with the fixed system at all considered R values. At $\mathcal{R}$ values of up to 1 (shown to the left of red line), the *CS* supplies jobs faster than either system can process. In such circumstances, the dedicated system achieves a ~0.3% higher utilisation on average due to the moving advantages (as described under the assumptions). At $\mathcal{R}$ values of more than 1 (shown to the right of the red line), the supply of jobs becomes a bottleneck, resulting in reduced utilisation rates for both systems. The plot also shows by how much the manufacturing capacity of the mobile system (at different IFs) can be reduced to maintain an equal throughput with the dedicated system.

It can be argued that the unreasonably low utilisation can only be caused by poor management decisions when setting up the shop floor. However, a clear conclusion is made in the literature review that the unpredictability in demand fluctuations and product customisation is steadily rising. This means that periods with low demand or high product handling requirements are very likely to cause low utilisation at some stages of the shop floor's life. For manufacturing plants with large order backlogs, such scenarios may arise when there are problems on the supply side of the logistics, for example.

The reduction is easier to facilitate for the mobile system, because mobile robots may easily be sent to work on other tasks, for example. For dedicated, fixed automation systems such an option is not practical, because it is difficult to reconfigure them to carry out other tasks and their repositioning is much more complicated as well.

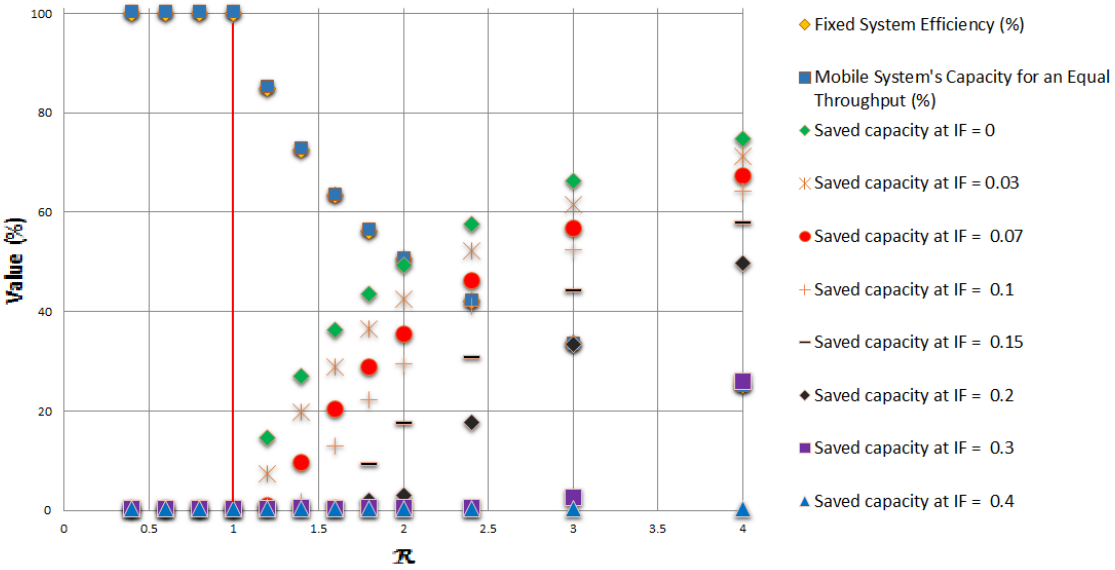


Figure 4-12: The results for experiment 5

4.3.6. Experiment 6

This experiment is done by varying the product mix at different $\mathcal{R}$ values. The product mix consists of Products A and B. The working content for product A is constant at 10,000 seconds of working effort, while that of product B is varied. The effects of how different product A to product B ratios affect the tardiness of the systems at different $\mathcal{R}$ values are examined. The due times of all products are set at the average of the working contents of the products after starting times ($D_j = S_j + C$). If a system exceeds its job due time ($F_j > D_j$), it is penalised by one penalty point per time step per product. The experimental setup for this experiment is shown in Table 4-6.

Factor	Value
Working Content of Product A	10,000
Working Contents of Product B	2000; 4000… 18,000; 20,000
$\mathcal{R}$	0.8, 0.96, 1, 1.04, 1.12, 1.4, 2, 3
Quantity of Each Product	50
Number of Manufacturing Resources per System	4
Sample Size	5

Table 4-6: The setup for experiment 6

The results of this experiment are shown in Figure 4-13 and Figure 4-14. At $\mathcal{R}$ values of less than 1, the dedicated system accumulates much fewer penalty points than the mobile system. At $\mathcal{R}$ = 1.04, the penalty points are nearly equal for both systems. As the *CS* bottleneck intensifies at higher $\mathcal{R}$ values, the mobile system accumulates lower penalty points than the dedicated system. The greatest differences are seen where the ratio of W_j value between products is greatest. The dedicated system is penalised least in the situations where the difference between W_j values is lowest.

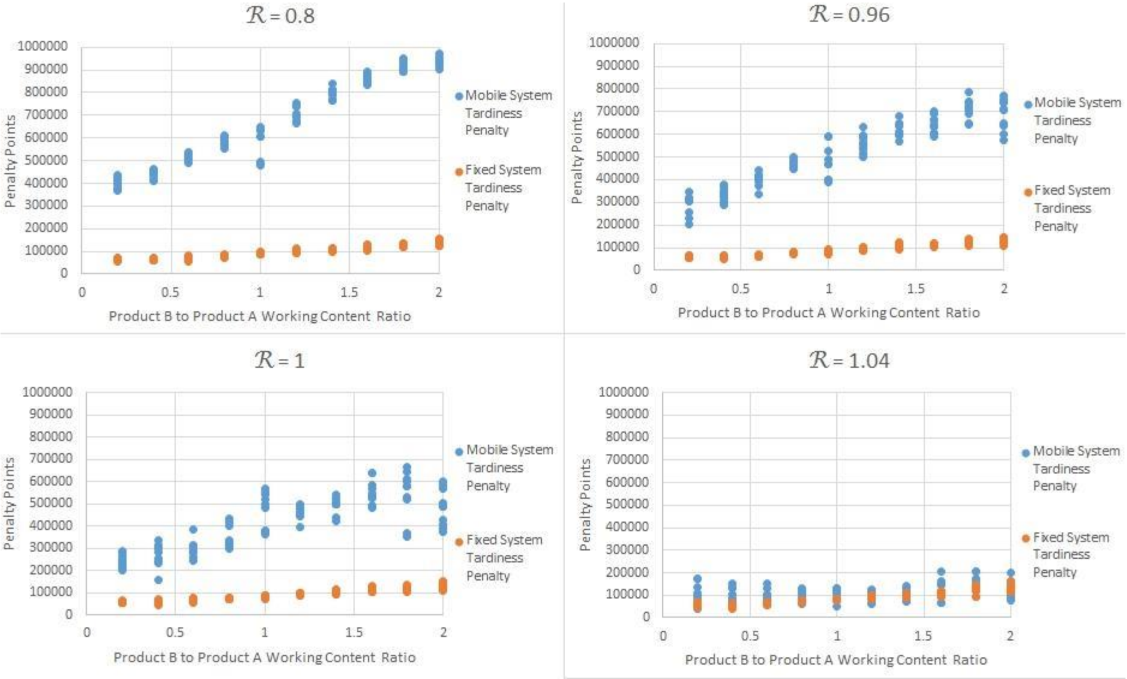


Figure 4-13: The results for the experiment 6 (1/2)

An unknown amount of the mobile system's penalty points can be attributed to the mobile system's behaviour model because it is not optimised in any way to avoid tardiness. The resource agents follow the first-in-first-out decision policy and therefore build up unnecessary penalty points. More advanced behaviour models (like the ones developed in the next 3 chapters) should achieve much fewer penalty points. Nevertheless, at certain extents of product supply bottleneck, the mobile system clearly outperforms the dedicated automation system.

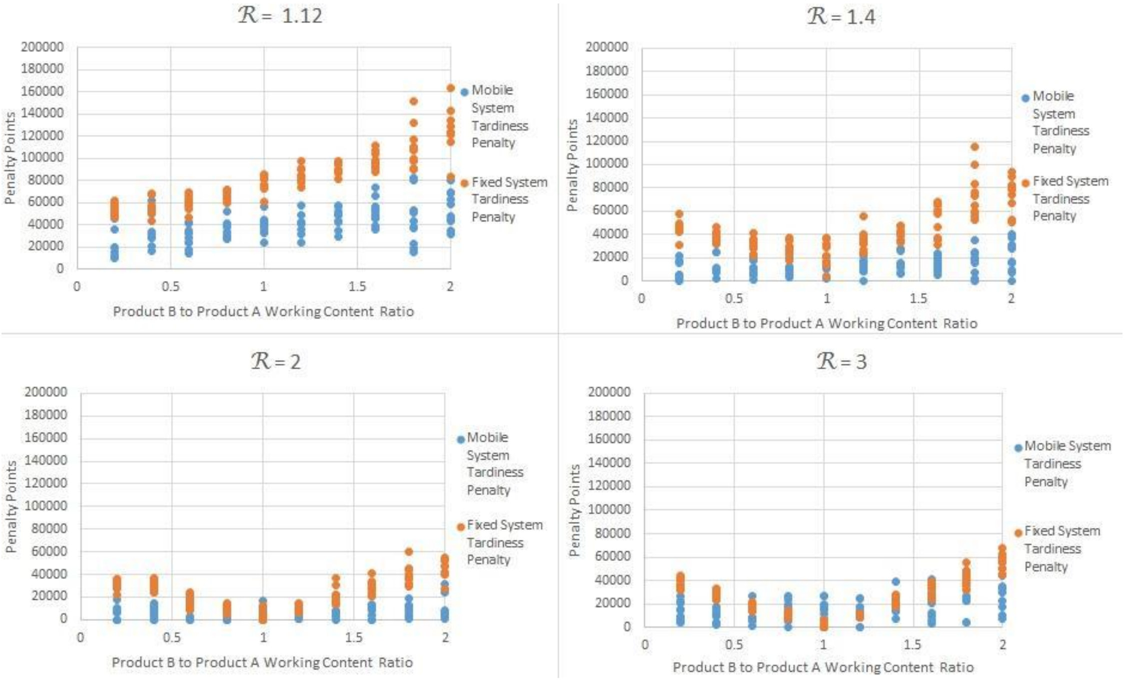


Figure 4-14: The results for the experiment 6 (2/2)

As for the dedicated system, the penalty points are mostly accumulated due to inherent physical limitations. Because the system has very little control over how much of its capacity to allocate to any product, it becomes physically impossible to meet the set due times. An illustration of this is shown in Figure 4-15. Being constrained to the gantry rail leads to the possibility that at some point, freed manufacturing resources of the dedicated system cannot assist others to complete jobs sooner.

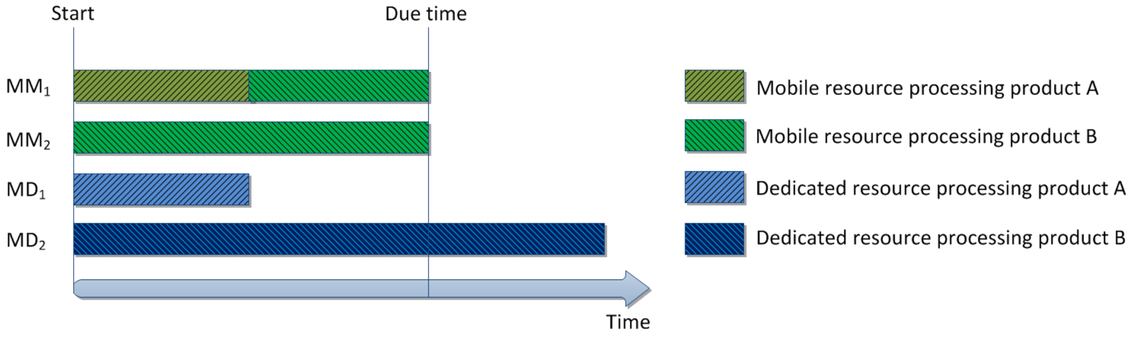


Figure 4-15: The difference in the ability to distribute manufacturing resources

4.4. Discussion

The mobile system completes rush orders much sooner than the dedicated system in every scenario. This is due to concentrating around the same product and combining work rates (product-centric scalability). The dedicated system is unable to do that: it can only allocate a single manufacturing unit to the rush order. On the other hand, it is able to continue work on the regular products when the mobile system is fully occupied with the rush orders. The main conclusion in this chapter is that the ability to move manufacturing resources between workstations is an advantage in situations where there is a need to effectively control the release times of products. However, the disadvantage behind it is the fact that mobile robots cannot do anything useful while moving and therefore some time is inevitably lost there.

Products with larger working content are more favourable for the mobile system because the frequency that the mobile system must relocate is reduced. This is a very useful finding for this doctoral thesis because it is aimed at large structure assembly. It may be impractical to deploy mobile robots for small-sized products due to logistical reasons like spatial constraints around products and small working content. The products that are targeted by this work are generally large enough to allow many mobile robots to fit around them without significant spatial concerns.

Therefore, mobility has two contrasting effects: On one hand, it allows manufacturing resources to increase their utilisation and have greater control over product release times; but on the other hand, moving is a non-value-adding activity that consumes a part of the system's spare capacity. Therefore, the moving benefits come at a cost of having a lower margin for disruptions.

The cooperative decision-making policy of the mobile robots causes larger spreads in results than the non-cooperative one. This means that in some situations cooperation is preferable and in some others, it is not. The self-organisation models must, therefore, consider more than just the launch time of a product when making decisions.

Increasing spare capacity leads to reducing the TUT. Depending on the number of deployed units of manufacturing resources, a large enough spare capacity may mean that the dedicated system simply has a free manufacturing unit that is committed to processing rush orders. The counterargument in favour of the mobile system in such a situation is the fact that their control of capacity distribution can lead to a shorter TUT than the dedicated system can even complete the rush order on its own. The versatility of the mobile system is another benefit because in an unpredictable environment it is difficult to estimate the appropriate capacity of a manufacturing system.

Increasing *IF* reduces the benefits of using mobile robots. However, it should be possible to mitigate the loss of effectiveness by tweaking the decision-making model of the mobile system to spread out the resources as much as possible.

In many experiments, considerable damage to the results of the mobile system is caused by the inability of the control model to plan forward. Most certainly, forward-planning control models that can achieve optimal solutions with respect to their given objectives (like the one proposed in [37]) can dramatically improve the mobile system's results in most of the experiments in this chapter.

4.5. Conclusions

The aim of this chapter is to investigate what the differences are when using a system of mobile robots instead of traditional dedicated automation systems. A comprehensive set of simulations is carried out under identical conditions for both types of systems with identical working capacities. The experimental results lead to several important findings:

- Mobility and scalability allow the mobile system to distribute its working capacity in desired ways.
- The gained control over capacity distribution enables greater control over product delivery times.
- Increased spare capacity shortens the time it takes to complete a rush order and recover the production loss. If the spare capacity is large enough, then production loss can be avoided at the cost of longer rush order completion time. With a large enough spare capacity, the mobile system’s TUT can be shorter than that of the fixed system.
- Moving decreases the availability of a system. It is best to move as little as possible and only when required. Therefore, a mobile system is particularly beneficial in large structure assembly, where there are products with large working contents. Such products reduce the proportion of time that is used for moving.
- The greatest differences in the systems are experienced when there are inconsistencies with the product supplies. Firstly, the mobile system can complete a rush order much sooner; however, it takes roughly the same time to fully recover into normal operating conditions. And secondly, the mobile system can achieve higher utilisation when there is a bottleneck in the product supply.
- A disadvantage of the mobile system that will need addressing before physically deploying mobile robots in large structure assembly is the possibility of interference. Clearly, some work must be done for a satisfactory distribution of mobile robots around workpieces to even out the workload and minimise interference with one another.

Due to the nature of the dedicated systems, there has never been a need for a control model that can be used by the mobile system in such scheduling problems. Therefore, the development of a more sophisticated model for this purpose should enable manufacturers to get the best out of the presented benefits in this chapter.

Finally, where and when is it better to use either of these systems? Many of the conclusions indicate a paradox: It is a very useful ability to move, however moving reduces the availability of the mobile system. This means that moving is an advantage and a disadvantage at the same time. The experiments show that the mobile system’s advantages increase with increasing product working contents and reducing movement frequency. Also, knowing that the manufacturing trends have been steadily pointing towards more customisation, unpredictability and shorter life cycles, the versatility and adaptability of mobile robots in manufacturing are expected to become more appealing as time moves on.

These conclusions verify the first hypothesis of this thesis that a mobile system has better control over product delivery times than a fixed automation system in large structure assembly.

Chapter 5 - Development and Validation of the Priority Aging Policy for the Hybrid Self-Organisation Model

5.1. Introduction

In Chapter 4, it is shown that mobile automation systems can have advantages over fixed systems in certain circumstances. This addresses the first knowledge gap and verifies the first hypothesis, as discussed in Chapter 3. The next knowledge gap is that fact that there were no known methods of controlling a mobile system in LSA. Thus, the second hypothesis proposes two fundamentally different ways for doing so and states their expected performance characteristics.

Based on the literature review, hybrid and decentralised self-organisation models are proposed to be able to handle a wide range of possible manufacturing environments. The hybrid model was expected to perform better in more static environments and the decentralised model in the more dynamic ones. The development of the hybrid model is presented first.

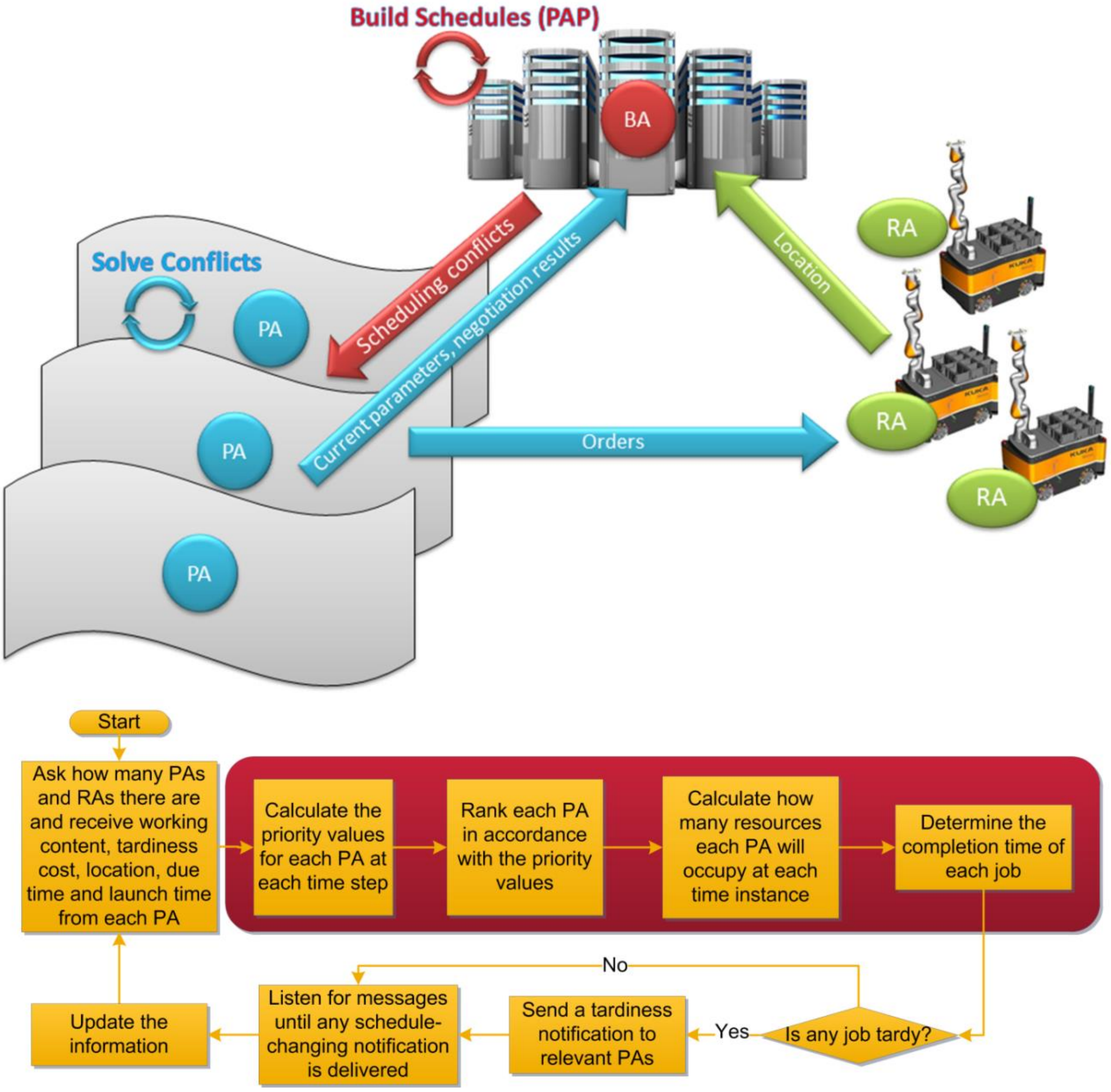
The hybrid model consists of distributed RAs and PAs; and a central blackboard agent. The objective of PAs is to attract RAs in a way that their jobs would be done by their due times. If it is impossible to meet the due times, then the interest is in minimising the tardiness. Agents in the hybrid model are cooperative and willingly share their information. It is described in full detail in Chapter 6 how at the start of any simulation and after disruptions, all PAs and RAs send their parameters to the BA. Based on all of the details, the BA then simulates through the schedule until the last PA has been completed. However, in order to do so, it requires knowing how many RAs will be working at each PA at each instance in time. Based on the urgency of each PA, it should possible to prioritise them and allocate RAs with the aim of causing minimal TWT. Thus, in order to implement the hybrid self-organisation model, a method for ranking PAs based on their priorities have to be developed.

The literature review in Chapter 2 reveals that one appropriate way of doing so would be by applying a priority aging policy (PAP). PAPs are commonly used in computing applications where there is a constant influx of high priority tasks that prevent low priority tasks from getting processed. The principle is based on increasing job priorities so that the initially low priority tasks get processed sooner. In this chapter, different PAPs are investigated in the context of the scheduling problem in LSA. It is assumed that no PAP will produce optimal schedules. Therefore, the PAP in the hybrid model will be used to produce an initial schedule that can then be improved through negotiations between agents.

Priority aging policies (PAPs) work by assigning priorities to PAs based on their launch times, due times and tardiness costs. PAs then inform RAs of their priorities at each time step (second). The RAs respond to the PA with the highest priority. The highest ranked PA then accepts as many RAs as it requires (or as many as there are, if insufficient) to be completed by the due time. Rejected RAs then respond to the next highest ranked PA and so on until each one has been allocated to a PA.

The structure of the hybrid model and the BA's flowchart are presented in Figure 5-1. The contribution of this chapter is focused on the part highlighted by the red background in the flowchart. It is the development and justification of a PAP for mobile robots in large structure assembly. The PAP governs the priority ranking order of each product agent (PA) in the hybrid self-organisation model at each instance in time. The priority ranking order determines which PAs are first to order resource agents (RAs) for processing their jobs. Therefore, if applied appropriately, it should provide initial schedules with (nearly) minimal TWT in this problem without any additional negotiations. It is then the purpose of the hybrid model in Chapter 6 to make any adjustments to further improve these initial schedules if possible.

Priority aging is commonly used in computing applications to ensure that none of the jobs are starved. This is a common problem in pre-emptive priority scheduling, where lower priority tasks never get processed due to constantly arriving higher priority tasks [124]. By priority aging, it is meant that the priority of each job increases in time. This way, given sufficient time, every task that initially had low priority, should eventually rise high enough in the rankings to get processed.



Build Schedules (PAP)
BA
Solve Conflicts
PA
PA
PA
Scheduling conflicts
Current parameters, negotiation results
Location
Orders
RA
RA
RA
Start
Ask how many PAs and RAs there are and receive working content, tardiness cost, location, due time and launch time from each PA
Calculate the priority values for each PA at each time step
Rank each PA in accordance with the priority values
Calculate how many resources each PA will occupy at each time instance
Determine the completion time of each job
Is any job tardy?
Yes
No
Send a tardiness notification to relevant PAs
Listen for messages until any schedule-changing notification is delivered
Update the information

Figure 5-1: The structure of the proposed hybrid self-organisation model

The challenge in this problem is not only that a task should *eventually* be processed, but also *in due time*. Therefore, the PAP developed in this chapter considers more than just the time a task has been active on the shop floor. The objective of the PAP is to achieve schedules with minimal total weighted tardiness (TWT). In this case, the weight would be the cost of tardiness per minute and tardiness itself is measured in minutes. Therefore, the PAP of the hybrid model considers the time left until due time and the tardiness cost.

At each moment, every PA is aware of how many RAs they require in order to be completed on time. Each PA then orders as many RAs as necessary to finish its job on time. With this approach, PAs with highest priorities order the necessary number of RAs first.

Because no PAPs had been described for such applications in literature, three PAPs are proposed and compared for this work. The aim of this chapter is to identify the PAP that causes the most predictable and smallest number of scheduling conflicts. This PAP should result in the smallest number of necessary adjustments and variety of problems to handle. The shop floor layout and scheduling constraints are the same as specified in Chapter 4.

5.2. Method

As no relevant literature on such a specific application had been found, three PAPs are proposed and compared to one-another. The NetLogo 5.3.1 software package [120] was selected for carrying out the simulations because this analysis does not require highly sophisticated behaviours by any of the agents.

By knowing the individual priority values of PAs at every time step, it is possible to predict how many RAs each one will order. From that, it is possible to determine the resultant TWTs and predict scheduling conflicts. Based on the obtained results, the best-performing PAP can be determined. Thus, the interest in this chapter is on the performance of PAPs and not on any individual behaviour of agents. The challenge is in understanding which of the three proposed PAPs is most robust for typical manufacturing scenarios and performs best in a range of different scenarios.

The simulation is set up with four PAs and ten RAs. Each RA has a due time d_j, a tardiness cost C_{tj} and a working content W_j. The flowchart for the simulation is shown in Figure 5-2. The simulation starts with acknowledging the details for each PA. It then proceeds by predicting how many RAs will be occupied by each PA at each minute. Following from that it is possible to predict which PA will be tardy and calculate its tardiness penalty.

The combined work rate of all existing RAs is equal to one unit of work per minute. This means that if all RAs were allocated to a PA with a working content of 10,000 units, the task would be completed in 10,000 minutes. Ten RAs are deployed on the shop floor in these simulations, which means that each one can contribute 0.1 units of working content per minute.

The results re measured in TWT. The TWT is the product of the tardiness cost and tardiness in minutes. For example, if a PA is tardy by 100 minutes and has a tardiness cost of 100 £/min, then the TWT of this PA is 100 * 100 = £10,000.

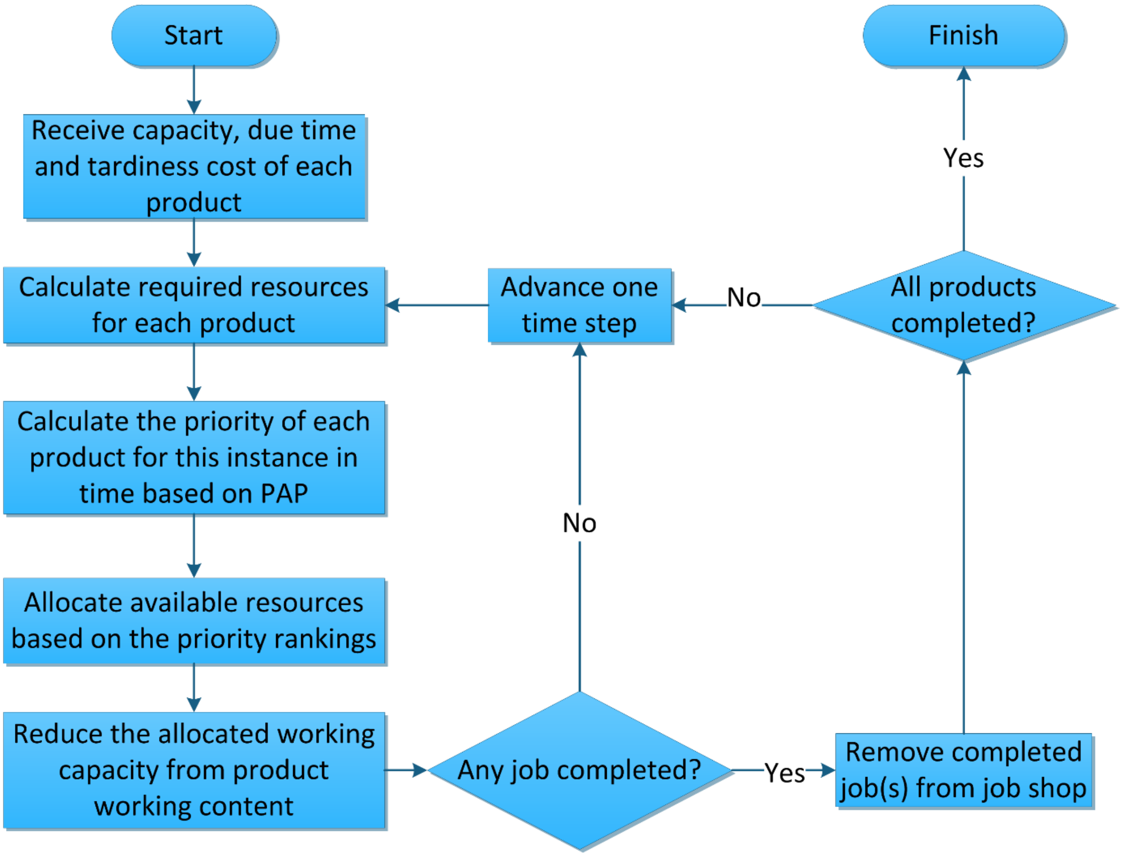

Start
Receive capacity, due time and tardiness cost of each product
Calculate required resources for each product
Calculate the priority of each product for this instance in time based on PAP
Allocate available resources based on the priority rankings
Reduce the allocated working capacity from product working content
Any job completed?
Yes
Remove completed job(s) from job shop
No
Advance one time step
No
All products completed?
Yes
Finish

Figure 5-2: The flowchart for the priority aging simulation

Moving times for RAs between jobs are ignored in these experiments for two reasons. Firstly, it is shown in Chapter 4 that the proportion of moving time of mobile robots in large structure assembly is negligible. Secondly, including moving times in this chapter would add noise to the analysis of pure PAPs.

5.3. Selection of Priority Aging Policies

This section describes the PAPs that are considered for the self-organisation model. The PAPs found in literature were developed for different purposes (not considering due time nor product-centric scalability), which is the reason why new ones have to be investigated in this chapter. In this thesis, the PAPs are always a function of the tardiness cost and time left until the due time. It must be noted that all of these formulae apply from the instance when a PA is launched until its due time. If the PA is completed without running over its due time, then it is removed. However, if a PA has gone past its due time without completion, then its priority remains equal to its tardiness cost until whenever it is completed. This way the loss can be correlated to the monetary value that is lost in a real manufacturing environment.

An illustration of the progression of priority with each PAP is shown in Figure 5-3. It is shown that the exponential PAP starts with much lower increments of increasing the priority than the linear PAP. The closer it gets to its due time, the more steeply it increases the priority. The cost-weighted exponential PAP was designed to further lower the initial increments and further increase the increments at late stages. Thus, the considered PAPs differ in the way how PAs gain their priorities over time.

The linear PAP is governed by Equation 5.1. The priority $P(t)_{lin}$ is always a function of the job's tardiness cost C_{tj}. At the start, time *t* is equal to the end of loading time t_l' and as time *t* progresses, the priority $P(t)_{lin}$ linearly increases until the due time $\mathrm{d_j}$. There, the fraction of the equation becomes 1 and consequently the priority $\mathrm{P(t)_{lin}}$ equals to the tardiness cost $\mathrm{C_{tj}}$.

$$P(t)_{lin} = C_{tj} * (\frac{t - t_{l'}}{d_j - t_{l'}}) \qquad \textit{(Equation 5.1)}$$

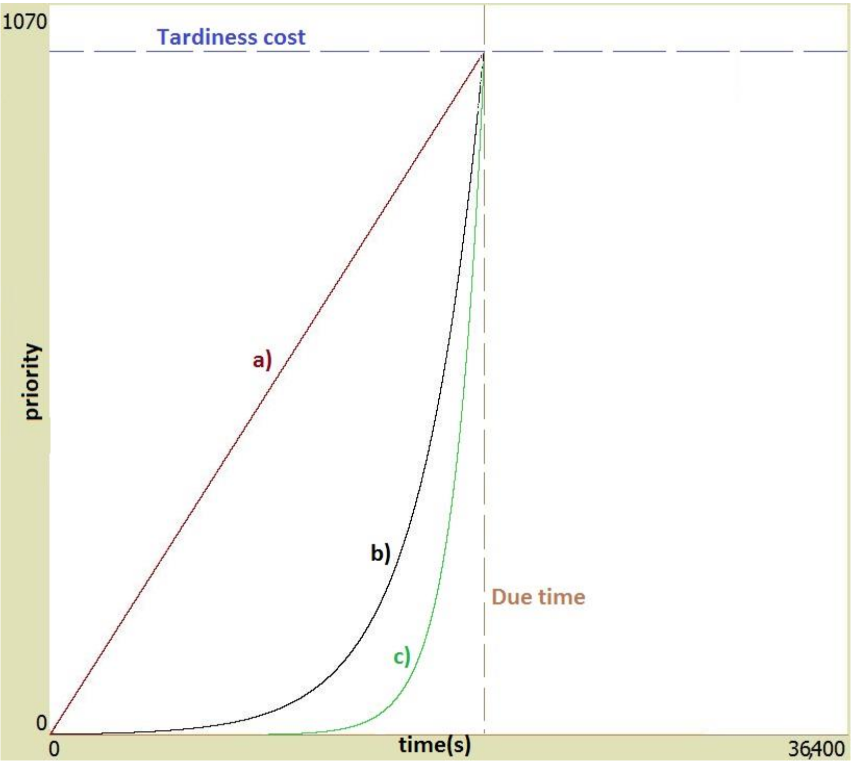


Figure 5-3: Illustration of the a) linear, b) exponential and c) cost-weighted exponential PAPs priority progression in time

The exponential PAP is governed by Equation 5.2. The equation is very similar to the linear one. The difference is that the part in brackets ($\frac{t - t_{l'}}{d_j - t_{l'}}$) that is multiplied by the tardiness cost C_{tj} in the linear PAP, is set as the power of the tardiness cost in the PAP. As a result, the priority builds up much slower from the start, but still reaches the same value (C_{tj}) at due time.

$$P(t)_{exp} = C_{tj}^{\frac{t - t_{l'}}{d_j - t_{l'}}} \qquad \textit{(Equation 5.2)}$$

The shape of the exponential curve can be manipulated in a number of ways. Equation 5.2 is a balanced equation because it applies the exponential relation in its most straightforward form. To test a PAP where the priority is further lowered at the early stages and PAs with higher tardiness costs get an additional advantage, the cost-weighted exponential PAP is proposed. The governing equation for this PAP is shown in Equation 5.3. The difference with the exponential PAP lies in the additional fraction with which the power of the tardiness cost is multiplied. The added fraction of the PA’s tardiness cost C_{tj} divided by the maximum tardiness cost of all considered PAs $C_{t,m}$ gives additional preference to PAs with higher tardiness costs. It must also be noted that the fraction does not let PAs with lower than maximum tardiness costs to reach $P(t)_{cwe}$ = C_{tj} before due time d_j. Nevertheless, once due time has been passed, $P(t)_{cwe}$ = C_{tj} until completion.

$$P(t)_{cwe} = C_{tj}^{\frac{(t - t_l{}') * C_{tj}}{(d_j - t_l{}') * C_{t,m}}} \qquad \textit{(Equation 5.3)}$$

5.4. Design of Experiments

The experiments are designed to analyse how each PAP handles various challenges set by the launched PAs. The analysed scenarios include individual and combinations of the following: just enough RAs for each job; shortage of RAs; close due times and large differences in tardiness costs. Trivial scenarios where there is spare capacity or otherwise no pressure for products to gain resources are not considered. To avoid using absolute numbers, all of the variable factors in these experiments are used as ratios. The time-based factors are ratios of the working contents per product and the tardiness costs of products are presented as ratios of one-another. The following values are fixed for the experiments:

1. The full manufacturing capacity of the mobile system is 1 unit of work per minute
2. The mobile system consists of 10 mobile resources (each one works at a rate of 0.1)
3. Each product has a working content of 10,000 units
4. The moving time is neglected
5. No disruptions are considered. This means that no rush orders are launched out of schedule and there are no additional challenges to consider beyond the initial schedule

The following factors are varied in the experiments:

- Ratio of available and required capacity (*ARC*): The fewer resources there are, the more difficult it is to allocate them optimally. In order to highlight how strong a shortage of resources is, it is presented as a ratio with respect to the product's working content. The formula for calculating this ratio is shown in Equation 5.4.

$$ARC = \frac{J * W_j - d_j}{W_j} \qquad \textit{(Equation 5.4)}$$

  The equation applies locally where there is pressure on the schedules. For example, if there is a single product with due time d_j = 10,000, then *ARC* = 0. If d_j = 8,000, then *ARC* = $\frac{1*10{,}000-8{,}000}{10{,}000}$ = 0.2. Where *ARC* is negative, it is neglected, because it represents no pressure on the schedule. I.e. if there were 4 products, the first 3 of which had d_j > 30,000 (*ARC* = 0) and the fourth product had d_j = 37,000, then *ARC* = $\frac{4 * 10{,}000-37{,}000}{10{,}000}$ = 0.3 for that product. If more than one product has *ARC* > 0, then they are added up. The *ARC* is generated by choosing due times d_j for products accordingly. Thus, *ARC* represents a measure of the manufacturing system's shortage of capacity with respect to any given schedule.
- The closeness of due times (C_{dj}): It is shown in Chapter 4 that when there is spare capacity, the mobile system has no pressure on the schedules and the allocation of resources is trivial. However, when *ARC* ≥ 0, there is some pressure on the schedules and effective distribution of resources becomes a challenge. When this happens and products have a wide spread of due times, it is natural to prioritise the earlier PAs to avoid the multiplication of their tardiness costs with a long time (see Equation 5.7). However, in situations with close due times of several PAs, the RAs are more difficult to allocate appropriately. The closeness is calculated by applying Equation 5.5.

$$C_{dj} = 1 - \left| \frac{d_{j,1} - d_{j,2}}{W_j} \right| \qquad \textit{(Equation 5.5)}$$

Variables $d_{j,1}$ and $d_{j,2}$ are the due times of two products. The absolute value of their difference is divided by W_j and subtracted from 1. This way, if two products are due at the same time, C_{dj} = 1. If one is due 5,000 minutes before the other one, then C_{dj} = 0.5. Any two due times that result in $C_{dj} \leq 0$ are of no interest in this work, because they are simply not close to one-another. The reason why they are benchmarked against the working content of a single product is because at C_{dj} = 0, the latter product can be completely processed after the earlier one and there is no challenge due to the closeness of due times. It must be noted that when this variable is used, the due time d_j of the product with higher tardiness cost is later than the due time d_j of the product with the lower tardiness cost.

Differences in tardiness costs d_{tc}: this is a challenge that is generally addressed in priority aging problems. Due to having a low tardiness cost and consequently a low priority, some jobs may be starved. In particular, such jobs are often starved because another job with a much higher priority and close due time is processed instead. The d_{tc} is calculated as shown in Equation 5.6. The greater tardiness cost is denoted as $C_{tj,2}$ and the smaller one as $C_{tj,1}$. This way, if the tardiness costs are equal, then d_{tc} = 0 and if one tardiness cost is half of the other one, then d_{tc} = 0.5. Therefore, the differences in tardiness costs are represented as ratios one one-another.

$$d_{tc} = 1 - \frac{C_{tj,1}}{C_{tj,2}} \qquad \textit{(Equation 5.6)}$$

The measured output from each experiment is the TWT. The calculation of TWT is shown in Equation 5.7 and it measured in £. It is a product of the PA's tardiness cost C_{tj} and how tardy it is (release time r_j minus due time d_j).

$$TWT = C_{tj} \times (r_j - d_j) \qquad \textit{(Equation 5.7)}$$

The experimental setup is shown in Table 5-1. It is a multi-factor experiment with three factors and four levels each. The working content of each product is fixed to 10,000 units. To maintain a simple overview of the results, only four products are used in this experiment. The challenge of each scenario is isolated to the first two products. The reason for the existence of products C and D is to verify that any considered scheduling challenges would not have a knock-on effect on the next products. The scheduling challenges due to changing the variables can be demonstrated with this quantity and any additional products would cause a recursive effect. The first two products are always launched at t = 0 and the other two depending on the *ARC*. For example, if *ARC* = 0.9, then the due time of the latest product out of the first two would be at t = 19,000. The third and fourth product would be launched at 19,000 as well, but have one due time $d_j \geq 30{,}000$ and another one $d_j \geq 40{,}000$. The latter two are not under pressure per se, because the challenge is isolated to the first two products (to avoid multiplication of penalties). Their purpose is to show how the schedule proceeds after challenging scenarios. Similarly, the closeness of due times C_{dj} and the difference in tardiness costs d_{tc} are isolated to a single pair of products to highlight the single challenge at that extent.

It was predicted that despite having some effect on the system, some of these challenges would not result in TWT = £0 at some of the settings. To avoid that, additional pressure on the schedules is created by setting fixed values for non-varied variables (see Table 5-1). Each experiment is simulated only once for each of the PAPs due to the absence of variability. The exact values for the whole experiment are shown in Appendix A.

The experiment with *ARC* was expected to increase each PAP's TWT as the *ARC* is lowered. This is expected, because when there is a shortage of resources, then regardless of they are allocated, some products will inevitably be tardy. The interest is to assess how the PAPs compare in scenarios of resource shortage.

Variable	Varied Values	Fixed Factors
ARC	0.7, 0.8, 0.9, 1	C_{dj} = 0.8, d_{tc} = 0.25
C_{dj}	0.7, 0.8, 0.9, 1	ARC = 1, d_{tc} = 0.25
d_{tc}	0, 0.25, 0.5, 0.75	ARC = 1, C_{dj} = 0.8

Table 5-1: The setup for the PAP experiments

The experiment with close due times (C_{dj}) is designed to assess how the PAPs allocate resources when there is a difference in tardiness costs for products at close due times. As mentioned earlier, the challenge arises when a product with a higher tardiness cost is due later than a product with a lower tardiness cost. In these cases, the product with the lower tardiness cost can get starved and result in some TWT. Additional pressure on the schedules is created by setting *ARC* = 1, meaning that there is no spare capacity in the system and TWT = £0 should be achievable if the resources are allocated ideally. Thus, if any job is starved, it will result in some TWT.

The experiment with differences in tardiness costs (d_{tc}) is analogous to the one above, however instead of changing the closeness of due times, the ratio of tardiness costs between products is varied. Essentially, this addresses the same challenge as the previous experiment but varies the other factor. This enables the assessment of the PAPs' performances from a different perspective. The expectation was that there should be no TWT where the tardiness costs are equal because there is a sufficiency of resources (*ARC* = 1). Once d_{tc} increases, the problem becomes more challenging and it is important to determine which PAP handles it in the best way (lowest TWT).

5.5. Results and Discussion

The resulting TWT values from each PAP in the *ARC* experiment are shown in Figure 5-4. The exponential PAP performs consistently the best out of the three, displaying the lowest TWT at each setting. The linear and cost-weighted exponential PAPs achieve higher TWTs than the exponential PAP at each setting by roughly a million. Whilst the linear PAP's TWT is roughly 200,000 units lower than the cost-weighted exponential PAP's at *ARC* = 0.7 and 0.8, this relationship consistently worsen at higher *ARC* values.

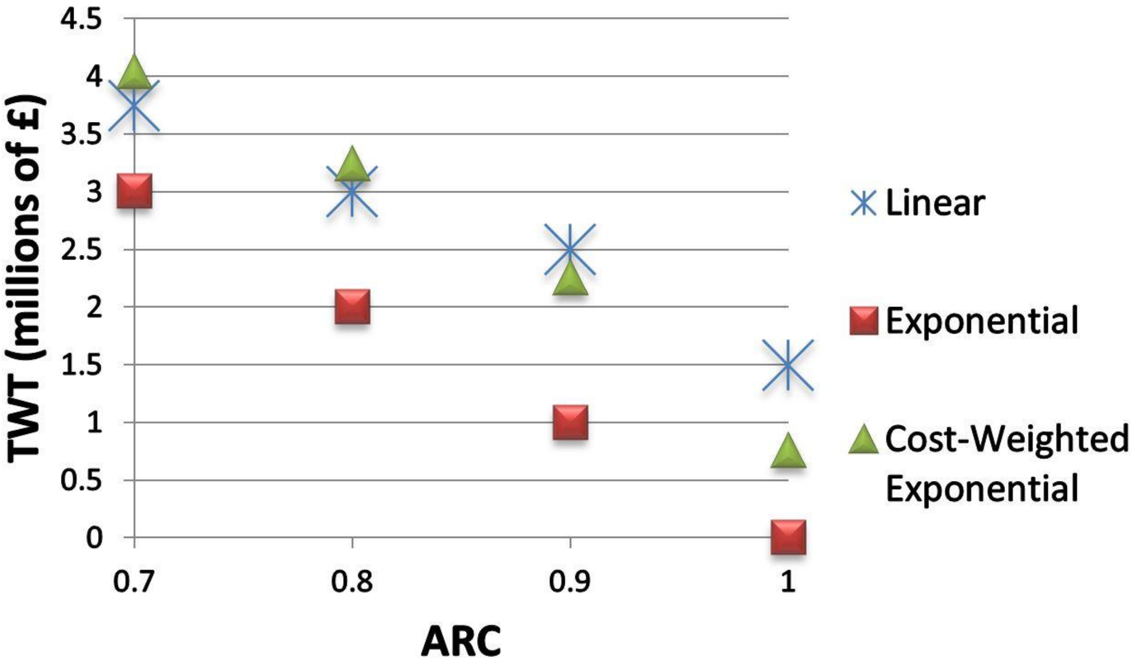


Figure 5-4: The results for the ARC experiment

The reasons for the differences are shown in Figure 5-5. There, an example plot of priority in time and consequent mobile resource schedules are shown at *ARC* = 0.8. The priority plots show how the priorities of products increase in time for all of the considered PAPs. At the bottom of the figure, it is illustrated at which product each RA is working throughout the timeline as shown at the top. A reduction in ARC causes the peaks of products A and B to move leftwards on the graph.

For the linear PAP, it is shown that product B's priority is the highest until its completion (at t = 18,000 minutes) despite product A having an earlier due time (16,000 minutes). In order to finish by d_j at t = 18,000 minutes, six RAs has to work at it for 2,000 minutes and then 5 RAs until d_j. This causes product A to starve, because it requires more than 5 RAs from the beginning until its d_j but does not reach a high enough priority for that.

That challenge is solved by the exponential PAP, where product B’s priority ages at a slower rate in the beginning and allows product A to receive the necessary RAs until its d_j at t = 16,000 minutes. This is reflected in the illustration of the RAs’ schedules at the bottom of Figure 5-5: product A receives seven RAs at the beginning of the schedule and reduces that amount to six when it becomes sufficient. The additional resource is necessary for it because on average, it needs 6.25 RAs per minute to finish by due time. This results in starving product B, which is second in the priority ranking order until t = 16,000 minutes. Once product A is completed, product B receives 10 RAs until its completion 2,000 minutes later than d_j. Despite the tardiness of product B, the TWT is lower than with the linear PAP, as shown in Figure 5-4. Thus, it can be stated that the exponential PAP outperforms the linear PAP due to slower aging of priority at the early stages of launching products.

The cost-weighted exponential PAP gives preference to products with higher tardiness costs. It is shown in Figure 5-5 that product A has a lower priority than product B at the start. Once product A exceeds its d_j, it briefly gets the highest priority because its priority is equalled to its C_{tj}. However, that soon changes back and product A is tardy. In addition to achieving roughly the same TWT as the linear PAP, the cost-weighted exponential PAP causes the mobile resources to change jobs more frequently. At the bottom of Figure 5-5, it is shown that many resources had to change between products A and B near product A’s d_j at t = 16,000 minutes. Therefore, it can be said that the cost-weighted exponential PAP is not well-suited for a shortage of resources.

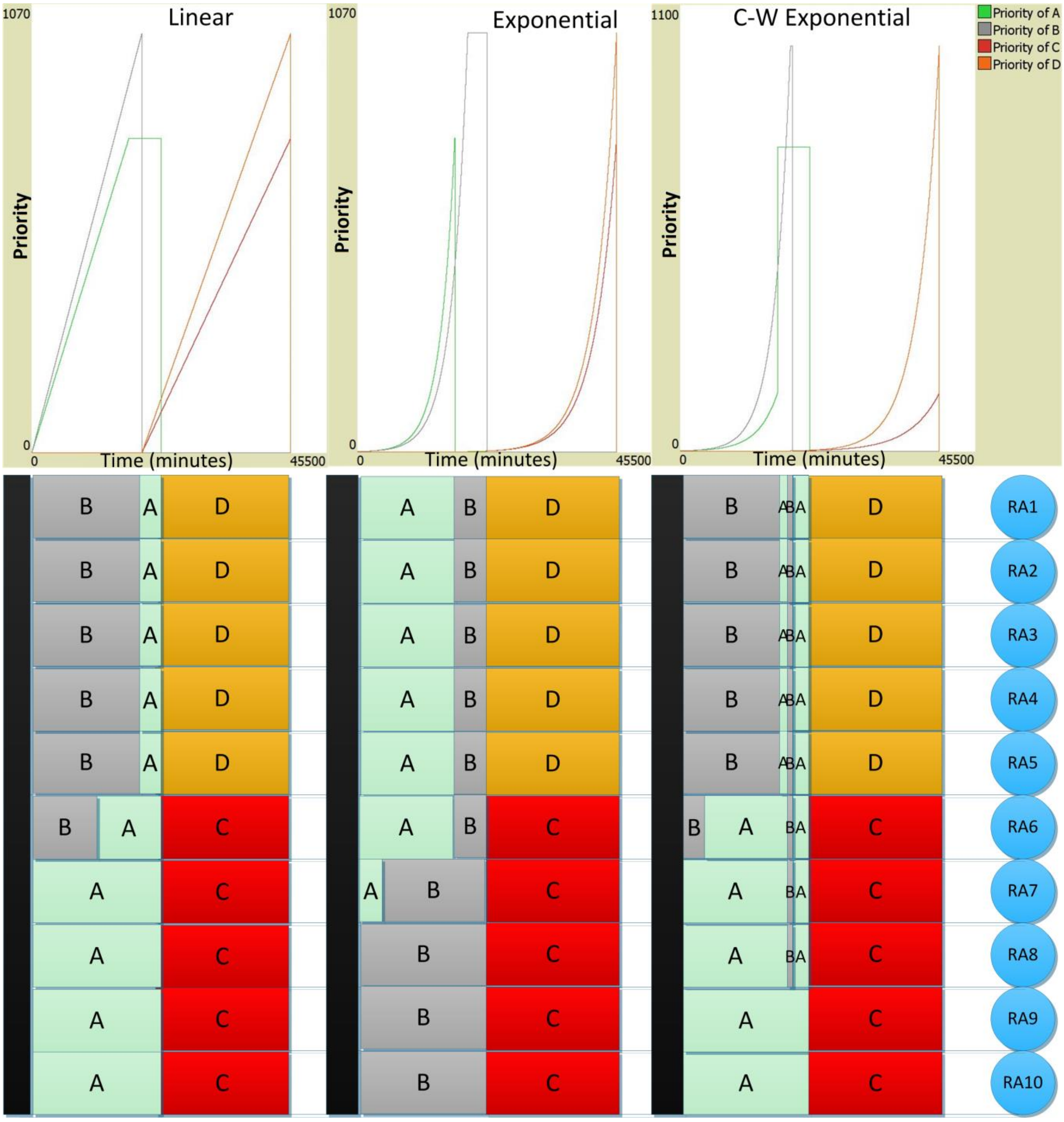


Figure 5-5: The priority in time plots and schedules for mobile robots in the ARC experiment

The results for the C_{dj} experiment are shown in Figure 5-6. The linear PAP performs the worst across the experiment, whilst the exponential PAP achieved TWT = £0 on all settings and the cost-weighted exponential PAP achieves roughly average results between them.

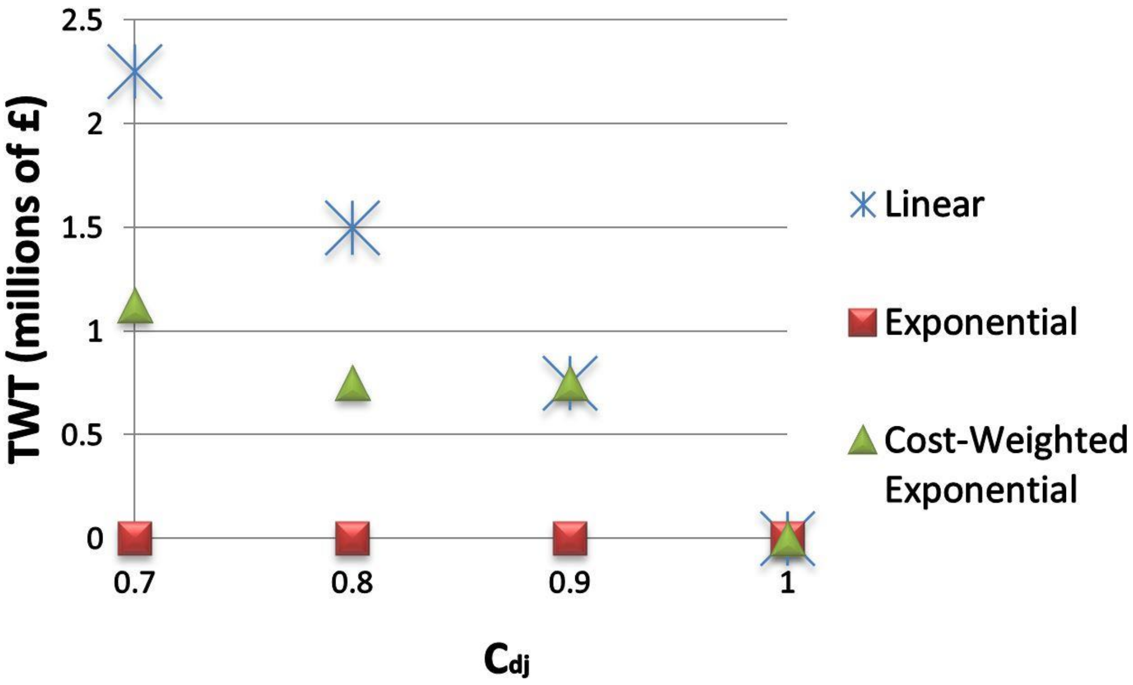


Figure 5-6: The results for the C_{dj} experiment

The reason why the linear PAP's TWT increases as the C_{dj} decreases can be deduced from Figure 5-7. There, a sample priority in time plot and respective schedules for RAs are shown at $C_{dj} = 0.8$. A reduction in C_{dj} causes the tip of product A's priority to move leftwards on the plot.

Because product A is due earlier and has a lower tardiness cost C_{tj} than product B, it is starved for exactly the time period that it is due earlier by. The green line under the linear PAP's priority plot represents product A's priority. Because it peaks at due time, it is clear that it cannot be ranked higher than product B (grey) unless its due time is early enough. Therefore, the linear PAP is not well-suited for scheduling products that have close due times.

The exponential PAP performs in the exact opposite way. It is shown in Figure 5-7 how despite the close due times, product A has the highest priority throughout its production time. This enables it to be completed in time without causing product B to be tardy.

The aging at the early stages for the cost-weighted exponential PAP is shown to be too slow. Product A is starved until its due time, after which it is boosted to the top priority rank until completion. As a result, product A is completed tardy and product B was completed on time. Like in the *ARC* experiment, the tardy product causes additional movement for the resource agents. Combined with the resultant TWT, such a PAP can be prohibitive for manufacturers.

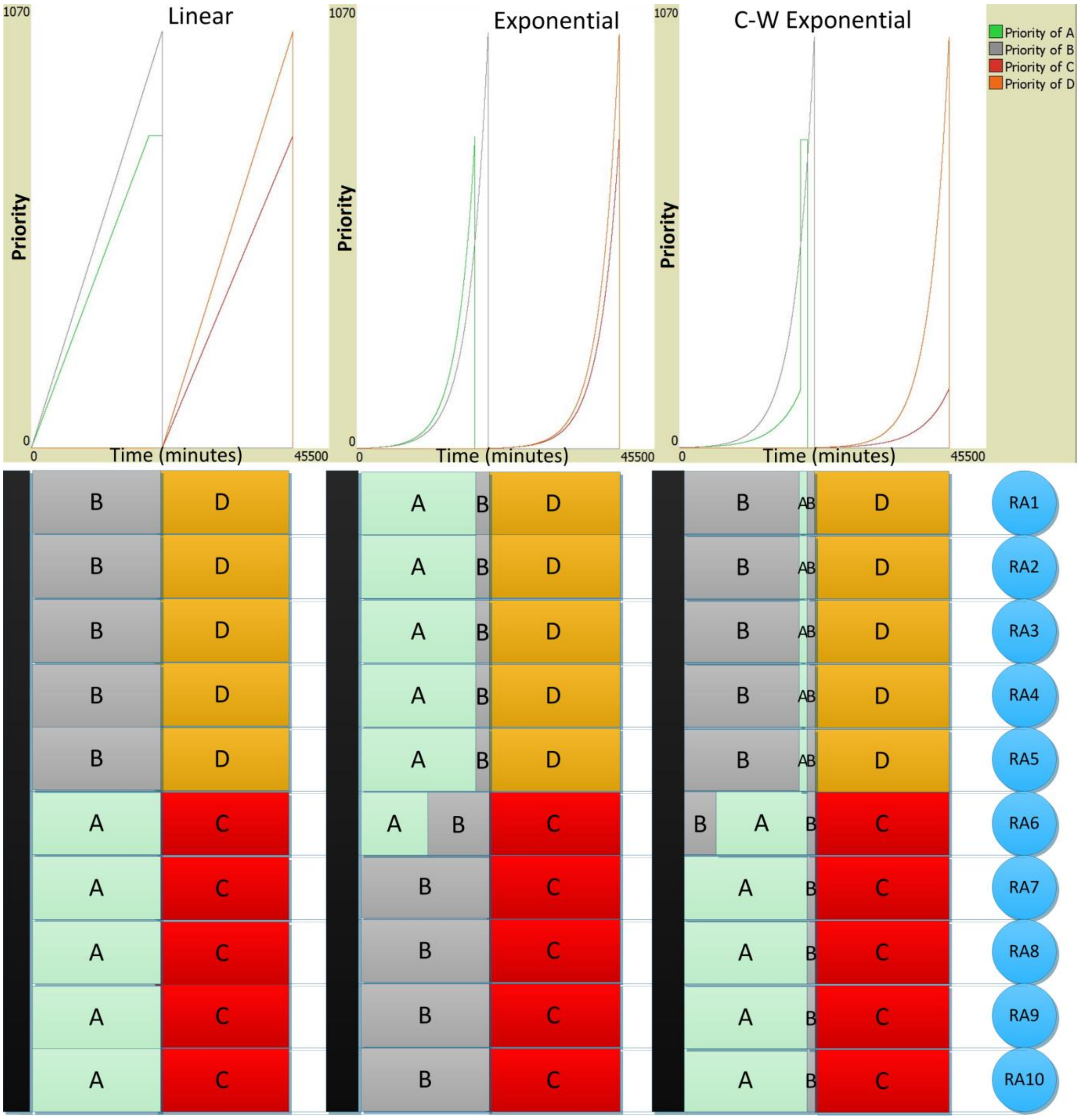


Figure 5-7: The priority in time plots and schedules for mobile robots in the C_{dj} experiment

The results for the d_{tc} experiment are shown in Figure 5-8. Here, at three settings out of four, all PAPs achieved the same results. This shows that large differences in tardiness costs are very difficult for all of the PAPs to handle. At d_{tc} = 0, all of the PAPs achieved TWT = £0, showing the effectiveness of all

PAPs at equal tardiness costs. At d_{tc} = 0, only the exponential PAP achieves TWT = £0 while the other two cause product A to be tardy. At d_{tc} = 0.5 and d_{tc} = 0.75, the differences in tardiness costs re too large for all of the PAPs and they show the same results. The differences in TWT between d_{tc} = 0.5 and d_{tc} = 0.75 are only because the tardiness costs per unit time are different. The time spent in tardiness in these parts of the experiment is equal for all PAPs.

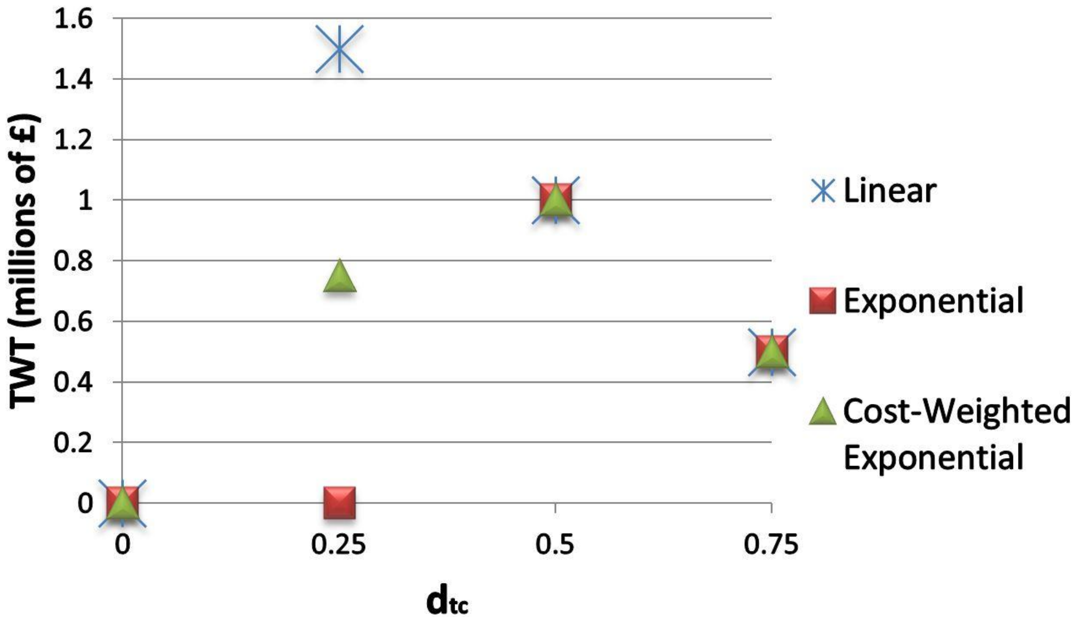


Figure 5-8: The results for the d_{tc} experiment

The priority in time plots and individual schedules for RAs in the d_{tc} experiment at d_{tc} = 0.75 are shown in Figure 5-9. The d_{tc} itself is represented as the ratio between the peak values of all products. I.e. with an increase in d_{tc}, the peaks of products A and C are lowered. From all of the shown priority plots, it is clear that none of the considered PAPs can ensure that the product with lower tardiness cost and earlier due time get completed on time. On the priority plots, it is shown that the exponential PAP's product A's priority was the closest to the priority of product B throughout its production. This is the reason why at d_{tc} = 0.25, the exponential PAP achieves TWT = £0 whilst the other PAPs do not. Therefore, even in the strongly challenging scenarios, the exponential PAP either performs as well as the other two or better.

The individual schedules for all of the RAs are identical in the provided example. This is because the products with the higher priority costs could attract exactly as many RAs as they needed and at no point are the products with lower priority costs able to rank higher than them.

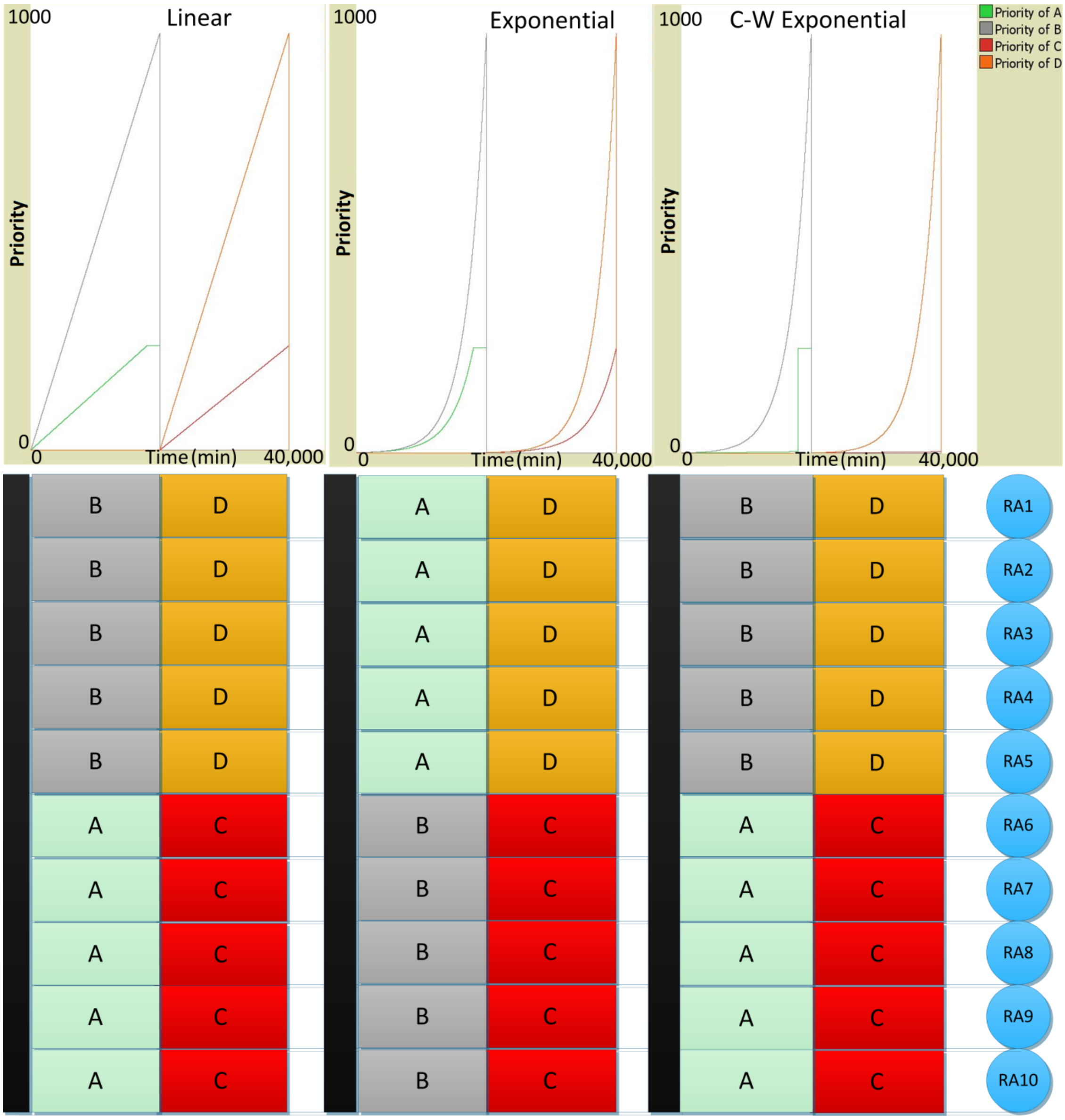


Figure 5-9: The priority in time plots and schedules for mobile robots in the d_{tc} experiment

As shown in this chapter, there is a big difference in how the priorities of products age in time. The less effective PAPs are unable to avoid tardiness in many cases because individual products have a small amount of working content left to be processed at their due times and accumulate some TWT as a result of that. The exponential PAP is very effective in allocating RAs in a way that causes the least TWT. In Figure 5-5, highest priority is given to the product with lower tardiness cost and earlier due time. However, in Figure 5-9, the highest priority was given to the later product with higher tardiness cost. In both of these examples, the achieved TWT is the lowest or at least equal to other PAPs. This proves that there is no trivial solution to the scheduling problems considered in this work.

The linear PAP shows mediocre performance throughout the experiments. It often accumulates TWT and on a few occasions outperforms the cost-weighted exponential PAP. All in all, this PAP displays some of the highest TWT values throughout the chapter and should not be used as part of the hybrid self-organisation model.

The exponential PAP consistently achieves the lowest TWT out of the three policies. It consistently enables products to reach the necessary priority ranks to order sufficient RAs. Where there is a mathematical shortage of RAs, the exponential PAP always achieves the lowest TWT out of the three. Close due times are handled without tardiness at all. In scenarios where products have large differences in tardiness costs, this PAP has the greatest difficulties. However, that applies to the other PAPs as well. Therefore, because the exponential PAP consistently performed the best out of these three PAPs, it should be used in the hybrid self-organisation model.

The cost-weighted exponential PAP is an extension of the exponential PAP. By giving more priority to PAs with higher tardiness costs, the intention is to assess how an even deeper curve on the priority plots performs. The carried-out simulations show that this PAP does not meet its intentions. It consistently accumulates some TWT and on average performs close to the linear PAP. Regulating such a PAP with agent behaviours is impractical due to having less consistency in the initial schedule and the frequency at which it moves RAs between products. Moreover, the inconsistent scheduling would result in a larger number of inter-agent negotiations, which is likely to cause excessive communication loads on the agents and consequent delays in processing.

5.6. Chapter Summary

In this chapter, three specifically designed PAPs are compared against one another in various scenarios. It is analysed how each one of them performs in a variety of scenarios. The exponential PAP is selected for the hybrid self-organisation model due to consistently achieving the lowest TWT in the given scenarios.

It is also shown that no PAP can produce an optimal schedule in all possible scenarios. Therefore, it is considered sufficient to select this PAP as the governing policy for building initial schedules in the hybrid self-organisation model. In Chapter 6, this PAP is incorporated into the hybrid behaviour model. There, a blackboard agent firstly builds the initial schedule by applying the exponential PAP. If it determines that a PA is predicted to be tardy by only applying this PAP, it sends out a notification about it. The PA then seeks to negotiate with other PAs in order to achieve lower TWT.

Chapter 6 - Hybrid Self-Organisation Model for Mobile Robots in Large Structure Assembly

6.1. Introduction

In this chapter, the hybrid self-organisation model for mobile robots in large structure assembly is described. As discussed in Chapter 5, an integral part of the model is the priority aging policy (PAP). The policy governs how high each product agent's (PA) priority is at any instance in time. The resource agents (RAs) prioritise their tasks based on this order. In this chapter, the PAP is integrated into a self-organisation model.

The literature review reveals that scheduling for static environments should be done by centralised algorithms, while dynamic environments are best handled by distributed systems. Hence, the second hypothesis (Chapter 3) states that effective self-organisation in a wide range of possible scenarios can be achieved by developing two self-organisation behaviour models with fundamentally different architectures. The first one, the hybrid model, was expected to perform better in more static environments, where the foreseeable schedule is highly predictable and has rare disruptions. The other (decentralised) model is described in Chapter 7.

This model is considered hybrid because it consists of decentralised and centralised elements. PAs and RAs send all their information to the blackboard agent (BA) at the start. The BA then builds an initial schedule based on the PAP. It, therefore, predicts each PA's priority and how many RAs each PA will occupy at each instance in time. If it predicts that a job will be tardy, it notifies the respective PA about it. The PAs in this model attempt to solve these conflicts independently based on the information they receive from the BA. Thus, the model is less rigid than a fully centralised scheduling algorithm, enabling a part of the computational overheads to be diverted from the central entity (BA). The difference with the decentralised model is that there is no central entity in there and that model is fully distributed.

In Chapter 5, three PAPs are investigated to determine the most appropriate one for the hybrid model. It is shown in Figure 5-1 how PAPs can be used in this model to build an initial schedule for the allocation of RAs to PAs. The exponential PAP is selected for this model due to consistently achieving the lowest TWT out of the considered PAPs.

6.1.1. Model Requirements

The requirements of this model are to: produce an initial schedule, reschedule after breakdowns, reschedule after scrapping jobs, respond to launched rush orders and negotiate to reduce TWT. The model is focused on the self-organisation behaviour of the mobile system and not the supply of products. Thus, it is assumed that before a schedule is built the PAs already know when they will be loaded to their workstations by the transportation system. Also, the number of workstations is unconstrained because that is part of the product supply system. In the verification section, it is confirmed that the stated requirements have been met.

The further sections of this chapter provide an overview of the proposed hybrid multi-agent architecture, agent behaviours and their interactions; and finally, the model is verified.

6.2. Structure of the Model

The structure is the foundation of this model. As described in the literature review, agents in MAS can be cooperative or competitive. It is discussed that in fully distributed systems each agent only has incomplete information and should be competitive so as to achieve its own objectives. That behaviour can then be adjusted to meet the needs of the system in the desired way. However, in the hybrid architecture of this model, agents are cooperative because there is a central source of information and the system becomes optimised.

The agents are the core components of the model. The main agents for the functioning of the model are the product (PA), resource (RA) and blackboard (BA) agents. In addition to them, an entry agent (EA) is used to add and remove agents whenever necessary in the simulations. The behaviours of all agent types are described in detail in their respective sections below.

It is shown in Chapter 5 that the initial schedules built based on the PAP will not necessarily be optimal. Hence, a negotiation protocol is developed for the situations where the initial schedule has predicted that some PAs will be tardy. When such a situation occurs, the potentially tardy PA is notified by the BA. The mentioned PA then seeks to negotiate with other PAs with the aim of finding a solution with lower TWT. This way, the BA is subjected to lower computational overheads due to delegating a part of the scheduling process to the other agents.

The functions of the agent types are the same as described in Chapter 5: the BA builds a schedule based on the exponential PAP; if any of the PAs are predicted to be tardy, they seek to swap resources with other PAs in a way that minimises TWT; if not tardy or the swap deals have been done, PAs proceed to order as many RAs as necessary to be processed by due time by stating their current priority values (based on PAP) to RAs; the RAs accept the orders starting from the highest-priority PAs. The next sections describe each agent type and their respective behaviours.

6.3. Agent Behaviours

6.3.1. Product Agent (PA)

As soon as launched, each PA sends its location, working content, due time and tardiness cost to the BA. It then proceeds to request for necessary RAs by messaging them its priority at each time step in accordance with the PAP. At the same time, it listens for two types of messages: negotiation requests from other PAs and tardiness notifications from the BA. The flowchart for the PA behaviour is shown in Figure 6-1.

From the PAs' points of view, when they require additional RAs, they send their priority values *P(t)*$_j$ to RAs. By using Equation 6.1, each PA knows how many resources are required by it at every time *t* before due time d_j. R_{rj} is always rounded up to the next integer to ensure that there is no remainder at due time d_j. If the product has gone past the due time d_j; R_{rj} = *m* to finish as soon as possible.

$$R_{rj} = \frac{WC_j - WD_j}{d_j - t} \qquad \text{(Equation 6.1)}$$

When more than necessary RAs accept the order, the exceeding RAs are rejected. This way, the RAs are dictated by the priority values of PAs *P(t)*$_j$ and how many RAs each one requires. Where there is an abundance of resources, the PA with the highest priority receives acceptance from the leftover RAs. In case of equal priority values for several PAs, the PA with the lower number *j* in *P(t)*$_j$ is prioritised to resolve the conflict. In this situation, the PA with the highest priority gets more RAs than necessary. The priority value for these additional RAs is set to be 0 on the highest priority PA so that they are free to move to other PAs at any instance.

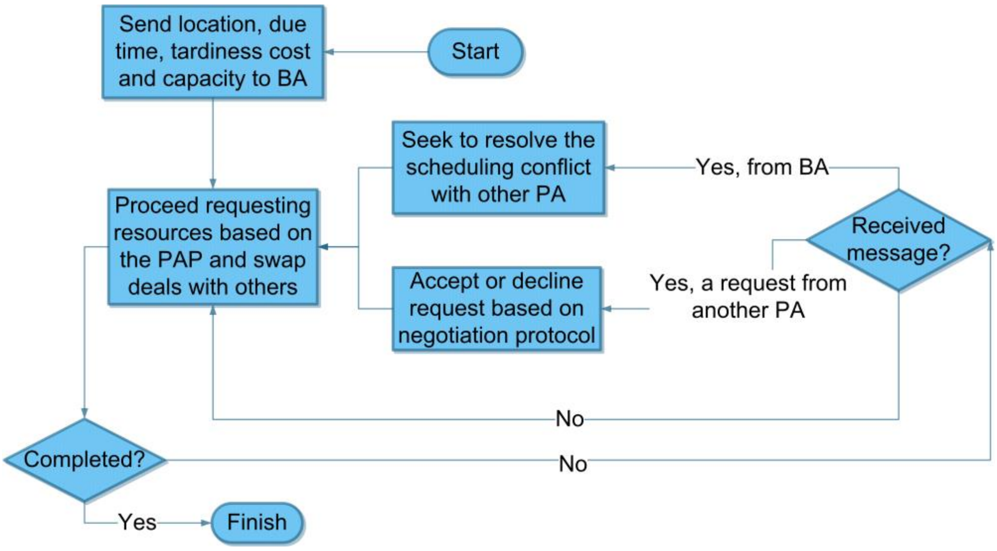

Send location, due time, tardiness cost and capacity to BA
Start
Seek to resolve the scheduling conflict with other PA
Yes, from BA
Proceed requesting resources based on the PAP and swap deals with others
Received message?
Accept or decline request based on negotiation protocol
Yes, a request from another PA
No
Completed?
No
Yes
Finish

Figure 6-1: The flowchart for the PA

Because the BA collects all the named properties of the PAs and the number of RAs, it can predict the schedules in advance. Using the PAP, it then calculates how many RAs each PA will order until the moment of completion of all jobs. In case any job is tardy in the predicted schedule, the BA sends the respective PAs tardiness notifications.

The notification includes the remainder that the tardy PA was predicted to not complete by its due time, PA names with later due times and the amount of work that will have been done on them by the due time of the notified PA (to ensure that sufficient RAs can be borrowed).

In case a message is received about scheduled tardiness, the PA seeks to negotiate with other PAs to find a way of reallocating RAs so that the TWT is reduced. Two outcomes are possible from this negotiation: borrowing RAs and no change. The negotiation protocol itself is described in section 6.4.2.

Borrowing RAs is the outcome where a PA_1 with an earlier due time receives some RAs until its completion from a PA_2 that has a later due time. Once completed, all of its freed RAs are returned to the PA_2 that agreed to lend them. The agreement to lend is achieved if the resultant TWT from this transaction becomes lowest. The initiator of this negotiation, PA_1, updates the BA, which then re-simulates the new schedule. PA_2 still orders RAs in the same way, but instead of ordering RAs to itself, it orders the RAs to work at PA_1 until the agreed working content is satisfied.

No change is the event where the borrowing of RAs has failed. This occurs when the result of the deal would result in higher TWT. This calculation is provided under section 6.4.2.

6.3.2. Resource Agent (RA)

The RAs are motivated solely by the priority values of PAs. As shown in Figure 6-2, each RA notifies the BA of its availability when it appears on the shop floor or when it completes jobs. It then listens to orders from PAs and proceeds by accepting the PA that stated the highest priority. If it does not get confirmation, then it accepts the second-highest and so on. If confirmed, it goes to work at that PA until either it gets completed or it hears an order from a PA with a higher priority. If no confirmation is received, it goes to work at the PA that has the highest priority, but listens to messages and will accept any order that comes from other PAs. Thus, the RAs fill the required numbers of resources R_{rj} at PAs starting from the highest-priority PA and if all of the required numbers are filled, they increase the work rates at the highest-priority PA.

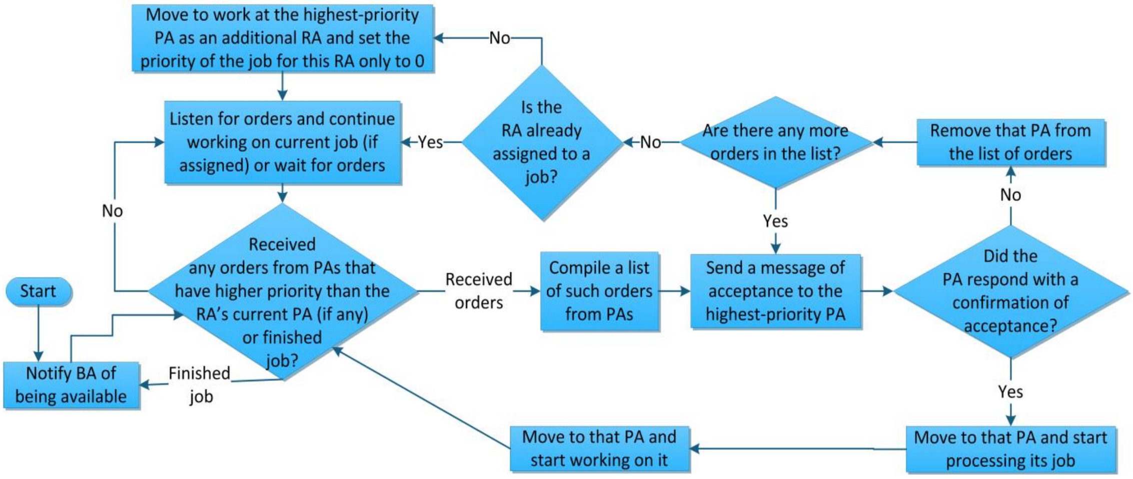

Move to work at the highest-priority PA as an additional RA and set the priority of the job for this RA only to 0
No
Listen for orders and continue working on current job (if assigned) or wait for orders
Yes
Is the RA already assigned to a job?
No
Are there any more orders in the list?
Remove that PA from the list of orders
No
No
Yes
Start
Received any orders from PAs that have higher priority than the RA's current PA (if any) or finished job?
Received orders
Compile a list of such orders from PAs
Send a message of acceptance to the highest-priority PA
Did the PA respond with a confirmation of acceptance?
Notify BA of being available
Finished job
Yes
Move to that PA and start working on it
Move to that PA and start processing its job

Figure 6-2: Flowchart for RA behaviour

6.3.3. Blackboard Agent (BA)

The BA is the source of information for all other agents. It stores information received from PAs and RAs; and passes to them in a processed form when necessary. To ensure that the initial schedule is checked and potentially tardy PAs are given the notification to negotiate, the BA processes the initial schedule based on the PAP before processing jobs commences. It can be argued that this way the transition between disruptions will be smoother. Certainly, the PAs without additional information could proceed by purely following the PAP while listening to notifications from the BA (a similar approach was shown in [102]). However, as discussed in the literature review, that approach is not optimised due to having limited knowledge of the environment. Thus, despite leaving the opportunity for modifications in the future, the more rigid approach was taken for this model to provide a stronger contrast with the decentralised model.

The behaviour of the BA is presented as a flowchart in Figure 6-3. When a simulation starts, the BA asks the Directory Facilitator how many PAs and RAs have been deployed. Each PA then informs it of its working content, due time, launch time, location and tardiness cost.

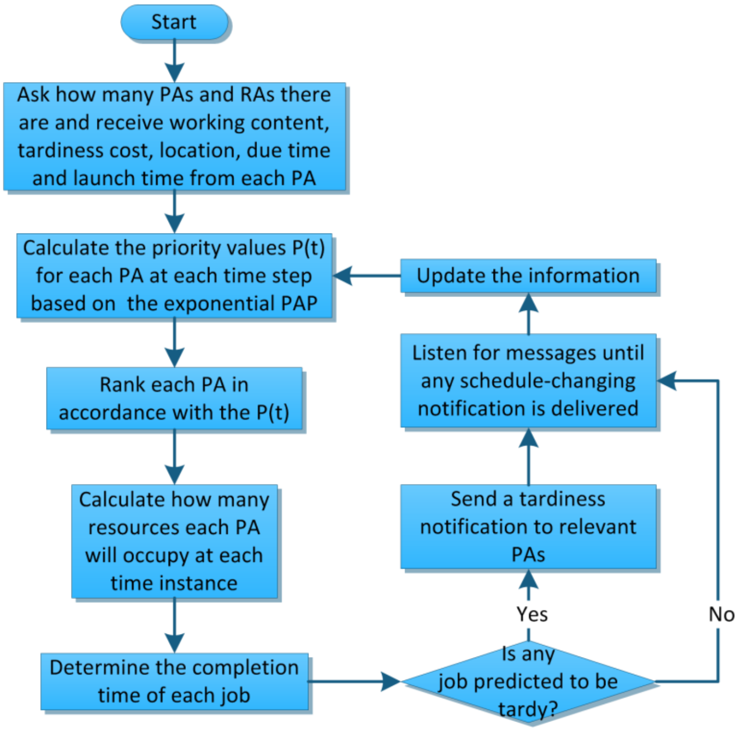

Start
Ask how many PAs and RAs there are and receive working content, tardiness cost, location, due time and launch time from each PA
Calculate the priority values P(t) for each PA at each time step based on the exponential PAP
Rank each PA in accordance with the P(t)
Calculate how many resources each PA will occupy at each time instance
Determine the completion time of each job
Is any job predicted to be tardy?
Yes
No
Send a tardiness notification to relevant PAs
Listen for messages until any schedule-changing notification is delivered
Update the information

Figure 6-3: The flowchart for the BA behaviour

As illustrated in Figure 6-4 the BA compiles a priority matrix in time steps for all PAs based on the PAP. The priorities at each time step are calculated using the exponential PAP, as described in Chapter 5. From the priority matrix, the BA compiles a ranking matrix. The ranking matrix ranks each PA based on its priority in relation to other PAs in time steps. Thus, by knowing the due times, working contents and tardiness costs of all PAs, the BA can predict how many RAs each PA will order without any resource swapping agreements. Hence, the BA determines when all of the jobs will be completed.

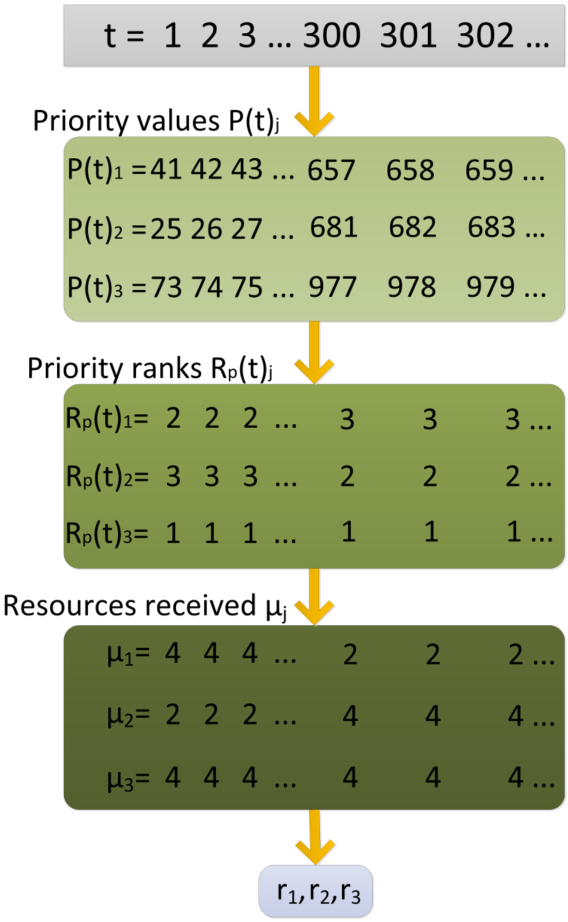

t = 1 2 3 ... 300 301 302 ...
Priority values P(t)j
P(t)1 =41 42 43 ... 657 658 659 ...
P(t)2 =25 26 27 ... 681 682 683 ...
P(t)3 =73 74 75 ... 977 978 979 ...
Priority ranks Rp(t)j
Rp(t)1= 2 2 2 ... 3 3 3 ...
Rp(t)2= 3 3 3 ... 2 2 2 ...
Rp(t)3= 1 1 1 ... 1 1 1 ...
Resources received μj
μ1= 4 4 4 ... 2 2 2 ...
μ2= 2 2 2 ... 4 4 4 ...
μ3= 4 4 4 ... 4 4 4 ...
r1,r2,r3

Figure 6-4: An illustration of the BA's calculations

If it predicts that no jobs will be tardy, the BA listens for any updates from agents (new launches, scrap, breakdown, etc.). If not, then a message is sent to the appropriate PAs for them to start negotiating with other PAs as described under section 6.4.2. If the BA receives at least one notification of a successful swap deal, it must recompile the schedule with the new details.

6.3.4. Entry Agent (EA)

The entry agent is a supporting agent for the numerical simulations in this model. Its purpose is to launch and destroy BAs, PAs and RAs. One BA and predetermined PAs and RAs are launched at the start of each simulation. Then, if a machine breaks down, then the respective RA notifies the BA and EA and gets removed from the simulation. EA creates RAs to correspond to additional machines getting deployed on the shop floor. When PAs have had their working content processed or scrapped, they notify the EA and get removed as well. Therefore, the EA exists to maintain the right number of agents in the simulation to be representative of the shop floor.

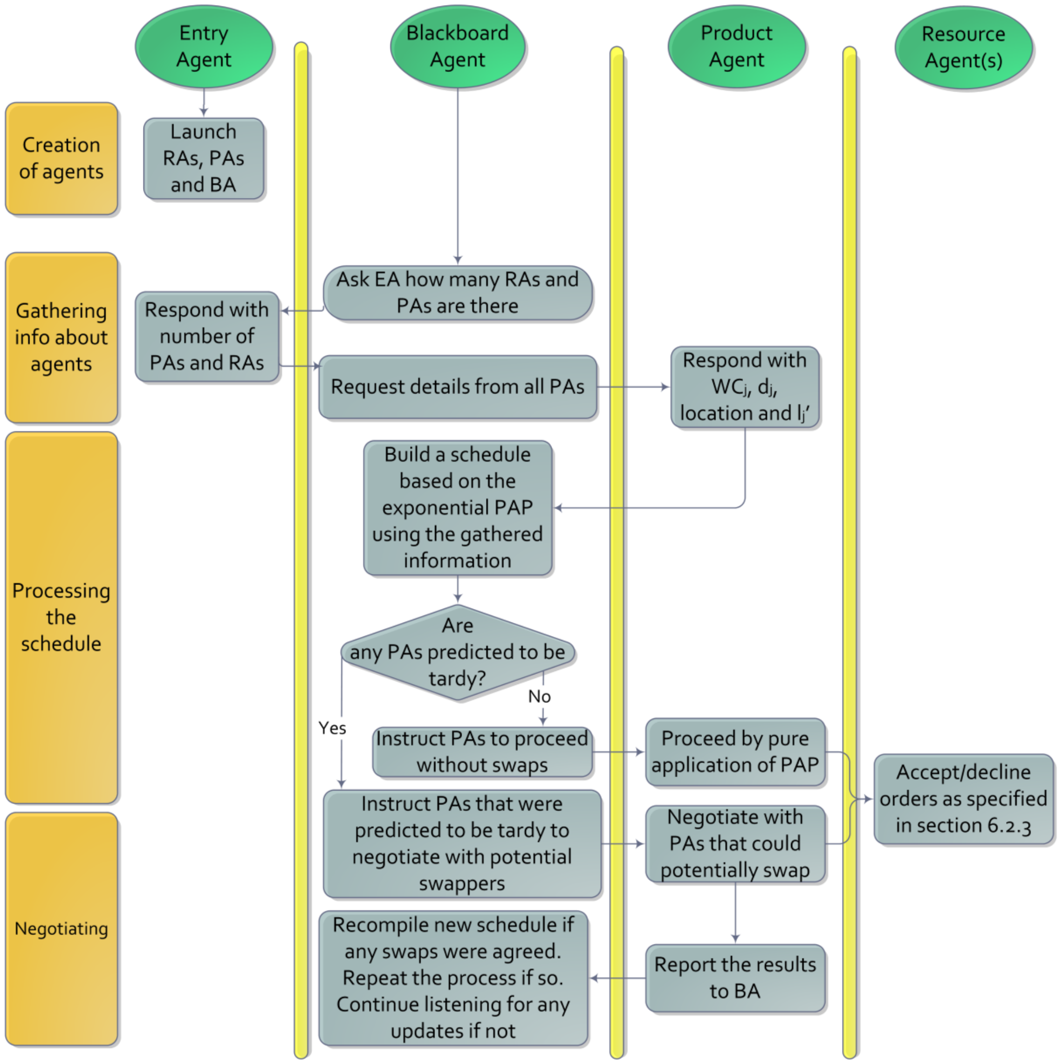
6.4. Agent Interactions

So far, each agent's purpose and behaviour have been described. This section describes the key interaction protocols that are used in the behaviour model. These occur during starting up and negotiating between PAs to lower TWT. They are explained in detail below. Trivial interactions like removal or additions of individual agents are neglected.

6.4.1. Start-Up

The flowchart for the start-up procedure is shown in Figure 6-5. First, the agents must be created. The system starts by creating the EA and manually inputting the parameters of all the other agents in it. The entry agent then launches the RAs and PAs, after which it launches the BA. This is to ensure that when the BA has been launched and initialised, it can ask for the exact number of PAs and RAs and request for the parameters of PAs. From those parameters, the BA compiles an initial schedule purely based on the PAP as was shown in Chapter 5. From that schedule, the BA can determine whether any PA will be tardy from the pure application of the PAP. If not, then it sends a message to PAs to proceed without negotiations. Whereas, if it determines that at least one PA is predicted to be tardy, it sends a message to it with details of its initial r_j, incomplete amount of working content at d_j, TWT and IDs of PAs that could potentially swap with it.

The PAs that could potentially swap it must satisfy two criteria, both of which are known to the BA at the moment of notifying: a later d_j and a completed amount of work on it at the requesting PA's d_j that is equal or greater than the shortage of the requesting PA's working content at that instance. Requesting a swap from PAs with earlier d_j-s is not sensible, because they would have only ordered the exactly sufficient amount of RAs throughout their production times anyway.



Entry Agent
Blackboard Agent
Product Agent
Resource Agent(s)
Creation of agents
Launch RAs, PAs and BA
Gathering info about agents
Ask EA how many RAs and PAs are there
Respond with number of PAs and RAs
Request details from all PAs
Respond with WC_j, d_j, location and l_j'
Processing the schedule
Build a schedule based on the exponential PAP using the gathered information
Are any PAs predicted to be tardy?
No
Yes
Instruct PAs to proceed without swaps
Proceed by pure application of PAP
Accept/decline orders as specified in section 6.2.3
Instruct PAs that were predicted to be tardy to negotiate with potential swappers
Negotiate with PAs that could potentially swap
Negotiating
Recompile new schedule if any swaps were agreed. Repeat the process if so. Continue listening for any updates if not
Report the results to BA

Figure 6-5: The flowchart for the start-up process in the hybrid model

The reasoning behind the second mentioned criterium is highlighted in Figure 6-6. It is an isolated example from Figure 5-9 of starving a product with an earlier d_j and a lower C_{tj} that theoretically has enough resources to be completed on time. The sufficiency of resources exists because for product A, d_j = 18,000 minutes and for product B, d_j = 20,000 minutes. At a manufacturing system's capacity of 1 unit of work per minute, the products (W_j = 10,000 for both) should ideally be completed on time. Whilst it is shown in Chapter 5 that many scheduling scenarios are handled very effectively by the exponential PAP, this particular scenario is not. Hence, the need for the swapping negotiation protocol arises. The successful negotiation should result in a swap deal between the two respective PAs that would divert the necessary amount of RAs from product B to product A so that the latter can be completed by its d_j. The timely completion of product A does not only free it from being tardy, but also frees the RAs that would otherwise be working at it between t = 18,000min and t = 20,000min, giving product B its “borrowed” resources back.

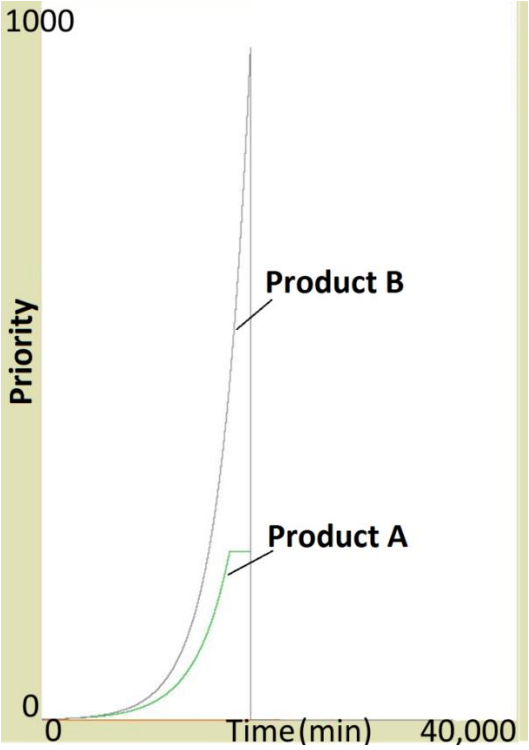


Figure 6-6: A sample priority plot in time for a starved product

6.4.2. Product – Product Negotiation

The negotiation between PAs is one of the key points in this model. As explained in section 6.3.1, the reason for its existence is the fact that the exponential PAP is not expected to produce a schedule with the lowest possible TWT in every scenario. Therefore, this interaction serves as a corrective measure. It is triggered when the BA notifies a PA about predicted tardiness. The negotiation protocol is presented in Figure 6-7. When a PA is notified about predicted tardiness, it sends a message to all PAs that have d_j-s after its own and have had a completed amount of work on them at the requesting PA's d_j that is equal or greater than the shortage of the requesting PA's working content at that instance. The message contains the sending PA's d_j, location, ID, predicted weighted tardiness, shortage at d_j, predicted release time r_j, and how much work has been done by that moment on the receiving PA (these were received from the BA). The d_j, location, r_j and ID are self-explanatory. The shortage is the working content that was predicted to not be met by the d_j. The responding PA considers sacrificing the shortage capacity (if it has that much work processed on it by then) by comparing the tardiness costs as a result of the swap if it were to be accepted.

If the r_j of product A is earlier or equal to the d_j of product B, then product B accepts the request and states that the resultant weighted tardiness WT = 0. This is because swapping the resources would not cause product B to be tardy at all. However, in the event that the r_j of product A is later than the d_j of product B, the calculation for the responding PAs is done by applying Equation 6.2:

$$WT_B = \left(r_{j,A} - d_{j,B}\right) * \ C_{t,B} \qquad \textit{(Equation 6.2)}$$

WT_B is the weighted tardiness that product B would be penalised by if it accepted the swap request. If accepted, it would be finished at product A's initially predicted release time $r_{j,a}$, because the manufacturing system will still handle the same manufacturing capacity in the given amount of time, regardless of where exactly the RAs were working. Thus, the equation calculates how much individual weighted tardiness there would be for product B if the swap deal is accepted. If the weighted tardiness for product B WT_B is lower than the weighted tardiness for product A in the initial schedule, it means that the swap deal would reduce the TWT.

It is shown in Figure 6-7 how a requesting PA1 interacts with a responding PA2. PA2 represents any responding PA that received a swap request. Based on the logic above, it decides whether to accept or decline the requested swap deal. The requesting PA1 waits for responses from all requested PAs and chooses the deal that results in the lowest TWT, disconfirms all other responders and notifies the BA that the specific deal has been reached. If PA1 does not receive any acceptations, then it notifies that BA that negotiations have failed and there is no need to rebuild the schedule.

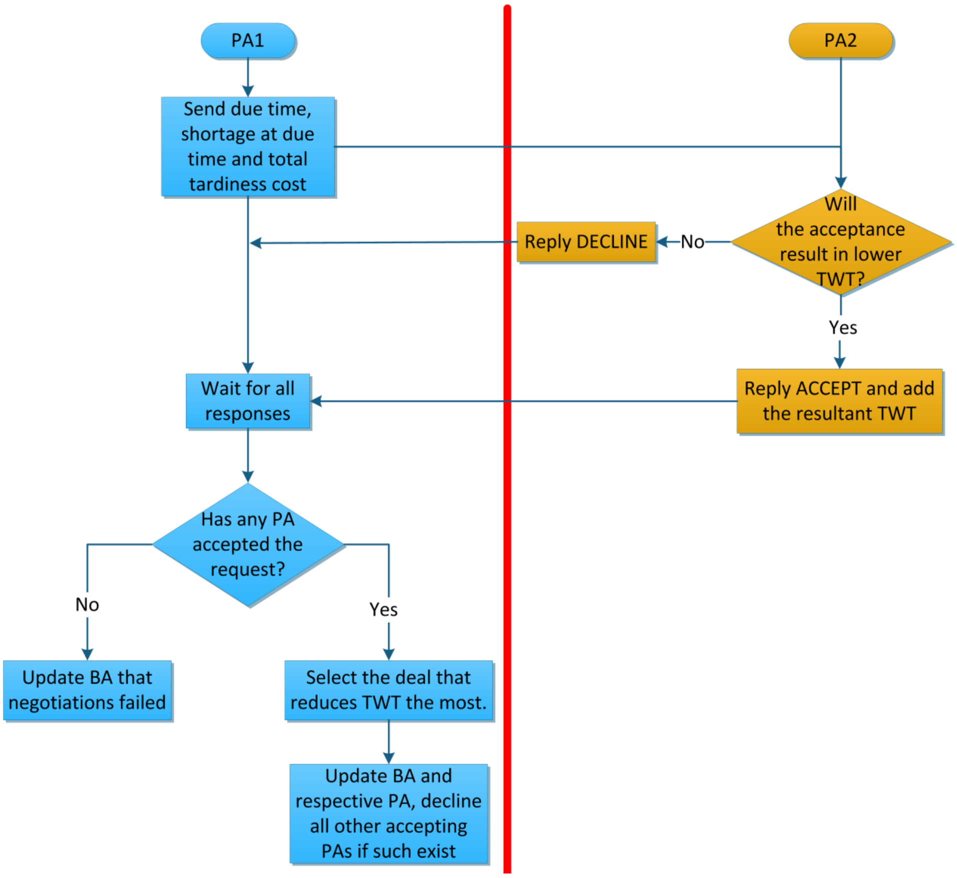

PA1
PA2
Send due time, shortage at due time and total tardiness cost
Will the acceptance result in lower TWT?
No
Reply DECLINE
Yes
Reply ACCEPT and add the resultant TWT
Wait for all responses
Has any PA accepted the request?
No
Yes
Update BA that negotiations failed
Select the deal that reduces TWT the most.
Update BA and respective PA, decline all other accepting PAs if such exist

Figure 6-7: The flowchart for the negotiation protocol for solving scheduling conflicts

6.5. Model Verification

In this section, the presented model is verified. Five experiments with straightforward input scenarios are set up to test whether the model produces the expected outputs. In these scenarios, each individual RA can contribute one unit of working content per time step. Each of the experiments was carried out with 5 RAs and 5 PAs, meaning that on average one unit of working content of each PA could be satisfied per one unit of time.

Essentially, each experiment requires the model to compile an initial schedule or reschedule (if any PA was predicted to be tardy in the initial schedule). The rescheduling mechanism itself is the same; however, the communication mechanism for each scenario differs and requires separate verification. The setups and individual weighted tardiness values for the experiments are shown in Appendix B.

Because this is a verification section, the interest of the experiments is to assess whether the model functions as intended. Thus, there is no interest in testing the model under challenging scenarios or constraints. Instead, the complexity of the problems is kept low so that they are intuitive to the reader and it is easier to predict the desired outcomes. The experiments below verify that the model builds an initial schedule, responds to a resource breakdown, responds to scrapping a product, handles a rush order and overcomes PA tardiness by triggering negotiation. Therefore, these experiments verify that the model can not only build a schedule but can also handle changes in the numbers of products and resources; and successfully use the negotiation protocol as described in section 6.4.2.

6.5.1. Experiment 1: Building a Schedule

The first experiment is set up to test the most basic function of the model: building a schedule. Each of the jobs has W_j = 50,000, d_j = 50,000 minutes (ARC = 1) and is successfully finished at t = 50,000min (TWT = 0). The schedule is illustrated in Figure 6-8. Due to not having any disruptions or tardiness, the model does not need to proceed with any negotiations nor rescheduling (as is the case in the next experiments).

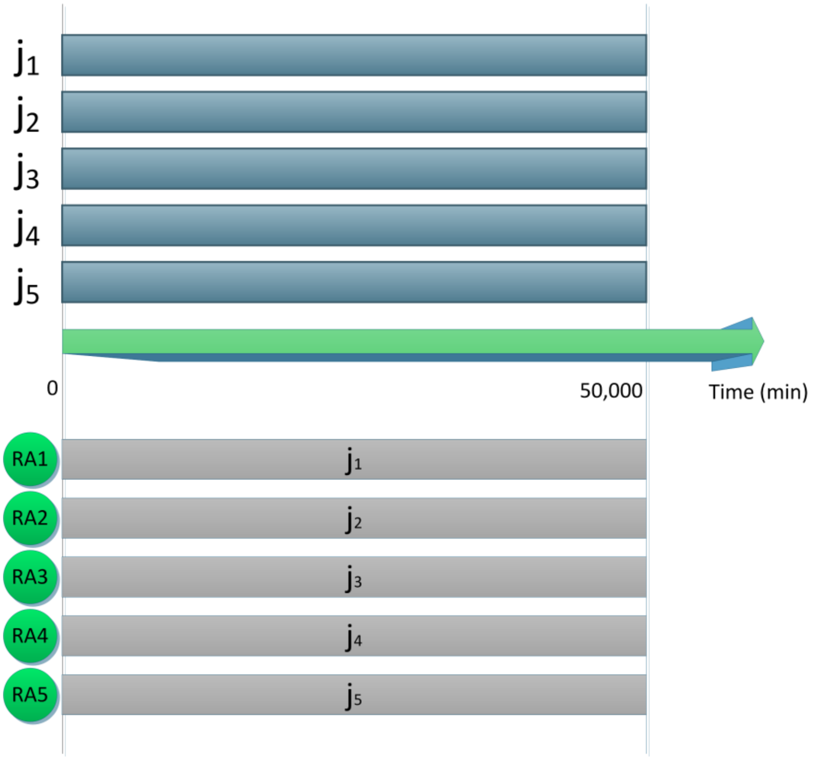


Figure 6-8: Illustration of the schedule for the jobs and RAs in the first experiment

6.5.2. Experiment 2: Responding to a Resource Breakdown

In this experiment, the model encounters a resource breakdown at *t* = 25,000min. The inputs to the experiment are the same as in the first experiment apart from the tardiness costs. The resource breakdown causes the system to have a shortage of resources. As designed in the model, the job with the lowest tardiness cost is then delayed so that the lowest possible TWT can be achieved.

The resultant schedule for this experiment is illustrated in Figure 6-9. It is shown that RA3 breaks down and is unable to proceed. Because j_3 has a higher tardiness cost and consequent priority, the model causes RA1 to change jobs from j_1 to j_3. The resource is diverted from j_1 because it was the job with the lowest priority. Hence, all of the other jobs were completed at *t* = 50,000min and the freed RAs worked together on j_1 after that.

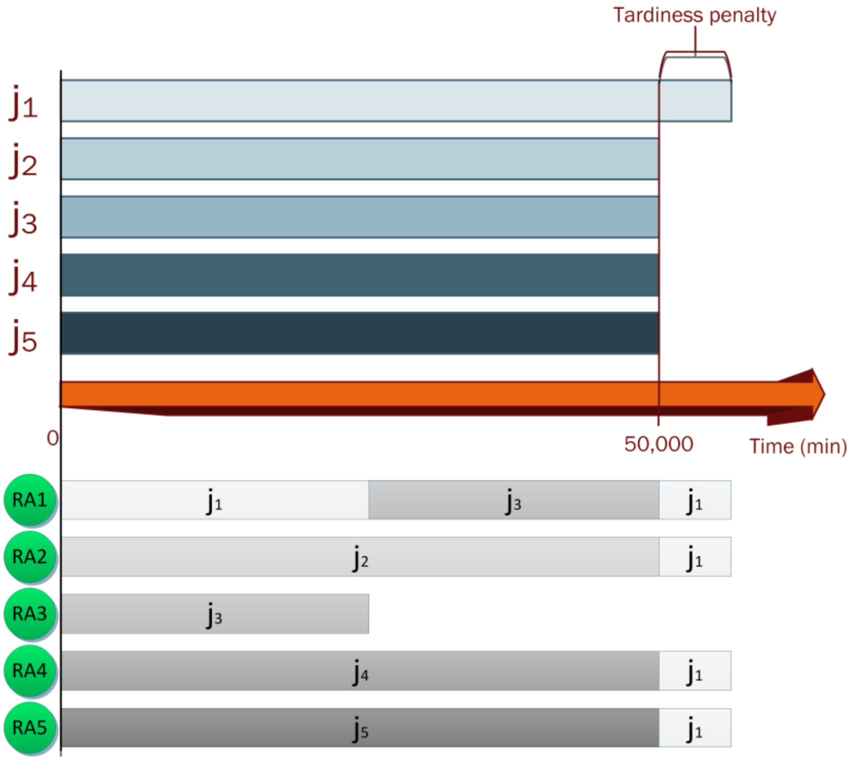


Figure 6-9: Illustration of the resultant schedule for the jobs and RAs in the second experiment

6.5.3. Experiment 3: Responding to Scrapping a Product

In this experiment, a product is scrapped at t = 25,000min. This means that work was done on it until that moment in accordance with the PAP. After the scrappage, an RA is freed to work on other jobs and consequently, they are finished earlier. The resultant schedule for this experiment is illustrated in Figure 6-10.

At t = 25,000min, RA4 is freed and moves to work at j_5. Due to additional resources, j_5 is finished before its d_j and two RAs become available for work. RA4 and RA5 then move to j_2 and j_3 respectively to finish those jobs early as well and then finish j_1. Therefore, the scrapped product reduces the total working content and the resources react accordingly to finish the rest of the products faster.

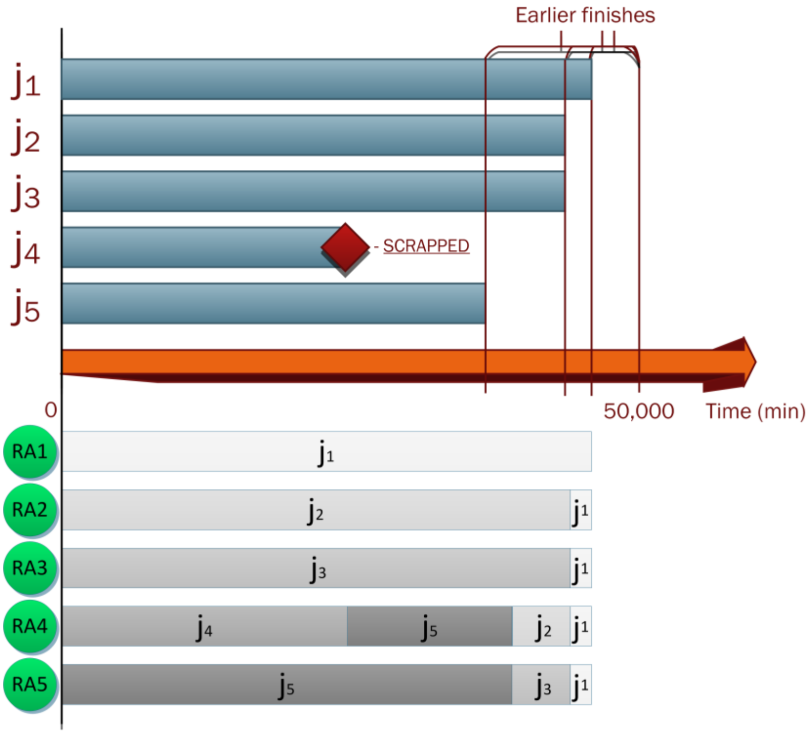


Figure 6-10: Illustration of the schedule for the jobs and RAs in the third experiment

6.5.4. Experiment 4: Handling a Rush Order

This experiment is set up to test whether the model can handle a rush order. The entry agent launches a rush order (j_6) at t = 10,000min with a due time of t = 45,000min. Consequently, the model deviates some of its RAs away from the regular jobs causing them to be tardy. The resultant schedule is illustrated in Figure 6-11.

RA1 and RA2 pause j_1 and j_2 respectively to complete the rush order on time. After completing it at t = 45,000min, they return to their paused jobs. Because of the rush order, those are not finished by their d_j-s. RA3, RA4 and RA5 are freed at t = 50,000min and then move to the tardy jobs. Due to a higher tardiness cost, these RAs first go to complete j_2 and finally j_1. As a result, the jobs with the lowest tardiness costs are tardy in order to complete the rush order on time.

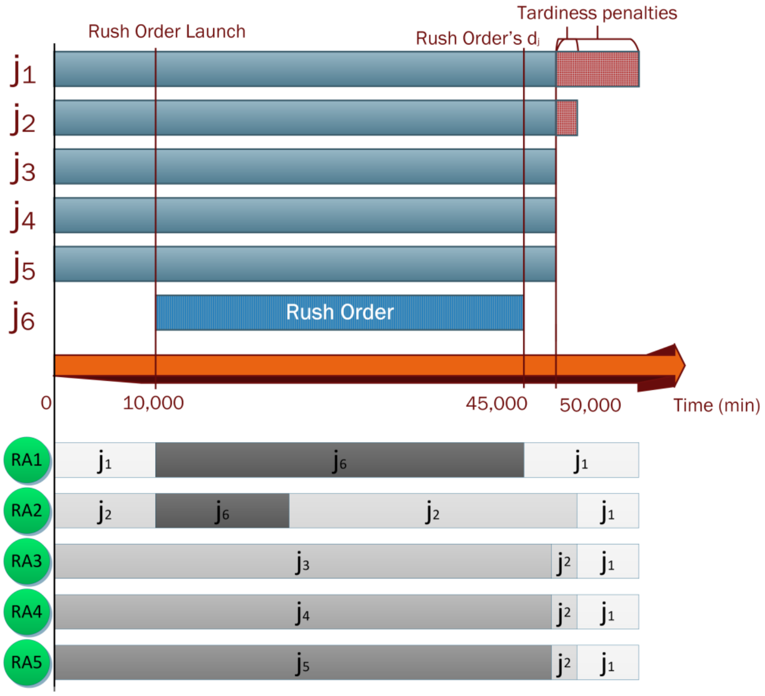


Figure 6-11: Illustration of the final schedule for the jobs and RAs in the fourth experiment

6.5.5. Experiment 5: Negotiation of a Tardy PA

The fifth and final experiment is carried out to verify the negotiation behaviour when a job is tardy. The experiment is set up in a way that causes the job with a low tardiness cost and earlier due time to be tardy despite mathematically having enough RAs to be completed on time (shown in Figure 6-12). This is because the $C_{t,1}$ is much lower than C_{tj}-s of other jobs and d_1 = 49,000min, whilst d_j = 50,000min for the rest of the jobs. Thus, j_1 is due earlier than others, but finishes at the same time as them and is therefore tardy.

The PA that is responsible for j_1 then starts the negotiation process with other PAs to see if the tardiness can be eliminated. Indeed, as the results in Figure 6-13 show, j_1 is accelerated and finished well before its due time while not having a negative impact on any other jobs. This is because the swap deal causes RA3 to be diverted to j_1 until its completion at t = 25,000min. After that, RA1 and RA3 are freed and moved to work at j_3 until its completion at t = 50,000min. The rest of the RAs and jobs are not affected by the swap deal. Thus, it is verified that when the BA predicts tardiness, it triggers a negotiation process which has been verified as well.

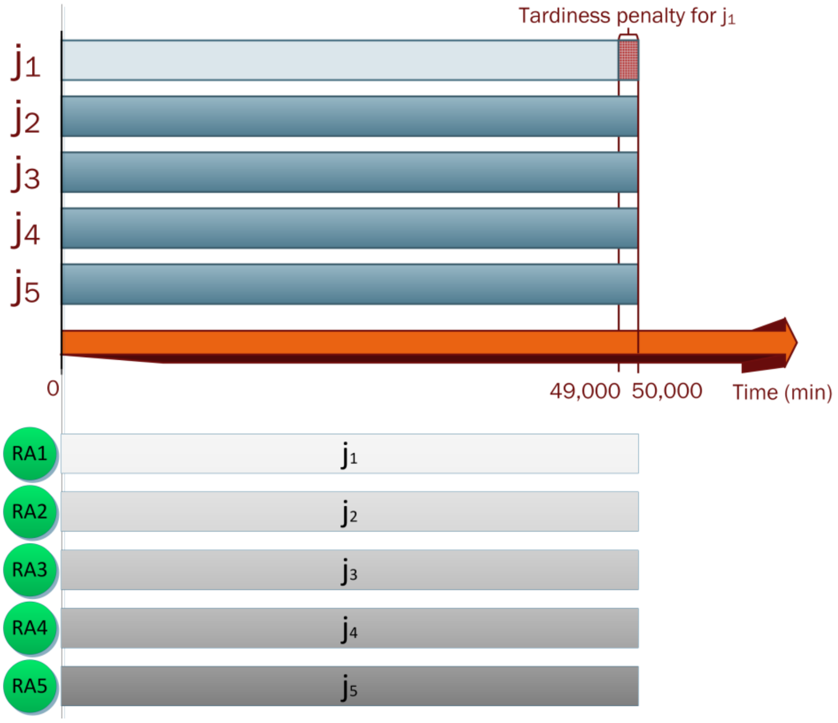


Figure 6-12: Illustration of the initial (purely PAP-based) schedule for the jobs and RAs in the fifth experiment

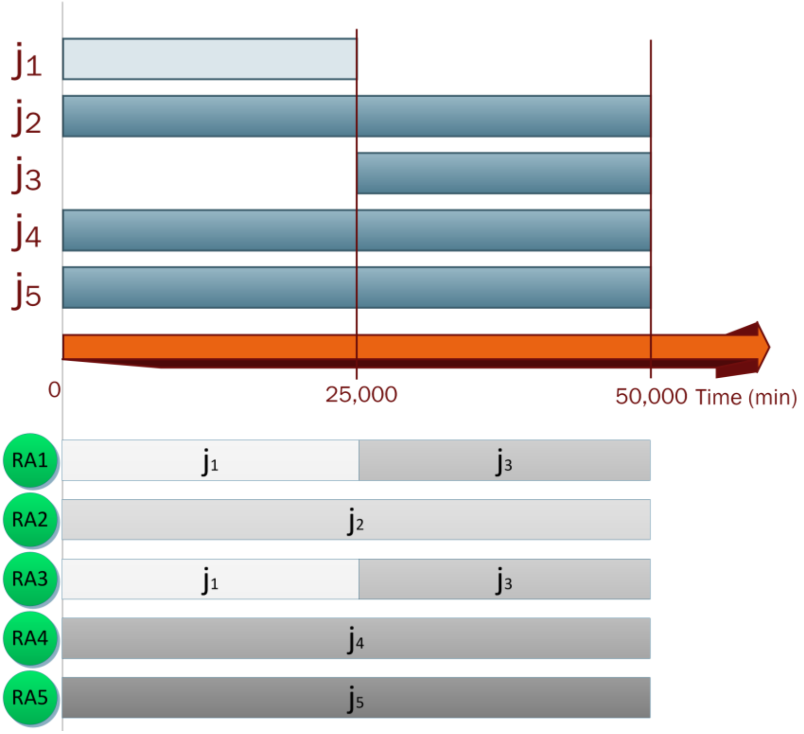


Figure 6-13: Illustration of the final schedule for the jobs and RAs in the fifth experiment

6.6. Chapter Summary

In this chapter, a hybrid model for self-organisation of mobile robots in large structure assembly is developed based on the PAP from Chapter 5. It consists of a blackboard agent, resource agents and product agents for operation. In addition to them, an entry agent s included to create and kill operating agents as and when necessary. The model is focused on the behaviour of the mobile system and not at supplying products to it.

The purposes and behaviours of each agent type are described and verified in five experiments. In combination with the exponential PAP that is developed in Chapter 5, the contribution of this chapter is the hybrid agent-based self-organisation model. The verification scenarios are not designed to be complex in order to be easily followed by the reader. Nevertheless, the experiments include scenarios where agents are added or removed and where the initial schedule based on the PAP requires improvement.

In Chapter 7, the development of the decentralised model for the same purposes is presented. Due to its fully distributed architecture, it was expected to respond to disturbances faster than the hybrid model, however, achieve worse results with respect to minimising the TWT. The capabilities of these models in terms of quickly responding to disturbances and minimising TWT are studied in Chapter 8.

As mentioned at the start of this chapter, in the event of any changes on the shop floor, the BA must recompile the whole schedule. This may not be feasible in real manufacturing environments, because the rescheduling can consume considerable time. Thus, one valid approach to that could be to proceed based on the PAP whilst the BA processes the information and trigger negotiations among PAs if necessary. However, as was discussed in the literature review, that approach could also lead to undesired results, as it would act as a fully distributed system for that period of time. Therefore, the shown hybrid model is designed in the described way to compare the performances of two contrasting architectures.

Chapter 7 - Decentralised Self-Organisation Model for Mobile Robots in Large Structure Assembly

7.1. Introduction

In this chapter, the decentralised model for the self-organisation of mobile robots is described. The model is intended for decision-making in the same environment as the hybrid model, where a blackboard agent is used for exchanging and processing information. The difference is that in this case the blackboard agent and PAP are omitted and the remaining agents are required to self-organise independently. Thus, this model is fully distributed because there is no central coordinating entity.

The model is developed to answer the research question of how to self-organise mobile robots in LSA. For the more static environments, the hybrid model is developed and verified in Chapter 5 and Chapter 6. In this chapter, a model for the more dynamic environments is presented. Therefore, in combination with the hybrid model, the expectation was that these two self-organisation models will enable appropriate self-organisation of mobile robots in a wide range of scenarios. By appropriate it is meant that the allocation of resources is be done in a timely and effective manner so as to not cause delays due to the scheduling process itself nor result in unreasonable schedules.

The model works by using a credit-exchange system: at launch, PAs are given a bankroll to complete the processing of their tasks. They use the credits from the bankroll to pay RAs for processing the tasks. The RAs are incentivised to work by credit offerings from PAs at each round of bidding. The bidding is done in the form of sealed bid auctions where each PA can post a single offering per round without knowing how much anyone else has offered. Using this method, the negotiation interactions are minimal and were expected to result in the fastest possible reactions to disturbances.

Like in many monetary interactions in the real world, the customer prefers to pay the least possible amount for the received services and the supplier prefers to earn the highest possible credits for their efforts. In this model, the PAs offer RAs to work at a fixed rate of credits per minute. The PAs start by placing lower bids than their maximum possible. This is done with the intention to firstly get some work done “cheaply” and secondly to preserve additional credits for the later stages of its production time (so as to be more competitive against other PAs). As the PAs become more pressured, their willingness to spend more increases and they can benefit from spending less in the earlier stages by having more credits left.

The RAs are driven only by the credit offerings. Regardless of whether they are already working or not, they listen for credit offerings from PAs. The PAs only offer a fixed amount of credits per minute of work to the RAs. However, the RAs also consider the moving distance between workstations. Whilst it has been shown in Chapter 4 that the moving times of mobile robots are negligible in LSA, the moving penalty in this model is still used as a dampener. This is to ensure that any offering made by PAs must be more than just “greater than” the current income of the RA. Thus, the movement penalty factor makes changing workstations “worth it” for RAs by ensuring that the newly accepted offer will compensate for the short trip. For simplicity, the time actual time spent moving is still neglected in the simulations.

7.1.1. Model Requirements

The model had the same requirements as the hybrid one, however with the exclusion of forward planning. Like in the hybrid model, the focus here is on the self-organisation behaviour of the mobile system and not the product supply mechanism. Because no agent has full knowledge of the environment, schedule inefficiencies cannot be predicted and the additional inter-agent negotiations are unnecessary. The expected advantage of the decentralised model is to be able to respond to any disruptions locally and with minimum effect on the rest of the system. Therefore there is no reason to plan schedules ahead and consequently, there are no forward planning requirements for this model. The model is required to allocate resources to products through negotiations at each round of bidding. This was expected to automatically translate into responses to all possible disruptions because this way a reassessment of the situation happens at each time step.

7.2. Structure of the Model

Because the BA is omitted in this model, the remaining acting agents are the product agents (PAs) and resource agents (RAs). The PAs attract RAs by offering virtual credits. The amount of credits assigned to each PA at launch time is called the bankroll (*b*) and it is the product of the tardiness cost C_{tj} and working content W_j as shown in Equation 7.1

$$b = C_{tj} * WC_j \qquad \textit{(Equation 7.1)}$$

This way, the priorities of PAs of the hybrid model are transformed into bargaining power (credits) in the decentralised model. The negotiation in this model is based on sealed bid auctions at each time step. In such auctions, every participant can bid only once per round without knowledge of how much anyone else had bid. The reason for this approach is to minimise the interactions between agents and achieve fast results. This is because each PA can only send its credit offering only once at every round of bidding and no additional negotiation can be done.

Based on the literature review, this model was expected to respond to disturbances faster than the hybrid model. Also, it was expected that adding and removing agents should be a smoother process, considering that it has only a local effect in the system as opposed to fully rescheduling in the hybrid model. The disadvantage of this model was expected to be the effectiveness of minimising TWT. Thus, the expectations were that any form of disruption on the schedules would be overcome faster than at the hybrid model (lower computational overheads); however, the resultant TWT on those schedules may not always be as low (not optimised).

The communication structure between PAs and RAs for this model is shown in Figure 7-1. Similarly to the hybrid model, RAs are interested in earning the highest possible credits for working. However, in this model, RAs also take into account a movement penalty when considering offers from PAs. This penalty is used as a dampener in the system to control how willing the RAs are to move from one workstation to another. Therefore, a higher credit offering may not always attract more RAs.

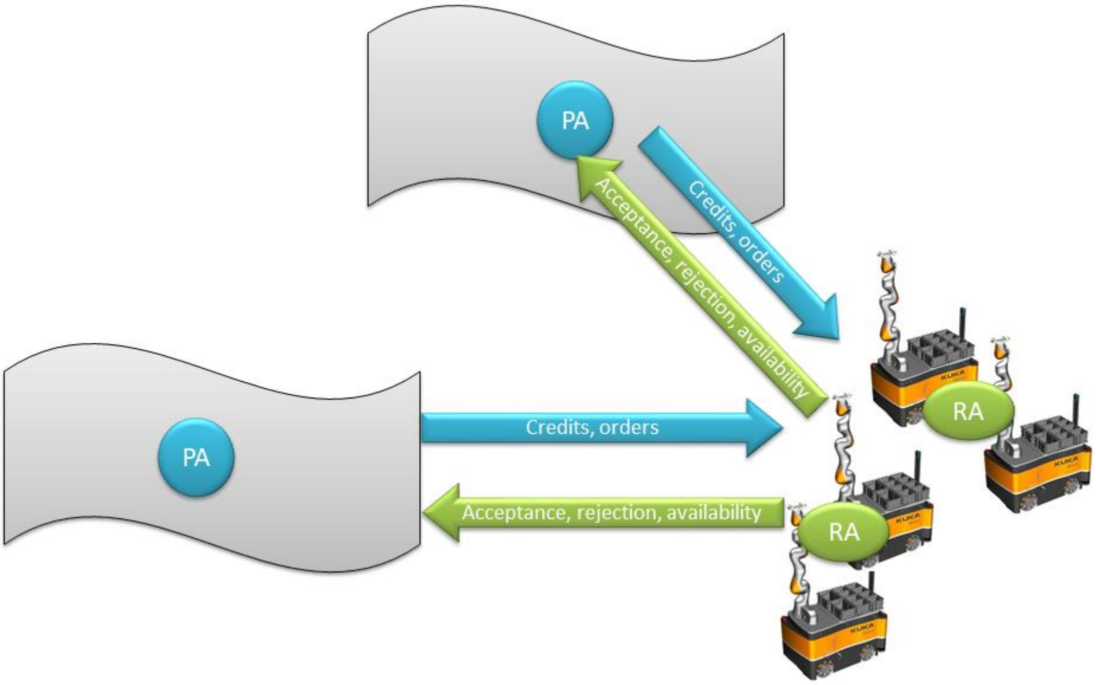

PA
Credits, orders
Acceptance, rejection, availability
PA
Credits, orders
Acceptance, rejection, availability
RA
RA

Figure 7-1: The decentralised model's communication structure

7.3. Agent Behaviours

7.3.1. Product Agent (PA)

The aim of each PA is to meet their job due times by offering credits their credits from their bankrolls b to RAs for processing the tasks. The credit offering formula of PAs is shown in Equation 7.2. The credit offering C_o is dependent on how many credits are in the PA's bankroll b, the remaining working content on it W_j and the bid gap G_b.

$$C_o = \frac{b}{WC_j} * (1 - G_b) \qquad \textit{(Equation 7.2)}$$

Knowing that $b = C_{tj} * WC_j$, the credit offering effectively becomes as shown in Equation 7.3 at the first bidding round for every PA.

$$C_o = C_{tj} * (1 - G_b) \qquad \textit{(Equation 7.3)}$$

The purpose of the bid gap G_b is to save some credits in the bankroll *b* for later stages when it may be necessary for PAs to bid higher in order to attract the necessary amounts of RAs. At the end of each bidding round, if any PA has not received as many RAs as necessary, then they decrease G_b by i_{gb} for the next round. The bid gap increment i_{gb} is calculated at launch time using Equation 7.4. The reason for this formula is that i_{gb} should be proportional to the starting bid gap G_b itself and to the time that it is expected to be active on the shop floor. The formula below is designed in such a way that a PA with insufficient RAs would decrease the bid gap G_b to 0 by the time when the PA has spent 80% of its time from launch to due time d_j. This should then ensure reaching high credit offerings at the late stages as was found to be effective in the hybrid model.

$$i_{gb} = \frac{(1 - G_b)}{(d_j - t) * 0.8} \qquad \textit{(Equation 7.4)}$$

This is discussed below. Otherwise, they continue the same payment amount to the already received RAs and offer 10% of the normal credit offering to other RAs in hopes of cheaply increasing the work rate.

7.3.1.1. The Spending Strategy of PAs

This is a measure of how conservatively or aggressively a PA is willing to spend their credits. As discussed in section 7.3.1, two factors influence this strategy: the bid gap increment I_{gb} and the starting bid gap G_b.

This behaviour can be explained using two examples. The first one is illustrated in Figure 7-2. There, at the first bidding round, it is shown how the PA chooses to offer a smaller amount than its maximum possible. The calculation for this is shown in section 7.3.1. The reason for doing so is to preserve credits for later stages to ensure higher chances of getting the necessary RAs then. The maximum credit offering itself is calculated by using Equation 7.5.

$$M_o = \frac{b}{wc_j} \qquad \text{(Equation 7.5)}$$

In Figure 7-2, it is shown how the maximum offer grows in relation to the actual offer, because the PA received sufficient RAs while using a bid gap. Thus, by saving some credits at early stages, it has the option of spending more at later stages if that becomes necessary.

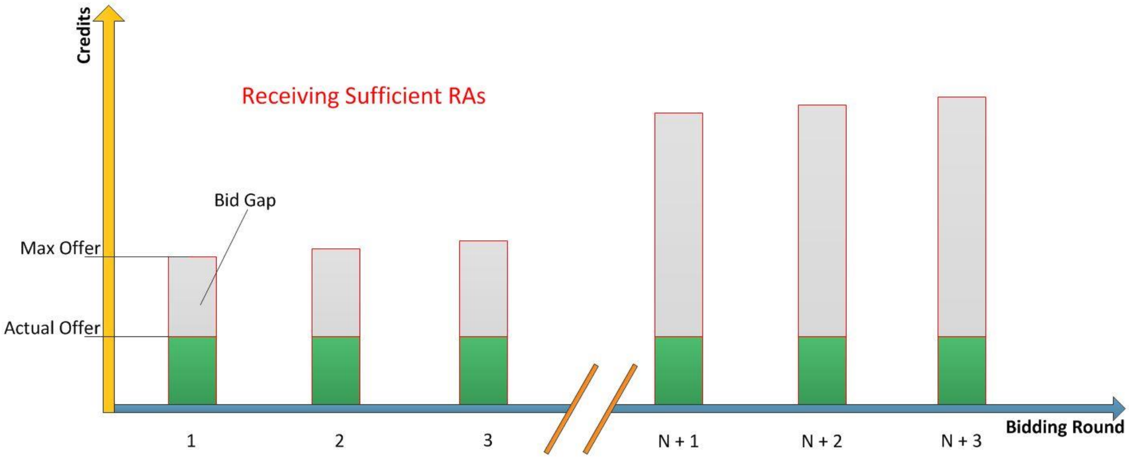


Figure 7-2: The PA bidding behaviour when receiving sufficient RAs

The opposite situation is illustrated in Figure 7-3. There, the PA has not got sufficient RAs allocated to it and therefore must increase the offerings. It starts with the same initial bid gap G_b, but then increases it by the big gap increment I_{gb} at each bidding round.

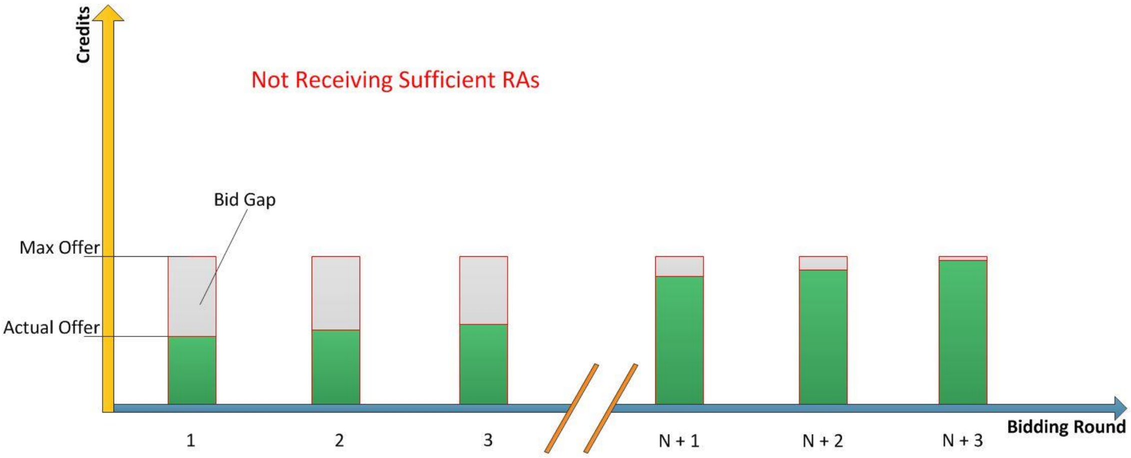


Figure 7-3: The PA bidding behaviour when receiving insufficient RAs

By using these control variables, it becomes possible to control how PAs spend their credits. Certainly, spending high credits, in the beginning, can attract a high amount of RAs, but that could cause problems to the PAs at later stages when they have relatively fewer credits left to spend. Contrarily, low offers in the early stages can lead to having insufficient RAs and consequent problems at late stages.

It must be noted that this spending strategy makes it impossible to completely run out of credits before the product is completed. The credits are only spent if an RA has worked at the PA for a time step (minute). According to Equation 7.2 and Equation 7.5, even if PAs started with their maximum possible offers M_o, the bankroll b would only reduce by that set amount per unit of working content that has been done on it.

7.3.2. Resource Agent (RA)

The behaviour of the RA is straightforward. It receives credit offerings from PAs, after which it calculates the offered value by using Equation 7.6.

$$V_o = C_o - d * P_m \qquad \text{(Equation 7.6)}$$

From the RA's point of view, it is unable to earn the offered credits until it moves the distance from its workstation to the new one. The distance between workstations is multiplied by the movement penalty factor (P_m) of the RAs to determine whether the offered value is higher than the current value. If so, then the RA moves to the new PA. This way, the P_m is a measure of how easily an RA can be motivated to change workstations. $P_m = 0$, if the RA is free.

7.3.2.1. The Movement Penalty Factor (P_m) of RAs

This factor is used to regulate how easily an RA can be enticed to move from one PA to another. The purpose of having this factor is to ensure that no two PAs would outbid one-another at consequent betting rounds, causing RAs to perpetually keep moving between their workstations. Thus, this factor works simultaneously as a dampener in the system and as a reward to entice mobile robots to move from location to another.

7.3.3. Entry Agent (EA)

The EA exists to launch and remove agents in simulations. It is also used as a time coordinating unit by waiting for all agents to notify it once they have finished their negotiations and consequently signalling for the next round.

When setting up a scenario, the EA is given the expected launch times of all PAs. As discussed in section 7.1.1. the assumption is that the expected product supply times have already been established. In case if there are no available workstations, the PAs are put into a queue and wait. Conversely, when PAs have been processed, they signal that to the EA so that the EA would remove them and free their workstations. Similarly, if RAs need to be added or removed, the EA processes that in an identical manner apart from having a maximum limit. Thus, the EA acts as the entity that controls the numbers of agents on the shop floor.

7.4. Agent Interactions

This section describes the various interactions that can occur in the model during operation.

7.4.1. Start-Up

The start-up is the most communication-intense event on the shop floor because it requires each agent to establish itself within the system. Such an event occurs after a full stop of production, (i.e. starting new production or after a blackout). The main actor in this event is the EA because it is responsible for launching PAs and RAs, and messaging them their details. Ideally, in a real scenario, this part would be different, because of product intelligence. There, the PAs would already possess all of the necessary information about themselves when being deployed. The RAs should know their behaviour parameters and be able to determine their locations independently as well. In the simulation environment, however, this is the first convenient opportunity to give them the information, which is why it is done by the EA. The general flowchart for this is shown in Figure 7-4. There, it is shown that at the first instance, the EA launches the RAs and PAs. After that, it messages them their details and they may start negotiating. This is where the EA plays no further role until it is needed for adding or removing agents from the shop floor.

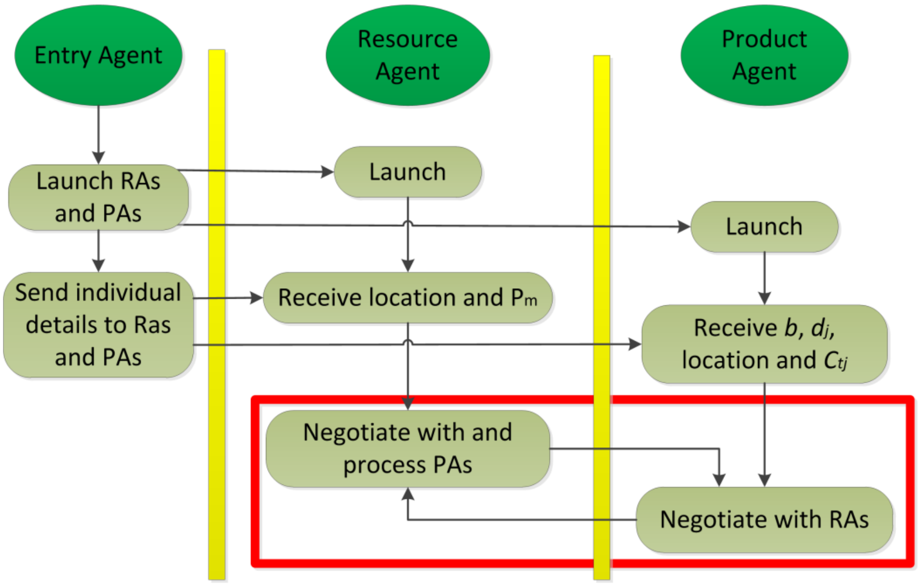


Figure 7-4: The swim lane diagram for the start-up procedure in the decentralised model

The section surrounded by the red rectangle is the main interaction protocol in this model. The detailed flowchart for it is shown in Figure 7-5.

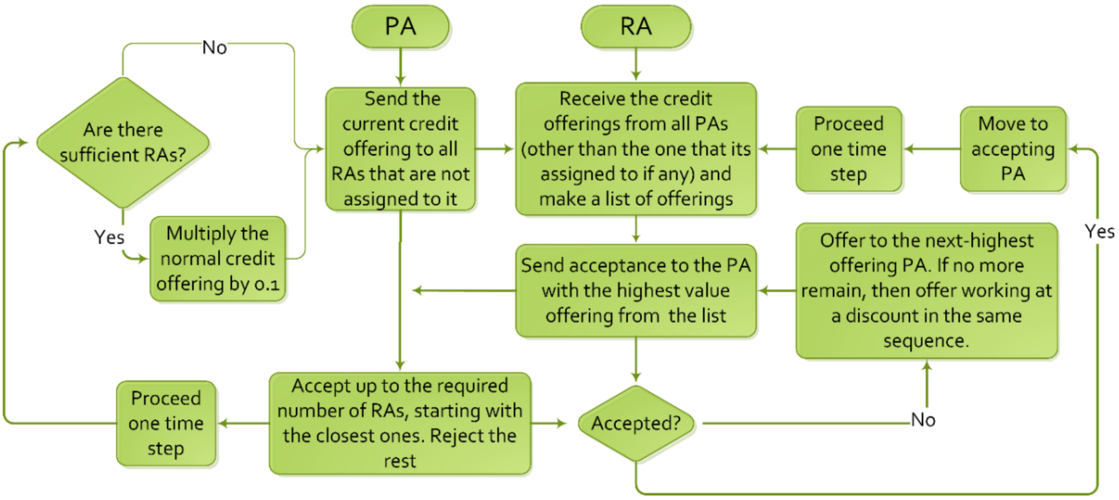

PA
RA
No
Are there sufficient RAs?
Yes
Multiply the normal credit offering by 0.1
Send the current credit offering to all RAs that are not assigned to it
Receive the credit offerings from all PAs (other than the one that its assigned to if any) and make a list of offerings
Proceed one time step
Move to accepting PA
Yes
Send acceptance to the PA with the highest value offering from the list
Offer to the next-highest offering PA. If no more remain, then offer working at a discount in the same sequence.
Proceed one time step
Accept up to the required number of RAs, starting with the closest ones. Reject the rest
Accepted?
No

Figure 7-5: The negotiation protocol between PAs and RAs in the decentralised model

At the start of each round of bidding, the PAs send their credit offerings to RAs. If the RAs calculate that any offered values are higher than what they currently have, then they send a message of acceptance to the respective PAs (starting with the best offering one) and add what the distance is between them. The PAs then accept as many RAs as they require to be finished on time (closest ones first) and reject the rest. The rejected RAs then offer themselves to the next highest PA in the higher offerors list if it exists and so on. The calculations for the credit offerings of PAs were shown in section 7.3.1.1 and the offers' value calculation for the RAs is shown in section 7.3.2. In both cases, where there is a conflict due to having multiple responders with equal value, the first one to respond is prioritised to overcome the conflict.

7.4.2. Handling Disruptions

In contrast to the hybrid model, the agents in the decentralised model handle disruptions without any mediation. Because there is no coordinating entity, the reactions to disruptions are intrinsic. Therefore, only the involved agents must be updated when a disruption occurs (i.e. if an RA breaks down, it notifies the PA whose job it was processing at that instance and the EA to get removed). The rest of the system proceeds as if nothing happened. The next sections describe how each considered disruption is met separately.

7.3.3.1. Responding to a Breakdown

To handle a machine breakdown, the compromised RA informs its PA that it is no longer working there and requests the EA to delete it. This way, the associated PA can take that into account in the next round of bidding. If the RA was not assigned to any PA, it is simply removed from the system without any effect on anything else until it is fixed. The swim lane diagram for this process is shown in Figure 7-6.

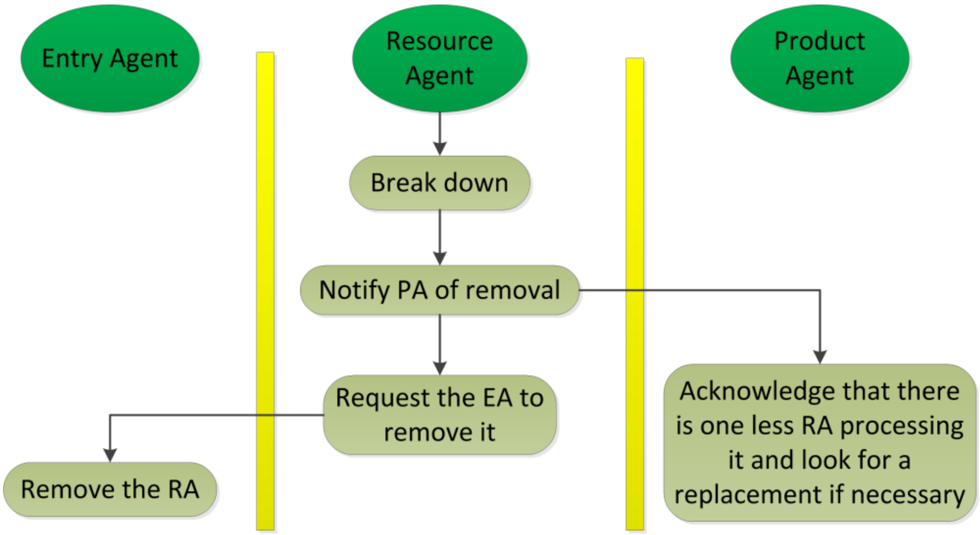

Entry Agent
Resource Agent
Product Agent
Break down
Notify PA of removal
Request the EA to remove it
Acknowledge that there is one less RA processing it and look for a replacement if necessary
Remove the RA

Figure 7-6: The swim lane diagram for a resource breakdown

7.3.3.2. Responding to a Rush Order

The rush order PA is launched out of schedule. From the production line management perspective, rush orders should only be accepted if it is financially beneficial to do so. Therefore, a rush order in this model can be treated like any other PA (see section 7.3.1), however, it is given more credits than the regular PAs due to its higher tardiness cost (it would not be a rush order without this). This results in a high likelihood of the rush order offering the highest amounts of credits to RAs. The exact values for those factors on a real shop floor are determined by the C_{tj} and d_j values of the rush order PA's job *j*. Therefore, in the context of sealed bid auctions, rush order PAs are bidders just like the regular PAs, but with the exception of being much wealthier.

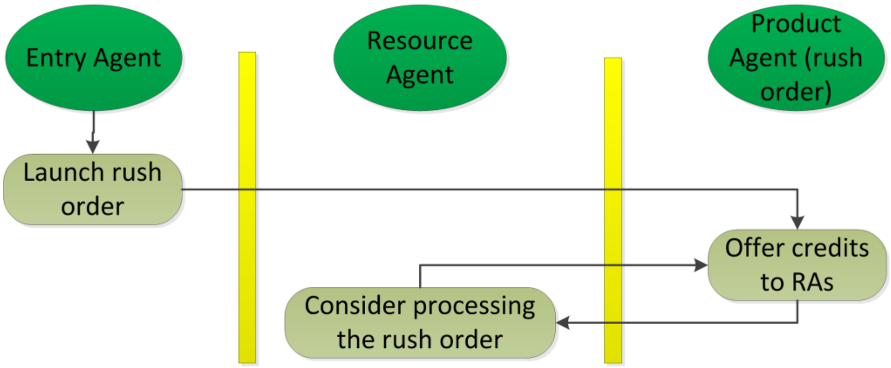


Figure 7-7: The swim lane diagram for a rush order launch

7.3.3.3. Responding to Scrapped Products

When a PA is scrapped, it stops its contract with all RAs, notifies the EA and deletes itself. The swim lane diagram for this is shown in Figure 7-8. The RAs can then accept the highest offered contract from other PAs without considering any movement penalties (as discussed in section 7.3.2).

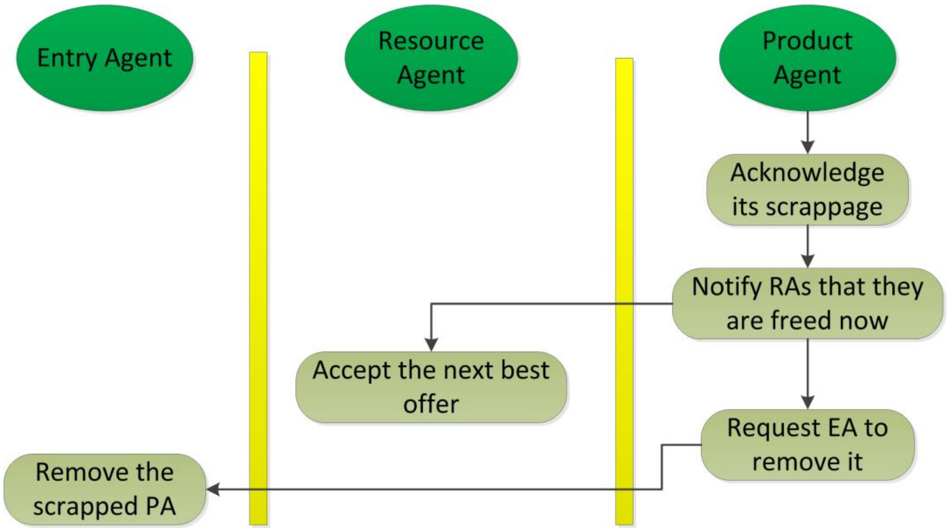

Entry Agent
Resource Agent
Product Agent
Acknowledge its scrappage
Notify RAs that they are freed now
Accept the next best offer
Request EA to remove it
Remove the scrapped PA

Figure 7-8: The swim lane diagram for a scrapped product

7.5. Production Scenarios

The production scenarios are affected by a number of factors that reflect the simulated situations on the shop floor. These are launch times, due times, tardiness costs and working contents for the PAs. These factors are described below.

Launch time of PA: This is the time from which onwards RAs may start working on the given PA. Before this, PAs do not make any offers.

Due time of PA: This is the time by which PAs should be finished. Failure to do so incurs a penalty of a tardiness cost per each time step when the PA is tardy.

Tardiness costs of PA: This is the cost factor that PAs get penalised by each minute after they fail to meet their due times.

Working content of PA: This is how much work must be done on a PA before it is considered completed.

7.6. Model Verification

In this section, the model is verified. The same experiments are carried out as in the verification of Chapter 6, but with the exception of the negotiation between PAs. This is because PAs do not get notifications of potential tardiness. The distances between the workstations are considered for the purposes of calculating the movement penalty of RAs, whereas the actual movement time is still neglected. The setup and individual resultant weighted tardiness values are presented in Appendix C. Just like in the hybrid model, the first experiment has exactly the sufficient number of RAs to complete all PAs on time and in the further experiments, different forms of disruptions happened to complicate the decision-making.

The following settings were set in the model for these verification runs:

Spending strategy of PAs: Starting $G_b = 0.9$.

Moving penalty factor of RAs: $P_m = 1$.

The shop floor plan is shown in Figure 7-9. There, workstations are marked as $WS_{x,\ y}$, where x and y represent columns and rows respectively. $WS_{1,1}$, $WS_{2,1}$, $WS_{3,1}$, $WS_{4,1}$ and $WS_{1,2}$ were allocated to PAs 1-5 respectively. In the rush order experiment, the PA with the rush order is set up at the location of $WS_{2,2}$. There is a distance of 60m between each adjacent workstation. For instance, $WS_{1,2}$ is located at 0m in the x-direction and 60m in the *y*-direction. The adjacent workstation $WS_{2,2}$ is located at 60m in the horizontal direction and 60m in the vertical direction. The distances affect the moving penalties for RAs and consequently the acceptance criteria for offers. Starting at the mobile robot rest area does not cause any penalties at the start, because the movement penalty for unassigned RAs is always 0. The detailed setups and results for these verification tests are shown in Appendix C.

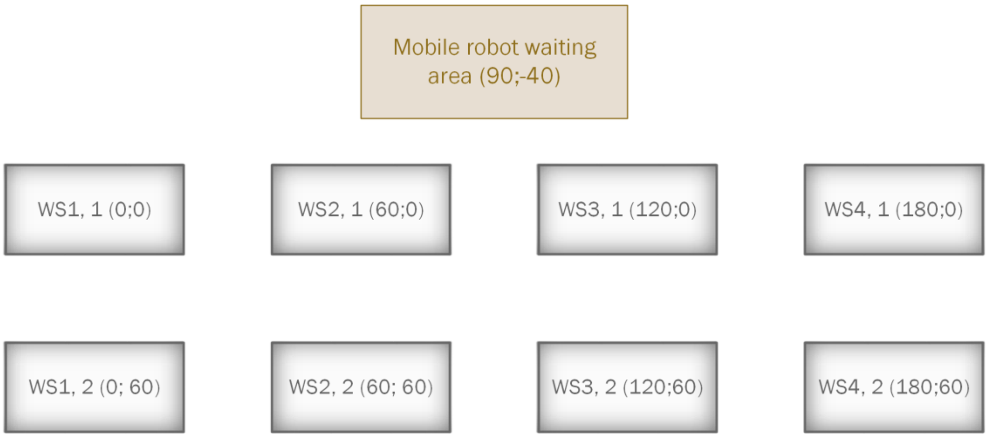


Figure 7-9: The shop floor plan for the validation experiments

7.6.1. Allocation of Resources

This experiment is carried out to verify whether the most basic function of the model is working. The results confirm that the model handles the resource allocation. Most jobs are completed slightly early and only PA 5 is exactly on time. The early finishers occurred due to the starting state where all PAs had no RAs allocated to them and all RAs being available. This smoothly transitioned into a balanced state where each PA had a single RA allocated to them and a self-correction in the late stages when close to due times.

7.6.2. Responding to Breakdown

In this experiment, a random RA is broken down at t = 25,000 minutes. The effect of the breakdown on the system is a shortage of resources. The model correctly sacrifices the PA with the lowest tardiness cost and completes all others without any tardiness.

7.6.3. Scrapping a Product

In this experiment, PA4 is scrapped at t = 25,000 minutes. This frees an RA and causes there to be an abundance of resources from that moment onwards. As a result, all PAs were completed before their due times.

7.6.4. Handling a Rush Order

In this experiment, a rush order is launched at t = 10,000 minutes. It has a d_j = 45,000min and 5 times greater bankroll b as each other PA. The model correctly completes the rush order on time, causing two other products to be delayed.

7.7. Chapter Summary

The decentralised self-organisation model for mobile robots in large structure assembly is introduced in this chapter. The various parameters and agent interactions that affect the model's behaviour are described.

As opposed to the strict priority rankings that are shown in the hybrid model, the decentralised model uses a sealed bid auction mechanism to enable PAs to incentivise RAs to process the tasks on them. In this case, the strict priority rankings are replaced with economic considerations. This is because the starting amount of credits in a PA's bankroll is linearly proportional to the PA's tardiness cost. As a result, the most important products have the highest bargaining power in the bidding process. The spending strategy and movement penalty factors allow the behaviour of the model to be modified in several ways. The effects of this are described in detail in Chapter 8. The model is successfully verified in the same kinds of experiments as the hybrid self-organisation model in Chapter 6.

Chapter 8 - Validation of Doctoral Thesis

This chapter's purpose is to report the validation of the work done in this doctoral thesis. The work is based on two hypotheses. The first hypothesis is stated as follows: "**An appropriately controlled mobile system is more utilised, resilient and has greater control over product delivery times in dynamic scenarios of LSA than traditional, dedicated automation systems with identical working capacities.**" This hypothesis did not require a validation per se because it required a comparison of systems to justify the further work in this thesis and beyond it. In Chapter 4, the comparison was carried out through a set of simulations on representative models of both system types. Although the appropriateness of the used control model in these simulations can be argued, the advantages over fixed systems were proven. It was expected that with more fitting control models, the performance of the mobile system can be further improved.

The mobile system consistently achieves higher utilisation because it has the ability to reallocate freed resources to any product that needs to be processed. The resilience of the mobile system is not necessarily better than at the fixed systems if it is compared strictly by definition. Resilience is defined as the ability to mitigate or absorb the impact of a disruption and return to normal operating conditions. This was shown to be true with breakdowns [45] because mobile robots can be replaced almost effortlessly. However, it was found in this work that when rush orders arrive, the decisive factor is the spare capacity in the system. As additional working content arrives out of schedule, the movement freedom per se does not give an advantage in terms of returning to normal operation conditions. The mobility does, however, enable the manufacturer to have control over the completion time of the rush order by choosing how many resources to allocate to it. This is clearly an advantage and part of the hypothesis where resilience is concerned can be partially validated. Similarly to rush orders, the work rates can be controlled on regular products as well. This translates to having more control over the delivery time of each individual product, which can be of great importance for manufacturers. The experimental runs also show that the moving time proportions are negligible in comparison to the processing times. Thus, assuming that the tasks can be done to the necessary standard by the mobile system, the named advantages can be achieved at a negligible penalty due to mobility.

8.1. Comparison of Self-Organisation Models

The second hypothesis was stated as follows: "**The agent behaviour model for self-organising mobile robots in LSA based on the hybrid architecture will exhibit better self-organisation schedules but lower responsiveness than the model based on the decentralised architecture.**" The validation of this hypothesis firstly requires the behaviour models. Those are developed in Chapter 5, Chapter 6 (both for the hybrid model) and Chapter 7 (decentralised model). Whilst the functionality of both models is individually verified in Chapter 6 and Chapter 7, the validation of the stated hypothesis follows in the further sections of this chapter. The cooperative behaviour model from Chapter 4 is used as a benchmark for the results.

8.1.1. Problem Formulation

8.1.1.1. Job Shop Model

The scheduling problem, in this case, is different to the one considered in Chapter 4. This is because the comparison of systems requires an assessment of the fundamental differences between physical systems. Thus, the focus is on identifying the operational differences due to physical limitations. However, the comparison in this chapter compares the performance of behaviour models. The difference is that instead of gaining generic penalty points for being tardy, the time spent in tardiness is multiplied by the tardiness cost of the tardy products. This makes the simulation more realistic because in a real manufacturing environment there can be different importance to completing each individual product on time. Consequently, the tardiness cost in the simulations is linked to the economic harm that tardiness could cause in a real manufacturing environment.

Based on this scenario, the composing elements of the system and their expected relations can be clearly defined. Products $J_{1..n}$ from the set J_n with working content W_j of several hours of single-machine processing are loaded to workstations $WS_{i...1,2}$. Once loaded, mobile resources $M_{1...m}$ may move to them and start processing the products until completion at time C_j and subsequent unloading of the products.

The following constraints apply to the model:

$$S_j \geq 0,\ l_j \geq 0,\ u_j \geq 0,\ l_j' \geq 0,\ u_j' \geq 0,\ t_{l,\,j} \geq 0 \quad \forall j \in J \tag{1}$$

$$|C_{j,\ min}| = t_{l,\,j} + L_j + p_j\ \forall\ j \in J \tag{2}$$

$$S_j \geq l_j' \quad \forall\ j \in J \tag{3}$$

$$C_j = S_j + p_j \quad \forall\ j \in J \tag{4}$$

$$u_j \geq C_j \quad \forall\ j \in J \tag{5}$$

$$m_{j\ max} = m = 4 \tag{6}$$

$$(l_j,l_j') =]u_j,u_j'[,\ (l_j,l_j') =]l_{j+1},l_{j+1}'[\ \ \forall\ j \in J \tag{7}$$

$$(u_j,u_j') =]l_j,l_j'[,\ (u_j,u_j') =]u_{j+1},u_{j+1}'[\ \ \forall\ j \in J \tag{8}$$

The constraints are identical to the ones shown in Chapter 4, with the exception of excluding the ones specific for the fixed automation system. The first constraint (1) ensures that no activity (S_j, l_j, u_j, l_j', u_j', $t_{l,\,j}$) can take place before the simulation. The second constraint (2) specifies that the earliest possible completion time of any job $C_{j,\ min}$ is the added sum of time taken to load (L_j) and process (p_j) a job J_j after it was launched at $t_{l,\,j}$. This way, the unloading time is not considered for the TWT calculations, as that is dependent on the crane system's availability. The due time $t_{d,\,j}$ for RAs is set without considering the unloading as well. Constraint (3) defines that a job can only start being processed at time S_j after it has finished loading to a workstation at time l_j'. The completion time C_j in constraint (4) is the sum of the starting time S_j added to the processing time p_j for each agent. Under constraint (5), for each job, the unloading may be started at time u_j only as soon as the processing on that product has been finished at time C_j. Constraint (6) ensures that the maximum number of resources $m_{j\ max}$ that can be allocated to processing a single job is the full number of resources, which was set to 4. The crane system's availability is defined under constraint (7). It establishes that between the start l_j and finish l_j' of loading job j, there can be no unloading (u_j, u_j') or loading of other jobs (l_{j+1}, l_{j+1}') and vice-versa under constraint (8).

This provides an overview of the expected operation of such a system and establishes its boundary conditions. However, it does not establish how one can plan for such an environment. Thus, the simulations and analysis in this chapter should provide an overview of how either self-organisation model behaves in a number of scenarios. To achieve a conclusion, two key performance indicators are used that reflect the needs of such a system:

1) The TWT: This is a measure of how efficiently a model plans its processing of products with respect to due times and tardiness costs. Generally, there are negative consequences when a product is completed later than its due time. In this work, each product has a tardiness cost ($C_{t,\ j}$) which counts as a penalty for every unit of time that the completion of the product (C_j) has gone past the due time (d_j). Therefore, the TWT as a sum of all weighted tardiness costs is calculated as follows:

$$TWT = \sum(C_{t,j} * (C_j - d_j)) \qquad \text{(Equation 8.1)}$$

2) Computational effort for rescheduling. When any change or disruption occurs on the shop floor, the self-organisation models should respond with the best possible solution in the lowest possible time. The time steps in simulations are analogous to seconds in real manufacturing scenarios. As described in [86], the two fundamentally different behaviour models are expected to perform very differently in such circumstances. For the hybrid model, it is measured in seconds taken to compile a schedule. For the decentralised model, the measured value is the time taken (in seconds) for the longest time step in the

simulation. This is because instead of planning forward, the decentralised model makes decisions through sealed-bid auctions at each time step. Therefore, a disruption can only have an effect on a single negotiation round, where each agent makes its typical decisions just like in any other round.

8.1.2. Experiments and Results

In this section, the hybrid self-organisation model is compared to the decentralised model. As no other models have been proposed for this specific purpose, the cooperative behaviour model from Chapter 4 is used as a benchmark to compare against.

In experiment 1, the decentralised model, with different spending strategies and moving penalty factors, is compared to the hybrid model and the cooperative model in the assessment of minimising TWT in different scenarios. In experiment 2, the algorithmic efficiencies of the models are compared. This is done by measuring the computational overheads at various numbers of products and product working contents. As discussed in Chapter 7, the varied factors are the spending strategy of the PAs and the moving penalty factors of RAs. The hybrid model needs no adjustments because it is shown in Chapter 6 that it is optimised already. The cooperative model follows unchangeable rules and thus is not subject to changes either.

The experiments are carried out on a computer with an i3-5020U processor (dual-core, 2.2GHz), 64bit Windows 10 OS and in the JADE agent development environment (version 4.5.0). JADE was configured to use 2048MB of heap memory.

To show how the models apply to multiple machines, a small number of four RAs (equivalent to 8 mobile robots) are deployed to process PAs. There are 8 workstations (2 per RA). Each RA works at a rate of 1 unit of working content per second, i.e. if all four RAs processed the same product for 1,000 seconds, then there would be 4,000 units of work done.

The system configuration, which is typical for large structure assembly, introduces a limitation on the launch time, as this depends on the availability of the crane CS. The CS uses the first-come-first-served logic to load products on each of the 8 workstations. When there are no available workstations to load products on, the CS unloads a randomly chosen workstation with a completed product. In order to prevent the CS from being a supply bottleneck in the system, the time to load and unload products is set to 4,000 seconds each. This way, the 4 RAs need 10,000 seconds on average to process any product and the CS needed 8,000 seconds to have a product loaded and unloaded. As a result of that, the CS can always supply products faster than they can get processed.

The workstations were laid out as shown in Figure 8-1: The workstation layout for the experiments. The distances between workstations are scaled to represent those that would typically be seen in large structure assembly. For each workstation $WS_{a,\ b}$ (X; Y), "X" represents its coordinate in metres in the direction of the rows and "Y" represents its coordinate in metres in the direction of the columns. One of the currently largest manufactured products that need a great amount of drilling and filling is the Airbus A380 aircraft. Judging by its shape and size, the wing panels are roughly 40 metres long. The RAs consider the distances between workstations in a straight line and therefore to accommodate for turns and traffic, it seemed fair to set the gap between adjacent workstations to be 60 metres.

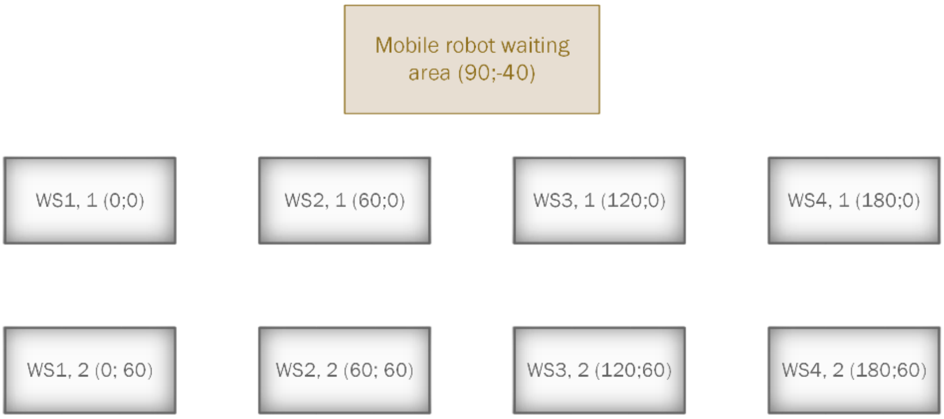


Figure 8-1: The workstation layout for the experiments

In order not cause a supply bottleneck at the start and to establish a scenario with some work-in-progress, it is assumed that half of all workstations $WS_{i\ldots1,2}$ are loaded. The crane system *CS* then loads new products to available workstations $WS_{i\ldots1,2}$ and unloads completed products. The production stops when the last product J_j is completed.

8.1.2.1. Experiment 1

This experiment consists of four sub-experiments, each one representing a specific scenario. In each one there are 20 products with a working content of 40,000 seconds each. They are launched in the predetermined order and have predetermined properties (working content, due time and tardiness cost). Such a setup reflects the order in which products are usually launched in the aircraft manufacturing industry, as there are long-standing orders that can be estimated to a good extent in advance. Problems with such an approach can occur when there has been a disruption of any kind. That can result in the reduction of available resources, new due times and the importance of certain products.

Scenarios with firstly abundant and then sufficient resources with various other complicating conditions are created for both self-organisation models. The general specifications for this experiment are shown in Table 8-1 and more detailed settings are presented in Appendix D. Sub-experiment 1a is designed to test whether both models with all considered variations could finish the products without tardiness. The reason for doing so is to confirm the finding in [20] that with sufficient spare capacity, any sensibly designed agent behaviour model can achieve TWT = 0 in the given problem. The flow of products is steady in the sense that every next product is launched with a later due time than the previous ones. Sub-experiment 1b has the same flow of products, however with no spare capacity. It is designed so that mathematically any deviation or error causes tardiness of a product. The reason for this sub-experiment is firstly to confirm that the hybrid model achieves optimal results, secondly, to compare how the sub-optimal variations of the decentralised model compare to one-another, and finally, how both models compare against the benchmarked cooperative behaviour model. In sub-experiment 1c, all settings of 1b other than the tardiness costs remain the same. Every other PA's tardiness cost is halved.

The variance is restricted in this work because the model is designed to be executed in a synchronous turn base manner, where all decisions are taken at the exact same time and there is no noise in the experimental setup. The aim of the work is to confirm the differences between the models and not determining the optimal setups of the decentralised model. Therefore, it is sufficient to run each experiment only once in a noiseless environment.

Sub-experiment	Spare Capacity	Additional Challenging Factors
1a	5%	None
1b	0%	None
1c	0%	Every other PA's tardiness cost is halved
1d	0%	Irregular intervals between product launch and due times. Impossible to have TWT = 0.

Table 8-1: The matrix of the first experiment's descriptions

The interest in this sub-experiment is to test the hybrid model's optimisation. Additionally, it is in assessing how the variations of the decentralised model can handle the differences in tardiness costs. Also, it is important to illustrate how these models compare to the benchmarked cooperative behaviour model. In the final sub-experiment, all settings from sub-experiment 1b other than the due times remain the same. In this case, some products with very rushed due times are designed to be launched so that they disrupt the natural order of product completions. It is also designed in such a way that mathematically it is impossible to complete all products on time.

The optimality of a schedule is calculated as follows: Knowing that each product's working content W_j is 40,000 and there re 4 mobile resources used in each experiment, on average, the throughput should be 10,000 seconds per product. Thus, it should be possible to complete the first product in 10,000 seconds, the second product in 20,000 seconds, tenth in 100,000 seconds and twentieth in 200,000 seconds. Examining Appendix D and applying this logic, it must be possible to complete the first three sub-experiments with TWT = 0. However, in sub-experiment 1d, the latest due time for the first 13 products is 120,000 seconds. This means that at least one product must be tardy by at least 10,000 seconds. Multiplying this by the tardiness cost $C_{t,j}$ = 1,000, the lowest possible TWT for this sub-experiment becomes 10,000,000.

The results for this experiment are shown in Figure 8-2. In every sub-experiment, the hybrid model achieves optimal results, as expected and described above. The decentralised model's results in sub-experiment 1a confirm the expectations that despite sub-optimality, the decentralised model achieves TWT = 0 at all its behaviour variations when there is 5% spare capacity. The cooperative behaviour model is clearly not well-suited for these scenarios, as it consistently shows some of the highest TWT. It is also the only model that results in some TWT in sub-experiment 1a.

The different factors of the decentralised model are denominated by *DXXYY*, where *XX* stands for the initial bid gap G_b and *YY* for the moving penalty P_m. I.e. *D0810* has an initial bid gap of 0.8 and a moving penalty of 1.0. The hybrid model always works under the same configurations and is denominated as "*Hyb*". The cooperative behaviour model is denominated as "*CBM*".

From sub-experiment 1b onwards, the decentralised model gains some TWT at all variations. The results show that the results consistently worsen when the moving penalty factor and initial bid gap are increased. In order to investigate the combinations of moving penalty factors and spending strategies on the TWT in the given sub-experiments, the results were plotted in 3D bar charts. On these charts, the spending strategy is becoming more aggressive when moving rightwards (decreasing variable), moving penalty factor increases when moving backwards and the TWT increases upwards (increasing variable).

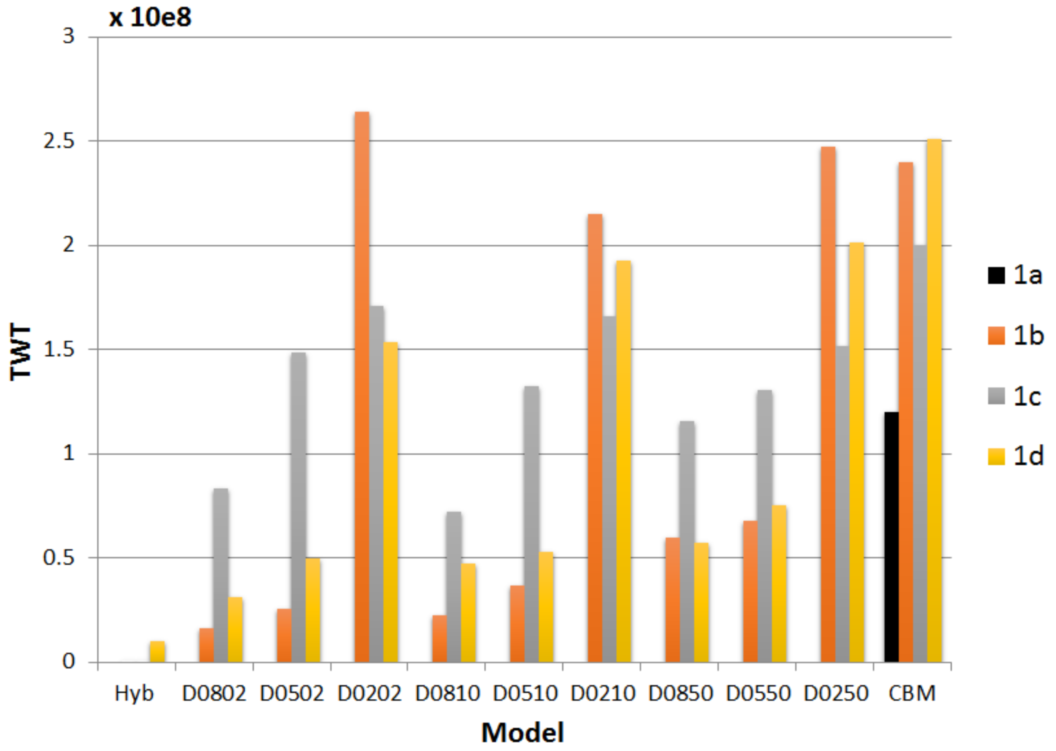


Figure 8-2: The results for experiment 1

The results of the decentralised model at different settings in sub-experiment 1b are compared in Figure 8-3. The increasing moving penalty factor steadily increases the TWT at the two more conservative spending strategies (0.8 and 0.5). However, at the aggressive spending strategy (0.2), the highest TWT is at the lowest moving penalty factor. Clearly, the aggressive spending strategy performs by far the worst. Also, it seems that a low moving penalty factor causes many unnecessary resource relocations at that strategy. More conservative spending strategies lower the resultant TWT. The results show that when there is no spare capacity, then aggressively spending credits at early stages is clearly not in the interest of PAs.

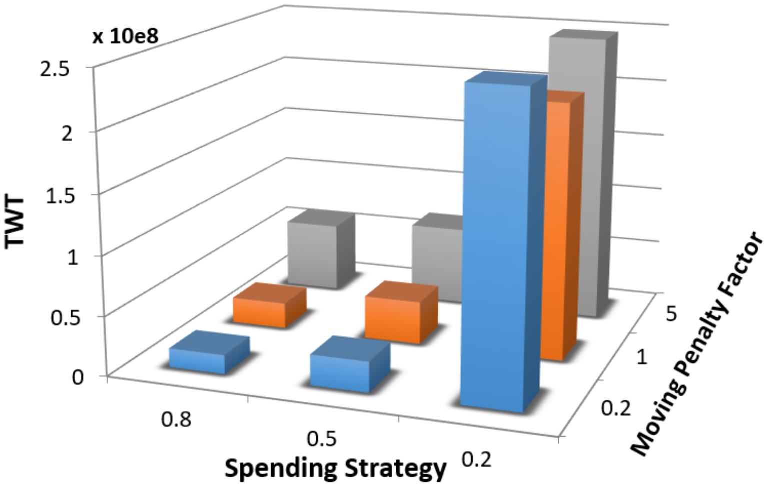


Figure 8-3: The decentralised model's results in sub-experiment 1b

The same graph plotted for sub-experiment 1c is shown in Figure 8-4. Similar trends are produced; however there is a much smaller spread in the results. The most aggressive spending strategy (0.2) only adds a small proportion of TWT on top of the balanced (0.5) one. The effects of the moving penalty factor are less clear in this sub-experiment and are generally smaller than the effects of changing the spending strategy.

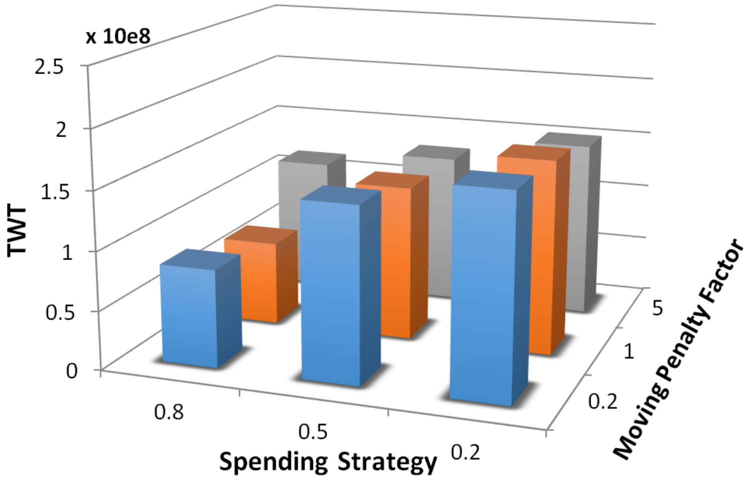


Figure 8-4: The decentralised model's results in sub-experiment 1c

The bar chart for the final sub-experiment is shown in Figure 8-5. The results are very similar to those achieved in sub-experiment 1b: a small increase in

TWT with increasing moving penalty factors; and a sudden increase of TWT for the most aggressive spending strategy.

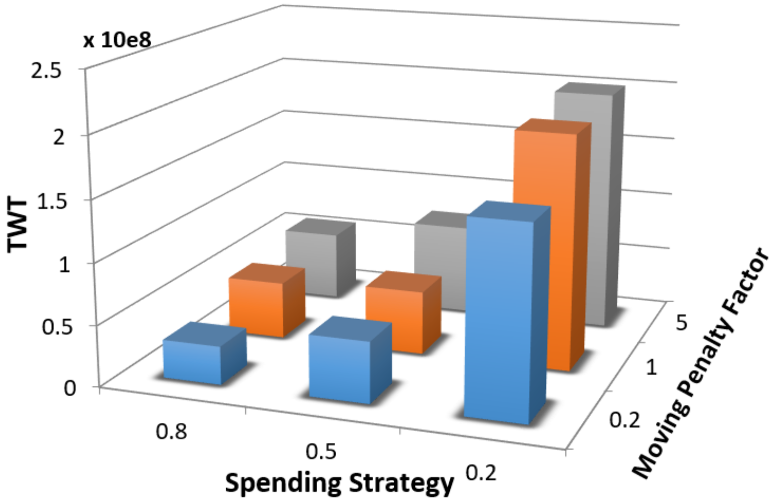


Figure 8-5: The decentralised model’s results in sub-experiment 1d

Throughout experiment 1, the spending strategy of PAs clearly had a much greater effect on TWT than the moving penalty factor and had a clear pattern of increase. The results at more conservative spending strategies (0.8) in each sub-experiment are almost always the best out of the 3 challenging sub-experiments. The TWT is consistently highest at the most aggressive one (0.2). Varying the moving penalty factor, however, is not that predictable. Increasing it usually increases the TWT, but far not as consistently as making the spending strategy more aggressive.

Out of the challenging sub-experiments, 1c has the smallest spread in results. This is because the challenge of large differences in tardiness costs between PAs is very difficult to handle for this decentralised model. The PAs with lower tardiness costs and consequently lower credits in the bank get out-powered by the wealthier PAs with higher tardiness costs. Therefore, there is very little that the model’s parameters could improve.

Conversely, sub-experiments 1b and 1d allow for better results from models with conservative spending strategies. In these sub-experiments, each PA has the same tardiness cost. Clearly, aggressively spending in the early stages is counterproductive for PAs, because that leaves them with low credits at the later stages and significantly higher TWT. Considering sub-experiment 1c, the results indicate that the purchasing power given to PAs has a great effect on the resultant TWT. Therefore, more efficient setups of the decentralised model should revise the credit allocating structure.

Experiment 2 links to experiment 1 by measuring the rescheduling computational effort for the models. The time taken to reprocess the schedule is then included in the schedule as a penalty. Thus, the responsiveness of the models is included in assessing the performance with relation to minimising TWT.

8.1.2.2. Experiment 2

In this experiment, the rescheduling computational effort of the hybrid and decentralised models is measured. By design, the decentralised model is not affected by increasing the frequency of disruptions that may arise in the manufacturing process. Thus, its' response time is considered as the baseline for the comparison against that of the hybrid model's. The cooperative behaviour model is not considered in this experiment, because in terms of responsiveness it behaves in an analogous way to the decentralised model. The purpose of this experiment is to highlight the penalty of the hybrid model due to the rescheduling effort in relation to the decentralised one. The hybrid model would be much less effective without planning forward because the BA would not know to notify PAs about predicted tardiness. On the contrary, the decentralised model is a very versatile self-organisation model where agents take fast and straightforward decisions at each round of bidding. Because the agents in this model do exactly the same at each round of bidding, from the computation perspective, the model is unaffected by disruptions. Thus, the hybrid model's performance in relation to the objective function of minimising TWT is dependent on its responsiveness to disruptions.

This assessment of is important because it becomes possible to estimate the total computational effort and its effects in a real manufacturing system. This can be especially useful if the frequency of disruptions can be estimated.

This experiment is designed to assess the responsiveness of both models. It consists of two sub-experiments: In sub-experiment 2a, the effect of varying the working content of products is assessed and in sub-experiment 2b, the effect of varying the number of products is assessed. These are the two factors that affect the size of the schedules and consequently the time it takes to process them. Other factors are of no interest in this experiment because they do not affect the processing time. It must be noted that there was no time pressure for PAs to be completed. Therefore, the hybrid model did not need to trigger the swapping negotiations and then reschedule with swapped resources.

Similarly to the first experiment, here there were also four mobile resources deployed. Certainly, in a realistic environment with frequent disruptions, the computational overheads would increase each time there is a disruption of any type. In this experiment, the scenarios are limited to a single hypothetical disruption.

Due to the limitations of the specified 2048MB of heap memory, the experiment was bound at a maximum of 90 products and working contents of 40,000 seconds each. Thus, sub-experiment 2a is carried out with 90 products and sub-experiment 2b is carried out with a working content of 40,000 seconds per job. The results are shown in Figure 8-6. For the hybrid model, a change in the working content is linearly proportional to the required computational effort when rescheduling. Whereas, increasing the number of products exponentially increases the required computational effort for the hybrid model.

To establish a comparison, the decentralised model is run through this sub-experiment as well. The longest round of bidding took 1.5 seconds to process. As such, this value provides a baseline for this model's rescheduling computational effort, because it performs the same actions at each round of bidding and is completely unaffected by disruptions on the shop floor.

These results reveal the key characteristics of the behaviour models. For the hybrid model, increasing the planning horizon increases the computational overheads linearly. Increasing the number of products increases the computational overheads exponentially. In both cases, the decentralised model responds in the same manner regardless of altering the abovementioned variables. Thus, the feasibility of the hybrid model is strictly dependent on those variables and on the frequency of disruptions.

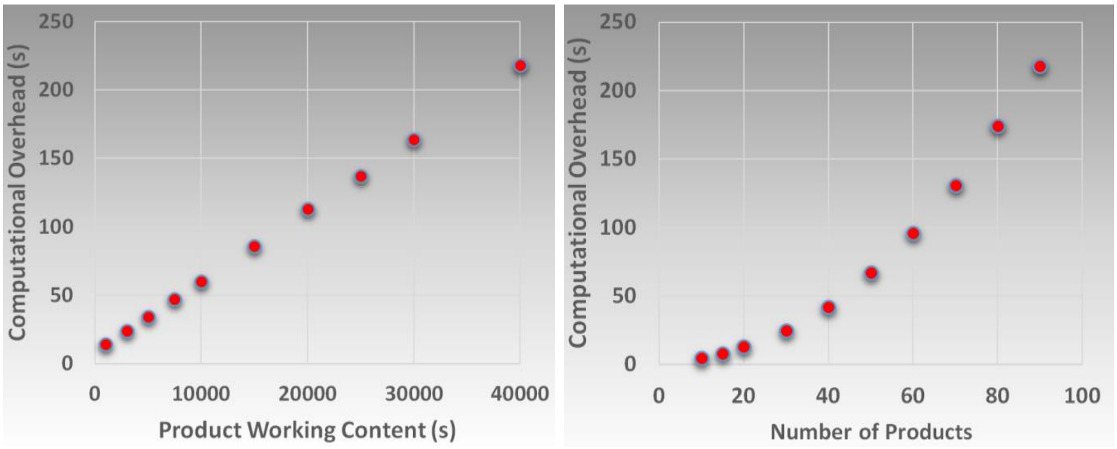


Figure 8-6: The results for sub-experiments 2a (left) and 2b (right)

At the equivalent settings of the first experiment (*n* = 20, $WC_{1...j}$ = 40,000), the hybrid model required approximately 13 seconds to process the schedule. Hypothetically, if the production process was halted for that duration, then the results for the hybrid model in the first experiment would look as represented by "Hyb*" in Figure 8-7. It represents a very extreme case of a hypothetical scenario, where the disruption occurrs shortly after the initial schedule was compiled. This way, the rescheduling proportion of the schedule is highest and the highest number of products is affected by it. It is shown that with 5% spare capacity, it still does not result in any TWT. With 0% spare capacity; a negligible amount of TWT is generated in relation to the decentralised model. Thus, the advantage of the decentralised model's responsiveness is not of substantial value in scenarios with a single disruption.

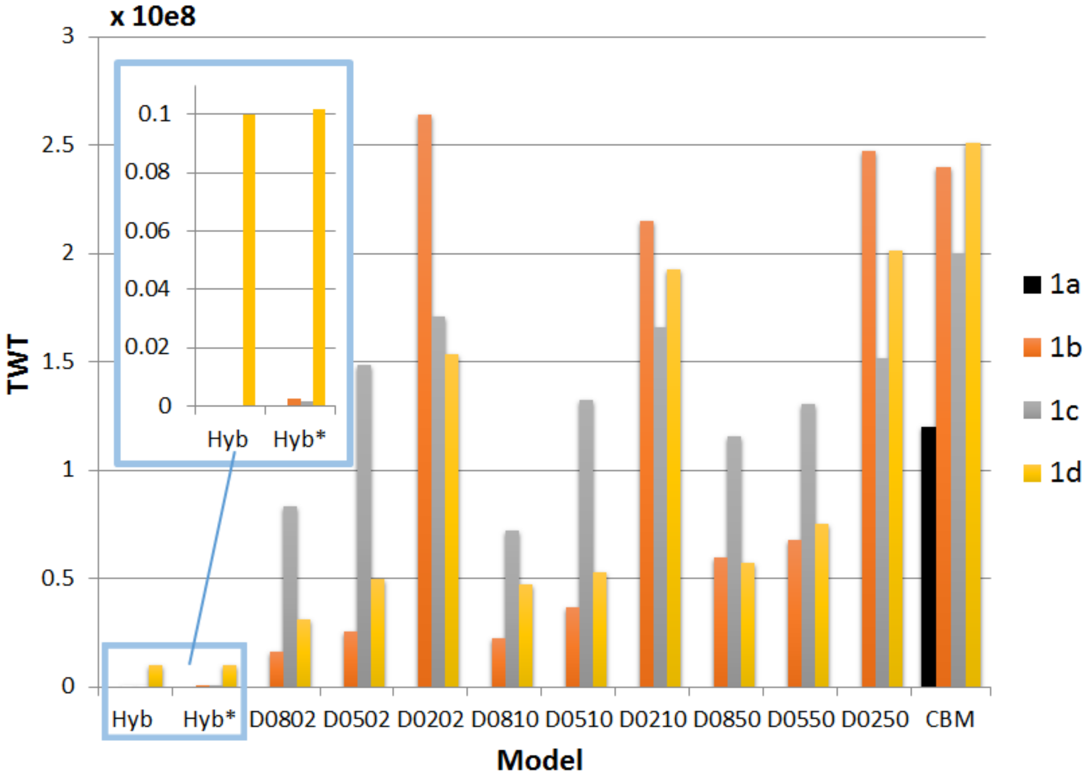


Figure 8-7: The results for the first experiments with added tardiness due to computational overheads

8.1.3. Discussion

The hybrid model achieves consistently the best and optimal results in the given simulations. That was expected because the model is optimised. The optimisation is achieved due to having a single entity (BA) in the system that receives global knowledge of the whole environment. However, for the same reason, it must process a large amount of information and notify PAs of tardiness when necessary. As shown in experiment 2, this can be very computationally demanding to do. Furthermore, if the initially built schedule is not optimal and PAs signal that they have agreed to swap resources, the shown times increase further. It's processing times reach approximately 233 seconds when processing 90 products with working contents of 40,000 units each. Certainly, the processing time for compiling the specific schedule may be dramatically decreased by using a more powerful computer or cloud computing services instead of the specified computer. However, the considered scenarios considered only a single stage of the assembly process. With added complexity in the scheduling problem, the computational effort for the hybrid model would increase further along the trend lines.

An argument in favour of the hybrid model is that the system may not necessarily need to stop for rearrangements after a disruption has occurred. In some cases it may be better to proceed with the old (sub-optimal) schedule for the duration of time when the new schedule. This would result in higher utilisation and lower TWT than completely halting the system. Such an approach could be suitable for environments with a low frequency of disruptions. However, if disruptions are frequent, it is possible that new disruptions occur during the time when the hybrid model is still responding to the previous one. As a result, there would be little to no sense in using the hybrid model at all. Thus, in addition to the challenges of extending the planning horizon and increasing the number of entities, the hybrid model is also limited by the frequency of disruptions.

Unavoidable problems for the hybrid model would be the excessive upscaling and code change requirements. With excessive upscaling, the required computational effort would eventually become too large for efficient operation. The other challenge with the hybrid model is its software code complexity. It is accepted that self-organisation models with centralised architectures have a greater volume and complexity of code than those with decentralised architectures. Therefore, expanding the hybrid model’s code further is demanding in two ways: the software engineering effort and the hardware that processes it.

The cooperative behaviour model from Chapter 4 performs very poorly and it clearly shows that the more sophisticated models are much better suited for the purpose. The decentralised model achievs sub-optimal results in situations where there is no spare capacity designed into the product flow. Where there is 5% spare capacity, there the model handles the experiment at all considered settings without accumulating any TWT. Considering that in the North American automotive industry, the machines typically operate at efficiency levels of 60-70% [59], the setup in sub-experiment 1a is not too optimistic. Such efficiency levels are intended to be increased in case if there is a need, i.e. due to scheduling difficulties or disruptions. Further, in the first experiment, the models are given tight due times (0% spare capacity) in various scenarios. As opposed to the hybrid model, the decentralised model does not achieve optimal results in these. A relatively regular pattern can be identified in the results. The model consistently performs better when the PAs were initially set to offer smaller credits at their bidding rounds. Thus, the PAs save credits for the later stages at the expense of having lower odds of attracting RAs at the start. This works well for two reasons: firstly, PAs have high bargaining power when close to their due times; and secondly, newly launched PAs cannot compete with the finishing ones yet. On the opposite end of the results, PAs that offer high amounts of credits from early stages onwards are much less competitive nearer to the due time when new PAs are already being launched. This finding is in agreement with the work that the hybrid model [125] was based on. Furthermore, outbidding competing PAs was further obstructed by the moving penalty factor that the RAs have to consider before moving from one PA to another. Very large proportions of TWT for the decentralised model with aggressive spending strategies resulted from when PAs had already missed their due times and could not outbid other PAs for the remaining few resources.

Certainly, the presented decentralised model is only one out of a vast range of possible models that could be developed for the given purpose. Moreover, it is tested at only 9 different setups, meaning that it is unlikely that it performed to its best ability in the given scenarios. The results, however, confirm a very important point from Chapter 4: In steady situations with sufficient spare manufacturing capacity, the self-organisation models needn’t be complicated at all. Without spare capacity, the decentralised model gains some TWT at every setting and scenario. This indicates that the given model must necessarily have some spare capacity in the system to compensate for the imperfections. In the future, it would be interesting to determine the exact amount at which the model started gaining TWT.

The power of the decentralised system would be amplified in a larger system with different stages of assembly and consequently with the need for different kinds of skills for mobile robots. The illustration in Figure 8-8 represents a possible shop floor layout for such a production plant. In a more complex variant of this problem, a mix of products with different skill and tooling requirements would be launched in designated areas on the shop floor. Firstly, this layout adds optimisation complexity, because now the individual skills of RAs and their requirements for PAs will have to be considered. Secondly, the larger job shop layout would have more agents on it. In such a layout, it would be possible to vary the individual spare capacities and eliminate bottlenecks in the areas by transferring resources between them.

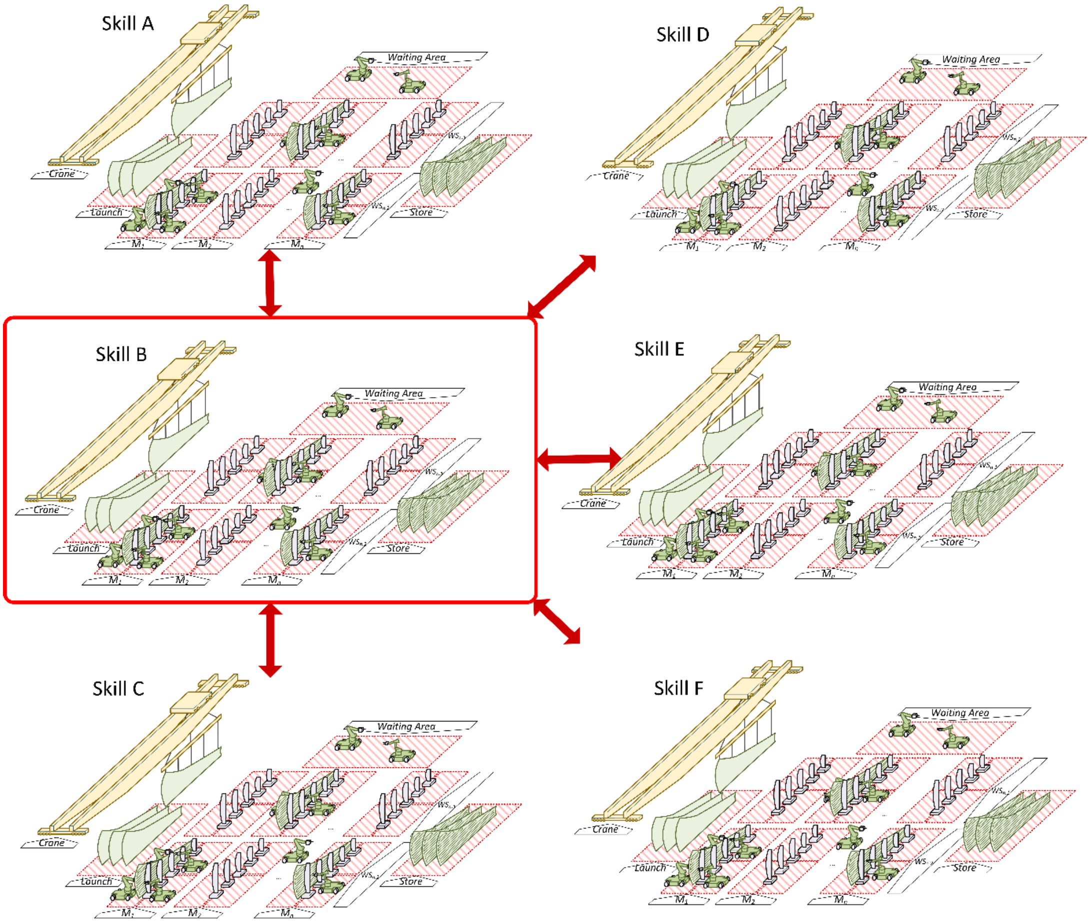


Figure 8-8: A sample expanded job shop layout

Knowing that the computational effort is exponentially proportional to the increase in the number of agents, the rescheduling time for the hybrid model, in this case, would further increase by a large amount. From the experiments in this paper, it is difficult to estimate how significant it would be for any specific system, as there can be many sizes and variations to it. Nevertheless, despite the sub-optimality of the decision-making, the decentralised model would continue with its normal behaviour and high responsiveness.

The only things that could make it respond slower would be the additional code and messaging required for negotiations. The communication load could be reduced by introducing more localised messaging, so that very distant (and therefore highly penalised) RAs would not even receive messages. It is fair to assume that in a model with more code and heavier communication demands, the time to reschedule from any possible disruption should never go up by a whole order of magnitude. Therefore, the decentralised system has a natural advantage over the hybrid model in terms of computational effort, dealing with complexity and the time required to respond to changes.

Because the processing times in large structure assembly are very long in comparison to the computational efforts shown in this chapter, the models can also take a pre-negotiating approach (similar to [102]). The advantage would be the fact that at each instance, the agents have either already negotiated or are currently negotiating on the next step(s), in effect eliminating the wait between predictable events. However, in such a setup, both models would not have an immediate response for an unpredictable event and would still need to negotiate/schedule as was done in this thesis. Furthermore, the high computational effort of the hybrid model could potentially make this infeasible due to the necessary time and hardware costs.

8.2. Chapter Summary

The first hypothesis did not require validation as such, because it was a comparison of manufacturing systems in representative scenarios. The second hypothesis is validated in this chapter by comparing the hybrid and decentralised models in a range of scenarios.

The experiments confirm the natural advantages and disadvantages of the model architectures. The hybrid model achieves optimal scheduling results at the expense of higher computational effort. The decentralised model, on the other hand, does not achieve the optimal results in challenging scenarios.

However, it achieves 0 TWT with 5% spare capacity under all behaviour settings and it shows that it constantly experiences low computational loads regardless of the environment. The cooperative behaviour model showed consistently some of the worst results due to not taking into account the actual due times of products.

The conclusion is that wherever the hybrid model's computational effort is not excessive nor is it expected to require many modifications in the future, it should be preferred to the decentralised system. This is due to more efficient utilisation of existing resources. However, if the system is very large or is expected to grow, then it could be worth using the decentralised model. Whilst the decentralised model may require a small proportion of additional capital investment in the beginning, it is highly likely to overcome many problems later in the manufacturing system's life cycle.

The decentralised model is a very versatile and adaptable model that does not get impeded by computational effort. Its weakness is the sub-optimality, which requires additional capital investment in order to have some spare capacity in the system to make up for it.

It can be concluded that if a mobile manufacturing system is not large enough to cause computational issues nor will it need many changes in its lifetime, the hybrid model is the better option. However, if the system is large enough to cause significant delays due to disruptions or is expected to grow into that, then it is definitely worth using the decentralised model. Whilst the decentralised model may require a small proportion of additional capital investment in the beginning to add spare capacity, it is highly likely to overcome many problems later in the manufacturing system's life cycle.

Chapter 9 - Conclusions

9.1. Thesis Overview

The first aim of this thesis was to investigate under which circumstances a system of mobile robots would have operational advantages in comparison to fixed automation systems in LSA. The second aim was to develop and compare two fundamentally different self-organisation models for autonomous resource allocations.

The literature review for the thesis was carried out in Chapter 2. It revealed that previously mobile robots had been successfully deployed for tasks where accuracy and structural stiffness have not been of decisive importance. Much research was done in the context of those types of applications and they provided several principles for work in this thesis. Firstly, the work published in the field of improving the physical capabilities of mobile robots was discussed. Whilst this topic is not of key importance for this thesis, it was important to highlight that the enabling technology was already reaching readiness. The review identified the knowledge gap that there was no work done to investigate under which circumstances mobile systems should be preferred to fixed systems. A systematic comparison between the two approaches was required to identify where mobility is beneficial. Secondly, the literature review discussed different means of self-organising mobile robots. No work was found to be directly focusing on self-organising mobile robots in LSA. In terms of scheduling algorithms, work from related fields was not found to consider product-centric scalability with the objective to minimise TWT. Thus, the second knowledge gap was that it was unknown how the planning of mobile robot allocations should be carried out in manufacturing environments of LSA. It was concluded that using agent-based behaviour models for controlling mobile robots was most common due to their analogous characteristics (limited knowledge of the environment; distribution/granularity; ability to communicate).

To deliver the first key knowledge contribution, the systems were compared in like-for-like simulated scenarios in Chapter 4. As a confirmation of the first hypothesis, it was found that mobile systems can utilise their manufacturing resources and control product delivery times better in dynamic scenarios. The higher utilisation is due to fixed automation systems having a limited working envelope and therefore being constrained to fewer workstations. This is not true for mobile systems because they can work at any reachable workstation on the shop floor. Furthermore, knowing that products are very large in size, it is physically possible to fit different numbers of mobile robots around the same product. This enables the manufacturing system to vary individual work rates on products and thus have higher control over product delivery times.

There are three key benefits to using mobile systems. Firstly, it is the scaling work rates on products, as discussed above. This is very important when individual products are parts of larger assemblies and any single delay can affect the release of the final products. Secondly, higher utilisation means that the manufacturer can choose to reduce the full manufacturing capacity of their system by the difference in utilisation. Thirdly, this enables scheduling maintenance and other operations on individual mobile robots more freely due to redundancy. Conversely, mobile robots cannot do any value-adding work while moving between products. It can be argued that the moving time proportion can have a meaningful economic impact; however, it is negligible in relation to the utilisation gained due to mobility in the first place. Thus, it was determined that in the context of LSA, overcoming technical limitations for deploying mobile robots can reap great rewards.

The second objective was to develop a hybrid agent-based self-organisation model for mobile robots in LSA. It is a novelty because no literature was found to ever combine priority aging with enhancing distributed negotiation models. This was the second knowledge contribution and it was delivered in Chapter 5 and Chapter 6. The model required two chapters because it consists of two key components: a priority aging policy (PAP) and an agent-based structure that uses it. The PAP is used to compile a schedule without any decision-making. It starts by ranking each individual product agent by its priority rating. The product agents then get resource agents allocated to them based on that ranking order. Three PAPs were selected for investigation. The simulation results show that the exponential PAP is consistently the most effective because it causes the smallest numbers of tardy products and consistently lowest TWT values.

However, the exponential PAP is not optimal. As is shown in Chapter 5, sometimes the schedules could be improved. The agent behaviour model in Chapter 6 is developed to address that. There, a blackboard agent is introduced to collect all the information about product and resource agents at the start of a simulation. It then simulates the full schedule from the beginning until the last product completion by using the PAP. If it identifies that any products are predicted to be late, it notifies the respective product agents so that they would seek to swap resources. Thus, a method of automatically allocating resources to products and a method of refining its imperfections were created in this model. This is shown to consistently achieve near-optimal results, as stated in the second hypothesis.

The third objective, to develop a fully decentralised model for mobile robots in LSA, is addressed in Chapter 7. This model is fully distributed and the product agents compete for resource agents. The novelty is in including product-centric scalability for products with due times. Each product agent is given a set amount of credits at launch time. To link the purchasing power of each product agent to the economics of the shop floor, this amount is directly proportional to the tardiness cost per unit time for each product. The credits are then offered to resource agents in sealed bid auctions. The bidding strategy is to start by offering few credits to resource agents at the early rounds of each PA’s processing time. If sufficient RAs are received, then the PA keeps paying them the same amount. However, if insufficient resources are received, then the bid is gradually increased at each consecutive round of bidding. Effectively, each agent proceeds independently and does not use any central entities for coordination. Thus, as the third key knowledge contribution, a fully distributed model was developed for mobile robots in LSA.

As the fourth and final core knowledge contribution, the self-organisation behaviour models are compared in Chapter 8. The hypothesis based on the literature review is that the distributed model would be more responsive, but at the cost of less efficient schedules. This is due to each agent possessing limited information as opposed to having complete knowledge, like in the hybrid model. The hybrid system was expected to achieve better schedules with respect to minimising TWT; however, the computational overheads are shown to grow exponentially with additional entities on the shop floor. In Chapter 8, that hypothesis is confirmed, reaffirming that the models work as intended and in line with existing theory. It is also reiterated how there is no need for sophisticated behaviour models when there is a spare capacity as small as 5% designed into the system.

Whilst it is difficult to quantify the exact circumstances due to many possible variations of manufacturing environments, the investigation confirms the expected trends. Shop floors with few entities on it should use the hybrid model due to more efficient scheduling. With an increase of entities, the hybrid model's computational overheads increase exponentially and can eventually become prohibitive. In this case, the decentralised model should be used. Another key parameter is the working content of products. It has been shown that the more working content there is the more information the blackboard agent must process per amount of products. The simulation results show that an increase in working content leads to a linear increase in computational overheads. This means that the effect is not as strong as for the number of agents, but is still important to take into account. It is possible to alter the results by changing factors such as CPU, negotiation time limit and scheduling horizon. However, the underlying trends are expected to always remain the same.

9.2. Limitations

The work in this doctoral thesis had two major limitations: the absence of a case study and limiting the work to simulations. Despite elaborate efforts, it was not possible to gain access to process planning information of a suitable manufacturing environment. Therefore, the best judgment was used when defining the manufacturing problem. The lack of physical testing exhibits its common disadvantages of idealisation in the numerical simulations. The properties of mobile robots and assembly products were estimated based on available evidence but did not necessarily replicate existing manufacturing environments. Thus, the results can be considered generic and not accurately attainable in industrial scenarios. Therefore, the extents of the differences between both manufacturing systems and self-organisation behaviour models may substantially vary in more specific scenarios.

Also, the work considered only the single stage of assembly where products require a large volume of unspecified working content. Mobile robots were assigned into groups controlled by resource agents. Such an approach is suited to tasks such as drilling and filling. However, it is too simplified for many other tasks. For example, it can be very difficult, if not impossible to increase the work rate of welding or painting tasks by adding machines to the specific products. Furthermore, in an assembly line of several stages, the scheduling problem can become very different. Thus, a limitation was the lack of consideration for a variety of tasks and multiple stages of assembly.

Furthermore, throughout the thesis, it was considered that the mobile robots are homogeneous, can reach every point of interest on every product and do not have routing issues, maintenance breaks or messaging failures. In reality, it may not be possible to attain such abilities. Accordingly, the work could be greatly refined by including a number of such impediments.

Finally, there is a potentially very complicated challenge with respect to the dimensional tolerances when processing products common to aircraft assembly. When, for example, a wing panel is subjected to external forces in one location, it is very likely to deform and cause dimensional changes in another section of the product. This is an engineering challenge that must be overcome in order to achieve the product-centric scalability, as widely applied in this thesis. Therefore, it is insufficient to only have mobile robots with the necessary structural stiffness and accuracy to be applied in this way.

9.3. Thesis Contributions

The following contributions were added to knowledge as a result of this doctoral thesis:

- A comparison of the manufacturing resilience and utilisation between mobile and fixed automation systems. In this like-for-like comparison, the differences due to mobility were identified. It was shown that mobility can increase machine utilisation and improve the ability to control product delivery times. These advantages are particularly useful when the product flow is unsteady or unpredictable.
- A hybrid self-organisation model for mobile robots in large structure assembly. This agent-based behaviour model includes a central blackboard agent that processes information and notifies product agents if they are predicted to be tardy. Due to having complete knowledge of the environment, the model is optimised. However, it can only handle small instances of the scheduling problem, because it was shown that adding agents to the simulation causes the computational overheads to increase exponentially.
- A decentralised self-organisation model for mobile robots in large structure assembly. This is a fully distributed model that relies on socio-economic rules to govern the behaviour of agents. Each agent has limited knowledge of the environment and, therefore, cannot guarantee optimal results. The advantage of this model is the fact that regardless of the number of agents, the response time remains very low. This is because each agent processes the same set of rules regardless of the environment.
- A comparison of hybrid and decentralised self-organisation models for mobile robots in large structure assembly. This contribution validates the second hypothesis. It was shown that the hybrid model should be preferred situations where there is a relatively low number of agents (under 100) and disruptions occur rarely. Contrarily, the decentralised model maintains high responsiveness regardless of the environment. This makes it suitable to use on large shop floors with frequent disruptions.

9.4. Future Work

As mentioned in the sections above, the work in this doctoral thesis only serves as a foundation for the idea of deploying mobile robots to LSA. The work clearly shows that there are a number of advantages in pursuing this idea. Whilst limited in scope, it is hoped that the work will generate more interest in this concept.

Many directions for future work have been identified. It is important to understand what challenges are brought about in the physical implementation of mobile robots. This can include challenges like routing (including issues with utility supply cables), communication overheads (in an environment with large, metallic objects), physical coordination of tasks and ways of improving structural stiffness and/or accuracy to guarantee that the tasks are consistently carried out to a high standard.

Another set of directions can be identified in the digital environment. Future contributions should build on this work by adding additional factors to the scheduling problem, proposing alternative self-organisation models and developing the switching mechanism for hybrid and decentralised models.

Using the switching mechanism, the mobile system would switch to the decentralised model whilst the blackboard agent processes the information when a disruption occurs. Once the hybrid model has built the schedule, the system would switch back. Analogous methods to this have been successfully executed in [102] and [105].

It is proposed that the decentralised model is tested with moving times included in the simulation. It would be interesting to investigate the impact of moving time in case-specific scenarios. Depending on the results, it could then be reasonable to set high moving penalty factors to discourage excessive moving.

Furthermore, it would be interesting to analyse how the models perform in noisy environments. Noise can be added in the form of asynchronous messaging or randomised experimental variables, for example. Such an analysis would require a much higher sample size for experiments. Knowing the results would enable predicting which behaviour factors would achieve the best results for the decentralised model. This can be considered a logical continuation of the work in this doctoral thesis. However, it was not possible to include it due to the limited duration of the PhD project.

Also, it is inconclusive how much spare capacity would be the minimum for the decentralised model to achieve TWT = 0 in the given scenarios. Because 5% is clearly more than enough and 0% is insufficient, it is important to evaluate the correct estimate. Certainly, the results in this thesis are limited to a set of perceived scenarios. More precise estimates could be reached through the investigation of case studies and the simulations of their representative models.

Case studies from representative manufacturing environments should also be used for testing the models. Eventually, the scenarios should expand to a multi-stage manufacturing shop floor that would use thousands of agents on it with different skills and requirements. The interest would be in seeing at which complexity the hybrid model would become unusable and whether the decentralised model could continue performing to the same standard. Thus, work must be done in the direction of narrowing the gap between academia and industry.

References

[1] Airbus, “Orders and Deliveries,” 2018. [Online]. Available: https://www.airbus.com/aircraft/market/orders-deliveries.html. [Accessed: 27-Nov-2018].

[2] Boeing, “Boeing_ Commercial,” 2018. [Online]. Available: http://www.boeing.com/commercial/#/orders-deliveries. [Accessed: 27-Nov-2018].

[3] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger, and O. Madsen, “Robot skills for manufacturing: From concept to industrial deployment,” *Robot. Comput. Integr. Manuf.*, pp. 1–10, 2015.

[4] B. Sarh, “Wing Structural Assembly Methodology,” no. 724, 2016.

[5] AssemblyMag, “Assembly Automation Takes Off in Aerospace Industry,” 2015. [Online]. Available: https://www.assemblymag.com/articles/92790-assembly-automation-takes-off-in-aerospace-industry. [Accessed: 27-Nov-2018].

[6] D. Remley, J. Rediger, P. Haworth, and R. Holden, “Slug Rivet Machine Installs 16 Rivets Per Minute Drill-Rivet-Shave,” 2009.

[7] N. Lum and Q. Luo, “E7000 ARJ -Manual Flexible Tooling for Horizontal Riveting System,” *SAE Int. J. Aerosp.*, vol. 7, no. 2, pp. 2014-01-2250, 2014.

[8] S. Smith, I. Moore, and T. Jackson, “HAWDE Five Axis Wing Surface Drilling Machine,” *Assembly*, no. 724, 2004.

[9] S. Hogan, J. Hartmann, B. Thayer, J. Brown, I. Moore, J. Rowe, and M. Burrows, “Automated Wing Drilling System for the A380-GRAWDE,” *System*, 2003.

[10] T. G. Jefferson, S. Ratchev, and R. Crossley, “Axiomatic Design of a Reconfigurable Assembly System for Primary Wing Structures,” *SAE Int. J. Aerosp.*, vol. 7, no. 2, pp. 2014-01-2249, 2014.

[11] Y. C. Chou, Y. L. Lin, and K. F. Chun, “A construction of knowledge rules for reactive planning of job-mix assignment to homogeneous serial batch

machines," *Int. J. Prod. Econ.*, vol. 151, pp. 56–66, 2014.

[12] R. Müller, M. Esser, T. Mbarek, N. Biyiklioglu, and S. Quinders, "Development of Modular Low Cost Automation Systems for Aircraft Assembly," 2011.

[13] A. Millar and H. Kihlman, "Reconfigurable Flexible Tooling for Aerospace Wing Assembly," *Aerosp. Technol. Conf. Expo.*, vol. 4970, p. 10, 2009.

[14] G. J. Burley and J. Corbett, "Flyaway Tooling for Higher Quality , More Cost-Effective, Aerostructure," no. 724, 1998.

[15] T. G. Jefferson, P. Benardos, and S. Ratchev, "Reconfigurable Assembly System Design Methodology: A Wing Assembly Case Study," *SAE Int. J. Mater. Manf.*, vol. 9, no. 1, pp. 31–48, 2015.

[16] S. Zhang and X. Xia, "Optimal control of operation efficiency of belt conveyor systems," *Appl. Energy*, vol. 87, no. 6, pp. 1929–1937, 2010.

[17] B. O'Rourke and R. Rounkles, "Automation for Unprecedented Production Rates," 2012.

[18] N. Boysen, D. Briskorn, and F. Meisel, "A generalized classification scheme for crane scheduling with interference," *Eur. J. Oper. Res.*, vol. 0, pp. 1–15, 2016.

[19] Z. Yuguang, X. Kai, and S. Dongyan, "Assembly unit partitioning for hull structure in shipbuilding," *Comput. Des.*, vol. 45, no. 12, pp. 1630–1638, 2013.

[20] S. Ljasenko, N. Lohse, and L. Justham, "A Comparison of the Manufacturing Resilience between Fixed Automation Systems and Mobile Robots in Large Structure Assembly," in *49th CIRP Conference on Manufacturing Systems*, 2016, vol. 0.

[21] T. Gray, D. Orf, and G. Adams, "Mobile automated robotic drilling, inspection, and fastening." pp. 1–7, 2013.

[22] S. Doltsinis, P. Ferreira, and N. Lohse, "A Symbiotic Human&#x2013;Machine Learning Approach for Production Ramp-up," *IEEE Trans. Human-Machine Syst.*, no. October, 2017.

[23] C. R. Duguay, S. Landry, and F. Pasin, “From mass production to flexible / agile production,” *Int. J. Oper. Prod. Manag.*, vol. 17, no. 12, pp. 1183–1195, 1997.

[24] S. J. Hu, J. Ko, L. Weyand, H. A. Elmaraghy, T. K. Lien, Y. Koren, H. Bley, G. Chryssolouris, N. Nasr, and M. Shpitalni, “Assembly system design and operations for product variety,” *CIRP Ann. - Manuf. Technol.*, vol. 60, no. 2, pp. 715–733, 2011.

[25] Hadeli, P. Valckenaers, M. Kollingbaum, and H. Van Brussel, “Multi-agent coordination and control using stigmergy,” *Comput. Ind.*, vol. 53, no. 1, pp. 75–96, 2004.

[26] T. Arai, Y. Aiyama, Y. Maeda, M. Sugi, and J. Ota, “Agile Assembly System by ‘Plug and Produce,’” *CIRP Ann. - Manuf. Technol.*, vol. 49, no. 1, pp. 1–4, Jan. 2000.

[27] W. J. Zhang and C. A. van Luttervelt, “Toward a resilient manufacturing system,” *Cirp Ann. Technol.*, vol. 60, no. 1, pp. 469–472, 2011.

[28] B. D. Youn, C. Hu, and P. Wang, “Resilience-Driven System Design of Complex Engineered Systems,” *J. Mech. Des.*, vol. 133, no. 10, p. 101011, 2011.

[29] Z. Wang, M. Liang, and P. G. Maropoulos, “High accuracy mobile robot positioning using external Large Volume Metrology instruments,” vol. m, pp. 484–492, 2011.

[30] N. Sariff and N. Buniyamin, “An Overview of Autonomous Mobile Robot Path Planning Algorithms,” no. June, pp. 183–188, 2006.

[31] M. Gen and L. Lin, “Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey,” *J. Intell. Manuf.*, pp. 1–18, 2013.

[32] L. Wan and J. Yuan, “Single-machine scheduling to minimize the total earliness and tardiness is strongly NP-hard,” *Oper. Res. Lett.*, vol. 41, no. 4, pp. 363–365, 2013.

[33] M. Saidi-Mehrabad, S. Dehnavi-Arani, F. Evazabadian, and V. Mahmoodian, “An Ant Colony Algorithm (ACA) for solving the new

integrated model of job shop scheduling and conflict-free routing of AGVs," *Comput. Ind. Eng.*, vol. 86, pp. 2–13, 2015.

[34] N. Kousi, G. Michalos, S. Makris, and G. Chryssolouris, "Short – term planning for part supply in assembly lines using mobile robots," *Procedia CIRP*, vol. 44, no. Figure 1, pp. 371–376, 2016.

[35] S. M. LaValle, "Planning Algorithms," 2006.

[36] F. Navarro-garcia, A. Serapio-palacios, P. Ugalde-silva, and G. Tapia-pastrana, "An Introduction to Swarm Robotics," *ISRN Robot.*, vol. 2013, 2013.

[37] J.-H. Liang and C.-H. Lee, "Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm," *Adv. Eng. Softw.*, vol. 79, pp. 47–56, Jan. 2015.

[38] T. M. Cheng, A. V. Savkin, and F. Javed, "Decentralized control of a group of mobile robots for deployment in sweep coverage," *Rob. Auton. Syst.*, vol. 59, no. 7–8, pp. 497–507, Jul. 2011.

[39] E. Arcaute, K. Christensen, A. Sendova-Franks, T. Dahl, A. Espinosa, and H. J. Jensen, "Division of labour in ant colonies in terms of attractive fields," *Ecol. Complex.*, vol. 6, no. 4, pp. 396–402, Dec. 2009.

[40] V. Botti and Adriana Giret, "Holonic Manufacturing Systems," *ANEMONA. A multi-agent Methodol. Holonic Manuf.*, 2008.

[41] P. Leitão, J. Barbosa, and D. Trentesaux, "Bio-inspired multi-agent systems for reconfigurable manufacturing systems," *Eng. Appl. Artif. Intell.*, vol. 25, pp. 934–944, 2012.

[42] A. Buschhaus, S. Krusemark, E. Karlidag, and J. Franke, "Universal fine interpolation algorithms for accuracy improvements of industrial robots," *Int. Congr. Ultra Mod. Telecommun. Control Syst. Work.*, vol. 2016–Decem, pp. 356–362, 2016.

[43] A. Pashkevich, A. Klimchik, D. Chablat, and P. Wenger, "Stiffness Analysis Of Multi-Chain Parallel Robotic Systems," vol. 3, no. 3, pp. 75–82, 2009.

[44] G. Michalos, N. Kousi, S. Makris, and G. Chryssolouris, “Performance assessment of production systems with mobile robots,” *Procedia CIRP*, vol. 41, pp. 195–200, 2016.

[45] G. Michalos, K. Kaltsoukalas, P. Aivaliotis, P. Sipsas, A. Sardelis, and G. Chryssolouris, “Design and simulation of assembly systems with mobile robots,” *CIRP Ann. - Manuf. Technol.*, vol. 63, no. 1, pp. 181–184, 2014.

[46] G. Michalos, S. Makris, P. Aivaliotis, S. Matthaiakis, A. Sardelis, and G. Chryssolouris, “Autonomous production systems using open architectures and mobile robotic structures,” *Procedia CIRP*, vol. 28, pp. 119–124, 2015.

[47] H. Kikuchi, Y. Maeda, M. Sugi, and T. Arai, “Automatic calibration of assembly robots for instantaneous installation ‘plug &amp; produce,’” *Adv. Robot.*, vol. 17, no. 1, pp. 41–52, 2003.

[48] R. Simmons, S. Singh, and D. Hershberger, “First results in the coordination of heterogeneous robots for large-scale assembly,” *Exp. Robot. …*, vol. 27, no. 2001.1, pp. 323–332, 2001.

[49] A. Pott, C. Meyer, and A. Verl, “Large-scale assembly of solar power plants with parallel cable robots,” *41st Int. Symp. Robot. 6th Ger. Conf. Robot.*, pp. 999–1004, 2010.

[50] M. Dogar, R. A. Knepper, A. Spielberg, C. Choi, H. I. Christensen, and D. Rus, “Towards coordinated precision assembly with robot teams,” *Springer Tracts Adv. Robot.*, vol. 109, pp. 655–669, 2016.

[51] S. Giordani, M. Lujak, and F. Martinelli, “A distributed multi-agent production planning and scheduling framework for mobile robots,” *Comput. Ind. Eng.*, vol. 64, no. 1, pp. 19–30, Jan. 2013.

[52] M. Wooldridge, *An Introduction to MultiAgent Systems*. 2002.

[53] B. Płaczek, “A self-organizing system for urban traffic control based on predictive interval microscopic model,” *Eng. Appl. Artif. Intell.*, vol. 34, pp. 75–84, Sep. 2014.

[54] S. Werner, H. T. K. Vu, and J. C. Rink, “Self-organization in development, regeneration and organoids,” *Curr. Opin. Cell Biol.*, vol. 44, pp. 102–109,

2017.

[55] Z. I. Foraida, T. Kamaldinov, D. A. Nelson, M. Larsen, and J. Castracane, “Elastin-PLGA hybrid electrospun nanofiber scaffolds for salivary epithelial cell self-organization and polarization,” *Acta Biomater.*, vol. 62, pp. 116–127, 2017.

[56] M. Saberi, K. Aghabayk, and A. Sobhani, “Spatial fluctuations of pedestrian velocities in bidirectional streams: Exploring the effects of self-organization,” *Phys. A Stat. Mech. its Appl.*, vol. 434, pp. 120–128, 2015.

[57] R. Frei and G. D. M. Serugendo, “Self-organizing assembly systems,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 41, no. 6, pp. 885–897, 2011.

[58] P. Leitão, N. Rodrigues, J. Barbosa, C. Turrin, and A. Pagani, “Intelligent products: The grace experience,” *Control Eng. Pract.*, vol. 42, pp. 95–105, 2015.

[59] X. Gu, X. Jin, J. Ni, and Y. Koren, “Manufacturing System Design for Resilience,” *Procedia CIRP*, vol. 36, pp. 135–140, 2015.

[60] C. Anderson and J. Bartholdi, “Centralized versus decentralized control in manufacturing: lessons from social insects,” *Complex. complex …*, no. September, pp. 92–105, 2000.

[61] W. Shen and D. H. Norrie, “Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey,” *Knowl. Inf. Syst.*, vol. 1, pp. 129–156, 2013.

[62] S. L. Khelifati and F. Benbouzid-Sitayeb, “A sequantial distributed approach for the joint scheduling of jobs and maintenance operations in the flowshop sequencing problem,” *2011 11th Int. Conf. Intell. Syst. Des. Appl.*, pp. 420–425, 2011.

[63] M. Mes, M. van der Heijden, and A. van Harten, “Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems,” *Eur. J. Oper. Res.*, vol. 181, no. 1, pp. 59–75, 2007.

[64] D. Trzyna, A. Kuyumcu, and H. Lödding, “Throughput Time Characteristics of Rush Orders and their Impact on Standard Orders,”

vol. 3, pp. 311–316, 2012.

[65] EUROP, “Robotic Visions to 2020 and beyond – The Strategic Research Agenda for robotics in Europe.” 2009.

[66] I. Lee and K. Lee, “The Internet of Things (IoT): Applications, investments, and challenges for enterprises,” *Bus. Horiz.*, vol. 58, no. 4, pp. 431–440, 2015.

[67] H. Kagermann, W. Wahlster, and J. Helbig, “Recommendations for implementing the strategic initiative INDUSTRIE 4.01,” no. April. p. 82, 2013.

[68] M. Wooldridge and N. R. Jennings, “Applications of Intelligent Agents,” *Agent Technol.*, 1998.

[69] P. Leitão, “Agent-based distributed manufacturing control: A state-of-the-art survey,” *Eng. Appl. Artif. Intell.*, vol. 22, no. 7, pp. 979–991, 2009.

[70] Q. Guo and M. Zhang, “A novel approach for multi-agent-based Intelligent Manufacturing System,” *Inf. Sci. (Ny).*, vol. 179, no. 18, pp. 3079–3090, 2009.

[71] N. R. Jennings, K. Sycara, and M. Wooldridge, “A Roadmap of Agent Research and Development,” *Auton. Agent. Multi. Agent. Syst.*, vol. 38, pp. 7–38, 1998.

[72] S. Franklin and A. Graesser, “Is It an agent, or just a program?: A taxonomy for autonomous agents,” in *Third International Workshop on Agent Theories, Architectures, and Languages*, 1997, pp. 21–35.

[73] D. McFarlane, V. Giannikas, A. C. Y. Wong, and M. Harrison, “Product intelligence in industrial control: Theory and practice,” *Annu. Rev. Control*, vol. 37, no. 1, pp. 69–88, 2013.

[74] C. Y. Wong, D. McFarlane, A. Ahmad Zaharudin, and V. Agarwal, “The intelligent product driven supply chain,” *IEEE Int. Conf. Syst. Man Cybern.*, vol. vol.4, p. 6, 2002.

[75] G. G. Meyer, K. Främling, and J. Holmström, “Intelligent Products: A survey,” *Comput. Ind.*, vol. 60, no. 3, pp. 137–148, 2009.

[76] A. D. Baker, “A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: Dispatching, scheduling, and pull,” *J. Manuf. Syst.*, vol. 17, no. 4, pp. 297–320, 1998.

[77] J.-H. Lee and C.-O. Kim, “Multi-agent systems applications in manufacturing systems and supply chain management: a review paper,” *Int. J. Prod. Res.*, vol. 46, no. March 2015, pp. 233–265, 2008.

[78] J. Dong, S. Chen, and J.-J. Jeng, “Event-Based Blackboard Architecture for Multi-Agent Systems.”

[79] D. Rudenko and A. Borisov, “an Overview of Blackboard Architecture Application for Real,” in *SCIENTIFIC PROCEEDINGS OF RIGA TECHNICAL UNIVERSITY Information*, 2007.

[80] D. D. Corkill, “Blackboard and Multi-Agent Systems & the Future,” *Collab. Softw.*, no. January 2003, 2016.

[81] M. O. F. Sarker, T. S. Dahl, E. Arcaute, and K. Christensen, “Local interactions over global broadcasts for improved task allocation in self-organized multi-robot systems,” *Rob. Auton. Syst.*, vol. 62, no. 10, pp. 1453–1462, Oct. 2014.

[82] A. Giret and V. Botti, “Holons and agents,” *J. Intell. Manuf.*, vol. 15, no. 5, pp. 645–659, 2004.

[83] P. Leitao, “Holonic rationale and bio-inspiration on design of complex emergent and evolvable systems,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5740 LNCS, pp. 243–266, 2009.

[84] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters, “Reference architecture for holonic manufacturing systems: PROSA,” *Comput. Ind.*, vol. 37, no. 3, pp. 255–274, 1998.

[85] J.-L. C. J.-L. Chirn and D. C. McFarlane, “A holonic component-based approach to reconfigurable manufacturingcontrol architecture,” *Proc. 11th Int. Work. Database Expert Syst. Appl.*, pp. 1–5, 2000.

[86] P. Leitão and F. Restivo, “ADACOR: A holonic architecture for agile and adaptive manufacturing control,” *Comput. Ind.*, vol. 57, pp. 121–130,

2006.

[87] J. Barbosa, P. Leitão, E. Adam, and D. Trentesaux, “Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution,” *Comput. Ind.*, vol. 66, pp. 99–111, 2015.

[88] P. Leitão and J. Barbosa, “Dynamic Switching Mechanism to Support Self-organization in ADACOR Holonic Control System,” *IFAC-PapersOnLine*, vol. 49, no. 31, pp. 161–166, 2016.

[89] F. P. Maturana and D. H. Norrie, “Multi-agent Mediator architecture for distributed manufacturing,” *J. Intell. Manuf.*, vol. 7, no. 4, pp. 257–270, 1996.

[90] A. M. Farid and L. Ribeiro, “An Axiomatic Design of a Multi-Agent Reconfigurable Mechatronic System Architecture,” *Submitt. to IEEE Trans. Autom. Sci. Eng.*, vol. 1, no. 1, pp. 1–10, 2014.

[91] Z. Alibhai and B. A. Sc, “What is Contract Net Interaction Protocol ?,” *Contract*, 2003.

[92] E. Oliveira, F. Mouta, and a. P. Rocha, “Negotiation and conflict resolution within a community of cooperative agents,” *Proc. ISAD 93 Int. Symp. Auton. Decentralized Syst.*, pp. 421–427, 1993.

[93] T. N. Wong and F. Fang, “A multi-agent protocol for multilateral negotiations in supply chain management,” *Int. J. Prod. Res.*, vol. 48, no. 1, pp. 271–299, 2010.

[94] S. S. Fatima, M. Wooldridge, and N. R. Jennings, “An agenda-based framework for multi-issue negotiation,” *Artif. Intell.*, vol. 152, no. 1, pp. 1–45, 2004.

[95] N. Gatti, F. Di Giunta, and S. Marino, “Alternating-offers bargaining with one-sided uncertain deadlines: an efficient algorithm,” *Artif. Intell.*, vol. 172, no. 8–9, pp. 1119–1157, 2008.

[96] K. Ozonat and S. Singhal, “Design of Negotiation Agents Based on Behavior Models,” *Web Inf. Syst. Eng. - WISE 2010*, vol. 6488, no. i, pp. 308–321, 2010.

[97] J. Zhang, F. Ren, and M. Zhang, “Bayesian-based preference prediction in bilateral multi-issue negotiation between intelligent agents,” *Knowledge-Based Syst.*, vol. 84, pp. 108–120, 2015.

[98] B. Renne, “Multi-agent Justification Logic: communication and evidence elimination,” *Synthese*, vol. 185, no. S1, pp. 43–82, 2012.

[99] K. Mansour and K. Ryszard, “An Approach for Coordinating the Bidding Strategy in Multi-Issue Multi-Object Negotiation with Multiple Providers,” *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2261–2272, 2014.

[100] S. D. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings, “Negotiating using rewards,” *Artif. Intell.*, vol. 171, no. 10–15, pp. 805–837, 2007.

[101] B. An, V. Lesser, and K. M. Sim, “Strategic agents for multi-resource negotiation,” *Auton. Agent. Multi. Agent. Syst.*, vol. 23, no. 1, pp. 114–153, 2011.

[102] P. Ferreira, S. Doltsinis, A. Anagnostopoulos, F. Pascoa, and N. Lohse, “A performance evaluation of industrial agents,” *IECON 2013 - 39th Annu. Conf. IEEE Ind. Electron. Soc.*, pp. 7404–7409, 2013.

[103] A. Agnetis, J.-C. Billaut, S. Gawiejnowicz, D. Pacciarelli, and A. Soukhal, *Multiagent Scheduling - Models and Algorithms*. 2012.

[104] A. T. Tolmidis and L. Petrou, “Multi-objective optimization for dynamic task allocation in a multi-robot system,” *Eng. Appl. Artif. Intell.*, vol. 26, no. 5–6, pp. 1458–1468, 2013.

[105] K. Macarthur, M. Vinyals, A. Farinelli, S. Ramchurn, and N. Jennings, “Decentralised Parallel Machine Scheduling for Multi-Agent Task Allocation,” pp. 1–16, 2011.

[106] Y. Liu, H. Dong, N. Lohse, S. Petrovic, and N. Gindy, “An investigation into minimising total energy consumption and total weighted tardiness in job shops,” *J. Clean. Prod.*, vol. 65, pp. 87–96, 2014.

[107] M. Mes, M. van der Heijden, and J. van Hillegersberg, “Design choices for agent-based control of AGVs in the dough making process,” *Decis. Support Syst.*, vol. 44, no. 4, pp. 983–999, 2008.

[108] V. Kaplanoğlu, “Multi-agent based approach for single machine scheduling with sequence-dependent setup times and machine maintenance,” *Appl. Soft Comput.*, vol. 23, pp. 165–179, 2014.

[109] D. Elvikis, H. W. Hamacher, and V. T&apos;kindt, “Scheduling two agents on uniform parallel machines with makespan and cost functions,” *J. Sched.*, vol. 14, no. 5, pp. 471–481, 2011.

[110] D. Elvikis and V. T’kindt, “Two-agent scheduling on uniform parallel machines with min-max criteria,” *Ann. Oper. Res.*, vol. 213, no. 1, pp. 79–94, 2014.

[111] C. Leet, J. Hahn, Y. Seot, C. Y. Parkq, M. Lee, and C. S. Kim, “Enhanced Analysis of Cache-related Preemption Delay in Fixed-priority Preemptive Scheduling *,” *IEEE Trans. Comput.*, pp. 187–198, 1997.

[112] F. Al-Saqqar, J. Bentahar, K. Sultan, and M. El Menshawy, “On the interaction between knowledge and social commitments in multi-agent systems,” *Appl. Intell.*, vol. 41, no. 1, pp. 235–259, 2014.

[113] K. Schild and S. Bussmann, “Self-Organization in Manufacturing Operations,” *Commun. ACM*, vol. 50, no. 12, pp. 74–79, 2007.

[114] F. L. Almeida, B. M. Terra, P. A. Dias, and G. M. Goncalves, “Adoption Issues of Multi-agent Systems in Manufacturing Industry,” *2010 Fifth Int. Multi-conference Comput. Glob. Inf. Technol.*, pp. 238–244, 2010.

[115] K. H. Hall, R. J. Staron, and P. Vrba, “Experience with Holonic and Agent-Based Control Systems and Their Adoption by Industry,” *Holonic Multi-Agent Syst. Manuf. Proc. Second Int. Conf. Ind. Appl. Holonic Multi-Agent Syst.*, vol. 3593, no. i, pp. 1–10, 2005.

[116] V. Marík and D. McFarlane, “Industrial Adoption of Agent-Based Technologies,” *IEEE Comput. Soc.*, no. February, pp. 27–35, 2005.

[117] J. A. Peixoto, C. E. Pereira, and J. B. de Oliveira, “Functionalities of multi-agent systems in Programmable Logic Controllers,” *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 60–64, 2016.

[118] Electroimpact Inc, “Electroimpact E7000 Squeeze Riveter,” 2017. [Online]. Available:

https://www.electroimpact.com/Products/Fastening/E7000.aspx. [Accessed: 19-Jan-2017].

[119] G. G. Meyer, J. C. Hans Wortmann, and N. B. Szirbik, “Production monitoring and control with intelligent products,” *Int. J. Prod. Res.*, vol. 49, no. 5, pp. 1303–1317, 2011.

[120] U. Wilensky, “NetLogo Home Page,” 2016. [Online]. Available: https://ccl.northwestern.edu/netlogo/. [Accessed: 13-Jul-2016].

[121] J. Barbosa and P. Leitao, “Simulation of multi-agent manufacturing systems using Agent-Based Modelling platforms,” *2011 9th IEEE Int. Conf. Ind. Informatics*, pp. 477–482, 2011.

[122] M. Thürer, C. Silva, M. Stevenson, and M. Thu, “Workload control release mechanisms : from practice back to theory building theory building,” vol. 7543, no. January, 2016.

[123] M. Wu and C. Chen, “A multiple criteria decision-making model for justifying the acceptance of rush orders,” no. April 2014, pp. 37–41, 2010.

[124] M. Nosrati, *Task Scheduling Algorithms Introduction*, no. June. 2012.

[125] S. Ljasenko, N. Lohse, L. Justham, and I. Pereira, “A Self-Organisation Model for Mobile Robots in Large Structure Assembly using Multi-Agent Systems,” in *SOHOMA’16 Workshop*, 2016, pp. 1–8.

Appendix A

		d_j	C_{tj}	W_j	t_l
ARC = 0.7	PA A	15,000	750	10,000	0
	PA B	17,000	1,000	10,000	0
	PA C	40,000	750	10,000	17,000
	PA D	40,000	1,000	10,000	17,000
ARC = 0.8	PA A	16,000	750	10,000	0
	PA B	18,000	1,000	10,000	0
	PA C	40,000	750	10,000	18,000
	PA D	40,000	1,000	10,000	18,000
ARC = 0.9	PA A	17,000	750	10,000	0
	PA B	19,000	1,000	10,000	0
	PA C	40,000	750	10,000	19,000
	PA D	40,000	1,000	10,000	19,000
ARC = 1	PA A	18,000	750	10,000	0
	PA B	20,000	1,000	10,000	0
	PA C	40,000	750	10,000	20,000
	PA D	40,000	1,000	10,000	20,000

Table A- 1: The setup for ARC testing

		d_j	C_{tj}	W_j	t_l
C_{dj} = 0.7	PA A	17,000	750	10,000	0
	PA B	20,000	1,000	10,000	0
	PA C	40,000	750	10,000	20,000
	PA D	40,000	1,000	10,000	20,000
C_{dj} = 0.8	PA A	18,000	750	10,000	0
	PA B	20,000	1,000	10,000	0
	PA C	40,000	750	10,000	20,000
	PA D	40,000	1,000	10,000	20,000
C_{dj} = 0.9	PA A	19,000	750	10,000	0
	PA B	20,000	1,000	10,000	0
	PA C	40,000	750	10,000	20,000
	PA D	40,000	1,000	10,000	20,000
C_{dj} = 1	PA A	20,000	750	10,000	0
	PA B	20,000	1,000	10,000	0
	PA C	40,000	750	10,000	20,000
	PA D	40,000	1,000	10,000	20,000

Table A- 2: The setup for C_{dj} testing

		d_j	C_{tj}	W_j	t_l
d_{tc} = 0	PA A	18,000	1,000	10,000	0
	PA B	20,000	1,000	10,000	0
	PA C	40,000	1,000	10,000	20,000
	PA D	40,000	1,000	10,000	20,000
d_{tc} = 0.25	PA A	18,000	750	10,000	0
	PA B	20,000	1,000	10,000	0
	PA C	40,000	750	10,000	20,000
	PA D	40,000	1,000	10,000	20,000
d_{tc} = 0.5	PA A	18,000	500	10,000	0
	PA B	20,000	1,000	10,000	0
	PA C	40,000	500	10,000	20,000
	PA D	40,000	1,000	10,000	20,000
d_{tc} = 0.75	PA A	18,000	250	10,000	0
	PA B	20,000	1,000	10,000	0
	PA C	40,000	250	10,000	20,000
	PA D	40,000	1,000	10,000	20,000

Table A- 3: The setup d_{tc} testing

Appendix B

	d_i	W_i	C_{ti}	r_i	WT
Job 1	50,000	50,000	1,000	50,000	0
Job 2	50,000	50,000	1,000	50,000	0
Job 3	50,000	50,000	1,000	50,000	0
Job 4	50,000	50,000	1,000	50,000	0
Job 5	50,000	50,000	1,000	50,000	0
				Total:	0

Table B- 4: The input parameters and results for the first experiment

	d_i	W_i	C_{ti}	r_i	WT
Job 1	50,000	50,000	1,000	56,250	6,250,000
Job 2	50,000	50,000	1,100	50,000	0
Job 3	50,000	50,000	1,200	50,000	0
Job 4	50,000	50,000	1,300	50,000	0
Job 5	50,000	50,000	1,400	50,000	0
				Total:	6,250,000

Table B- 5: The input parameters and results for the second experiment

	d_i	W_i	C_{ti}	r_i	WT
Job 1	50,000	50,000	1,000	44,956	0
Job 2	50,000	50,000	1,000	44,980	0
Job 3	50,000	50,000	1,000	45,000	0
Job 4	50,000	50,000	1,000	25,000	SCRAPPED
Job 5	50,000	50,000	1,000	44,973	0
				Total:	0

Table B- 6: The input parameters and results for the third experiment

	d_j	W_j	C_{tj}	r_j	WT
Job 1	50,000	50,000	1,000	60,000	10,000,000
Job 2	50,000	50,000	1,000	52,000	2,200,000
Job 3	50,000	50,000	1,000	50,000	0
Job 4	50,000	50,000	1,000	50,000	0
Job 5	50,000	50,000	1,000	50,000	0
Job 6	45,000	50,000	20,000	45,000	0
				Total:	12,200,000

Table B- 7: The input parameters and refined results for the fourth experiment

	d_j	W_j	C_{tj}	r_j	WT
Job 1	49,000	50,000	100	50,000	100,000
Job 2	50,000	50,000	1,000	50,000	0
Job 3	50,000	50,000	1,000	50,000	0
Job 4	50,000	50,000	1,000	50,000	0
Job 5	50,000	50,000	1,000	50,000	0
				Total:	100,000

Table B- 8: The input parameters and initial results for the fifth experiment

	d_j	W_j	C_{tj}	r_j	WT
Job 1	49,000	50,000	100	25,000	0
Job 2	50,000	50,000	1,000	50,000	0
Job 3	50,000	50,000	1,000	50,000	0
Job 4	50,000	50,000	1,000	50,000	0
Job 5	50,000	50,000	1,000	50,000	0
				Total:	0

Table B- 9: The input parameters and refined results for the fifth experiment

Appendix C

	d_i	W_i	C_{ti}	r_i	WT
PA 1	50,000	50,000	1,000	50,000	0
PA 2	50,000	50,000	1,000	50,000	0
PA 3	50,000	50,000	1,000	50,000	0
PA 4	50,000	50,000	1,000	50,000	0
PA 5	50,000	50,000	1,000	50,000	0
				Total:	0

Table C- 10: The setup and results of the first verification experiment

	d_i	W_i	C_{ti}	r_i	WT
PA 1	50,000	50,000	1,000	50,745	745,000
PA 2	50,000	50,000	1,100	49,998	0
PA 3	50,000	50,000	1,200	49,983	0
PA 4	50,000	50,000	1,300	50,000	0
PA 5	50,000	50,000	1,400	47,311	0
				Total:	745,000

Table C- 11: The setup and results for the second experiment

	d_i	W_i	C_{ti}	r_i	WT
PA 1	50,000	50,000	1,000	44,837	0
PA 2	50,000	50,000	1,000	39,976	0
PA 3	50,000	50,000	1,000	25,000	SCRAPPED
PA 4	50,000	50,000	1,000	45,602	0
PA 5	50,000	50,000	1,000	45,076	0
				Total:	0

Table C- 12: The setup and results for the third experiment

	d_j	W_j	C_{tj}	r_j	WT
PA 1	50,000	50,000	1,000	50,000	0
PA 2	50,000	50,000	1,000	60,000	10,000,000
PA 3	50,000	50,000	1,000	50,000	0
PA 4	50,000	50,000	1,000	50,000	0
PA 5	50,000	50,000	1,000	52,000	2,000,000
PA 6	45,000	50,000	5,000	45,000	0
				Total:	12,200,000

Table C- 13: The setup and results for the fourth experiment

Appendix D

Product Number	d_i	C_{ti}	W_i
1	40,000	1,000	40,000
2	40,000	1,000	40,000
3	40,000	1,000	40,000
4	50,000	1,000	40,000
5	60,000	1,000	40,000
6	70,000	1,000	40,000
7	80,000	1,000	40,000
8	90,000	1,000	40,000
9	100,000	1,000	40,000
10	110,000	1,000	40,000
11	120,000	1,000	40,000
12	130,000	1,000	40,000
13	140,000	1,000	40,000
14	150,000	1,000	40,000
15	160,000	1,000	40,000
16	170,000	1,000	40,000
17	180,000	1,000	40,000
18	190,000	1,000	40,000
19	200,000	1,000	40,000
20	210,000	1,000	40,000

Table D- 14: The experimental settings for sub-experiment 1a

Product number	d_j	C_{tj}	W_j
1	40,000	1,000	40,000
2	40,000	1,000	40,000
3	40,000	1,000	40,000
4	40,000	1,000	40,000
5	50,000	1,000	40,000
6	60,000	1,000	40,000
7	70,000	1,000	40,000
8	80,000	1,000	40,000
9	90,000	1,000	40,000
10	100,000	1,000	40,000
11	110,000	1,000	40,000
12	120,000	1,000	40,000
13	130,000	1,000	40,000
14	140,000	1,000	40,000
15	150,000	1,000	40,000
16	160,000	1,000	40,000
17	170,000	1,000	40,000
18	180,000	1,000	40,000
19	190,000	1,000	40,000
20	200,000	1,000	40,000

Table D- 15: The experimental settings for sub-experiment 1b

Product number	d_i	C_{ti}	W_i
1	40,000	1,000	40,000
2	40,000	500	40,000
3	40,000	1,000	40,000
4	40,000	500	40,000
5	50,000	1,000	40,000
6	60,000	500	40,000
7	70,000	1,000	40,000
8	80,000	500	40,000
9	90,000	1,000	40,000
10	100,000	500	40,000
11	110,000	1,000	40,000
12	120,000	500	40,000
13	130,000	1,000	40,000
14	140,000	500	40,000
15	150,000	1,000	40,000
16	160,000	500	40,000
17	170,000	1,000	40,000
18	180,000	500	40,000
19	190,000	1,000	40,000
20	200,000	500	40,000

Table D- 16: The experimental settings for sub-experiment 1c

Product number	$\underline{d_j}$	$\underline{C_{tj}}$	$\underline{W_j}$
1	40,000	1,000	40,000
2	40,000	1,000	40,000
3	39,000	1,000	40,000
4	40,000	1,000	40,000
5	50,000	1,000	40,000
6	60,000	1,000	40,000
7	90,000	1,000	40,000
8	95,000	1,000	40,000
9	80,000	1,000	40,000
10	100,000	1,000	40,000
11	120,000	1,000	40,000
12	120,000	1,000	40,000
13	110,000	1,000	40,000
14	150,000	1,000	40,000
15	140,000	1,000	40,000
16	160,000	1,000	40,000
17	180,000	1,000	40,000
18	170,000	1,000	40,000
19	190,000	1,000	40,000
20	200,000	1,000	40,000

Table D- 17: The experimental settings for sub-experiment 1d