2023-09-20

Decentralized Scheduling Using The Multi-Agent System Approach For Smart Manufacturing Systems: Investigation And Design

Ebufegha, Akposeiyifa Joseph

UNIVERSITY OF CALGARY

DECENTRALIZED SCHEDULING USING THE MULTI-AGENT SYSTEM APPROACH FOR SMART MANUFACTURING SYSTEMS: INVESTIGATION AND DESIGN

by

Akposeiyifa Joseph Ebufegha

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN MECHANICAL AND MANUFACTURING ENGINEERING

CALGARY, ALBERTA

SEPTEMBER, 2023

Abstract

The advent of industry 4.0 has resulted in increased availability, velocity, and volume of data as well as increased data processing capabilities. There is a need to determine how best to incorporate these advancements to improve the performance of manufacturing systems. The purpose of this research is to present a solution for incorporating industry 4.0 into manufacturing systems. It will focus on how such a system would operate, how to select resources for the system, and how to configure the system.

Our proposed solution is a smart manufacturing system that operates as a self-coordinating system. It utilizes a multi-agent system (MAS) approach, where individual entities within the system have autonomy to make dynamic scheduling decisions in real-time. This solution was shown to outperform alternative scheduling strategies (right shifting and dispatching priority rule) in manufacturing environments subject to uncertainty in our simulation experiments.

The second phase of our research focused on system design. This phase involved developing models for two problems: (1) resource selection, and (2) layout configuration. Both models developed use simulation-based optimization. We first present a model for determining machine resources using a genetic algorithm (GA). This model yielded results comparable to an exhaustive search whilst significantly reducing the number of required experiments to find the solution. To address layout configuration, we developed a model that combines hierarchical clustering and GA. Our numerical experiments demonstrated that the hybrid layouts derived from the model result in shorter and less variable order completion times compared to alternative layout configurations.

Overall, our research showed that MAS-based scheduling can outperform alternative dynamic scheduling approaches in manufacturing environments subject to uncertainty. We also show that this performance can further be improved through optimal resource selection and layout design.

Acknowledgements

I would like to acknowledge my supervisor Dr. Simon Li for his diligence and patience in his guidance over the course of assembling this body of work.

Dedication

First and foremost, I would like to dedicate this thesis to my mother Mrs. Edith Ebufegha. I would also like to dedicate this thesis to the most important people in the world to me; my soon-to-be wife, Hailie Sorge, as well as my siblings Vivian, Kate, Doubra, Tammy, Tonbra, and Ebi. This journey has been a long one, and I appreciate all that everyone did to make the journey a happy one.

Table of Contents

List of Tables

List of Figures and Illustrations

List of Symbols, Abbreviations and Nomenclature

Symbol	Definition
ST	Setup Time
PT	Processing Time
TT	Transfer Time
WT	Wait Time
FT	Flow Time
f_{ij}	Volume of parts flowing between points *i* and *j*
d_{ij}	Shortest distance between points *i* and *j*
dm_{ij}	Shortest distance between machines *i* and *j*
c_{ij}	Cost of transporting parts between points *i* and *j*
RM	Reachability matrix
SC_{ij}	Similarity coefficient
M	List of different machine types for use in the manufacturing system
P	List of different part types the manufacturing system can produce
O	List of different part operations the manufacturing system can perform
MCH	Vector listing the number of units of each machine type being used in the system
sm	Set of machines selected for use in the manufacturing system
OM	Operations that the given manufacturing system can perform
L	Vector containing the location assignment for each machine in the system
G	Matrix representation of available spaces for machine location assignment
D	Distance matrix showing relative distances between location assignments
Y	Matrix representation of machine location assignment
r_i	Row index for location of machine *i* on the facility grid
k_i	Column index for location of machine *i* on the facility grid
TDC	Total design cost
s	Order for parts received by the system for production cycle. This is a vector containing the volume of each distinct part ordered

y_s	The completion time for the production order *s*
y_{max}	Maximum threshold for completion time
σ_s	Standard deviation in the mean completion time for fulfilling production of order *s*
σ_{max}	Maximum threshold for standard deviation associated with completion time
β	Conversion factor that establishes monetary penalty associated with tardiness
α	Slack time desired between completion time and due date
DD	Due date
np	Maximum number of parts
nm	Maximum number of machines
no	Maximum number of operations
fit	Fitness Function

Chapter 1

Introduction

The rapid advancement of information technologies (IT) has led to great advancements in the manufacturing industry and paved the way for Industry 4.0. Industry 4.0, otherwise referred to as the 4^{th} Industrial Revolution, is an umbrella term covering a number of technological advancements including the industrial Internet of Things (IIoT), cyber-physical production systems (CPPS) and cloud manufacturing. The rise of Industry 4.0 brings with it an increase in data availability, data processing and data analysis capabilities, as well as the increased velocity of the data, and it is important to understand how to utilize these tools effectively for manufacturing. This is specifically the case with scheduling jobs where the manufacturing environment is subject to multiple sources of uncertainty and numerous unexpected events occurring randomly. The increased availability of data in real-time, as well as the increased data processing capabilities make it feasible to better account for stochastic disturbances during production by making real-time scheduling adjustments. These advances also raise questions on how to effectively design manufacturing systems to take advantage of the changes brought about by Industry 4.0. This research aims to investigate how the advent of Industry 4.0 can be integrated into smart manufacturing systems (SMS) to improve dynamic scheduling with better efficiency and stability. Specifically, our research can be divided into four distinct stages:

1) Developing a model for dynamic scheduling in smart manufacturing system (Chapter 3)
2) Evaluating the performance of the SMS model when scheduling in dynamic manufacturing environment (Chapter 4)
3) Developing a model for machine selection for smart manufacturing system (Chapter 5)

4) Developing a model for facilities layout design for smart manufacturing system (Chapter 6)

In this chapter, we will present background on the SMS as well as discuss the design of such systems. We will also outline the objective and scope of this research and highlight the main research contributions. The chapter will conclude with the organization of the thesis.

1.1 Background on the Smart Manufacturing System

Wang et al. (2016) define a smart manufacturing system as consisting of four layers; the physical layer (the smart product, machines, etc.), the industrial network layer (the network over which all communication occurs), the cloud layer (which provides data storage and potential processing), and the supervisory and control layer (which oversees the system). These layers combine to allow the system to self-organize and coordinate its resources to meet the product demand. The key benefit of SMS lies in this autonomy. It can allow the system to schedule jobs in real-time in order to produce parts as efficiently and quickly as possible. However, this benefit comes with increased system complexity which, in turn, comes with potential feasibility concerns. As such, it is important to study and confirm the benefits and limitations of using such a smart manufacturing system. It is important to determine how such a system would perform when subject to differing manufacturing environments. However, there is currently very limited literature that demonstrates the actual benefit of implementing smart manufacturing systems. As such, it is a goal of this study to develop a model for a smart manufacturing system, and to use simulation to analyze this system’s performance relative to alternative approaches for addressing mass customization under varying levels of uncertainty.

In this study, the SMS will be considered as a CPPS. A CPPS is a Cyber-Physical System (CPS) that is geared towards production. A CPS is a system of collaborating and interconnected digital and physical entities, where the digital entities provide data storage and data processing services to improve the performance of the physical system (Monostori, 2014). The digital entities in a CPPS are representations of the physical entities within the system. These digital entities act as the intelligence or the "brain" for the physical entities. As such, they allow for the entities within the system to be able to collaborate and coordinate themselves to make system decisions. Thus, the CPPS employs a distributed control structure by giving its entities this autonomy.

With CPPS, the distributed control is implemented in the form of a multi-agent system (MAS). This system consists of multiple intelligent agents that communicate with each other to schedule and execute work within the system. These agents typically communicate using contract net protocol. Whilst there are multiple agents in the system, it is possible for some agents to have more decision-making capabilities than others. The distribution of decision-making capabilities determines the control architecture. Systems with only one decision-making agent are said to be of hierarchical control architectures whilst those where all agents are decision-making are heterarchical. The type of control architecture used has a significant impact on the system's ability to perform optimally and its response to stochastic disturbances. Despite the research into CPPS implementation approaches, there are very few instances of their implementation and very little literature on the quantitative benefits of such systems. The literature does suggest, however, that there are potentially great benefits in employing CPPS for scheduling.

Our research is concerned with the planning and scheduling aspect of CPPS. In that regard, the key feature of CPPS lies in its distributed control structure. The traditional manufacturing system is typically centralized, scheduling decisions and policies are determined *a* priori by the shopfloor manager. However, due to the increased data processing and storage capacity, and the increased interconnectivity between devices (possible due to the incorporation of sensors and transmitting capabilities into devices), it is possible to delegate some of the decision-making to the components of the system. Also, due to the variety of data that can be collected and how quickly this data can be generated (real-time), there is an increase in the complexity of the decisions that can be made by the system as well as the speed of the system's response. This makes the CPPS a viable alternative to a conventional manufacturing system. CPPS allows for the manufacturing system to shift from a centralized control structure to a more distributed control structure which has various benefits. Barbosa et al (2015) findings suggest that this distributed control results in a system that is better suited for rapid responses to stochastic disturbances to the system. They suggest that implementing elements of distributed control into manufacturing systems can result in a system that yields near optimal performance in undisturbed states whilst being able to adapt to changes quickly and thus, produces a more robust system. Robust but efficient manufacturing systems through the use of distributed control is the major appeal of CPPS.

As mentioned earlier, the CPPS is typically represented in literature as a multi-agent system (MAS). This is a system consisting of multiple, intelligent agents. This sort of system is sometimes referred to as a holonic manufacturing system (HMS) in literature (Van Brussel et al., 1998), where instead of the term agent, the term holon is used. An agent can be described as an autonomous, decision-making entity within a system. To guide this decision-making, it is assumed that the agents are aware of all information about their current state and environment. In the context of a manufacturing environment, this would be akin to a machine agent being aware of whether or not the machine it represents is idle, the capabilities of said machine, and the current open jobs in the system (its environment) that the machine can fulfill. Each agent is self-contained with its own characteristics and behaviours. The characteristics and behaviours of the agents in a system are dependent on the intention of the system designer with regard to the system objective and the resolution to which the physical system is represented through the MAS. For example, if each aspect of the physical production system is represented by an agent (each part, transporter, machine, operator, etc. has an agent) and the system objective is to minimize flow time. Then, a part agent may be designed in such a way that it observes the system to find free machines and moves towards the earliest free machine that can service them. This, of course, assumes that it is the first agent to occupy this machine. After all, the MAS consists of many agents with different behaviours and each agent is able to interact with other agents and its environment (Kang & Choi, 2013). These interactions are governed by a predetermined negotiation protocol.

Distributed control of a system requires collaboration and communication between the agents that comprise this system. This requires establishing a protocol to guide this process. The most commonly used negotiation protocol when discussing distributed system task assignment is contract net protocol (CNP) (Kang & Choi, 2013; Jiang et al., 2017). CNP (Smith, 1980) is based on auction mechanism where agents are divided into contractors and clients. Clients issue tenders for tasks and contractors bid on the task using a currency that is designed for the system. It is an auction-based protocol, where task allocation is done based on auctions for each task executed in real-time. This protocol is often used when employing distributed control for the purpose of dynamic task allocation in scheduling problems (Ouelhadj & Petrovic, 2009). As far as we have seen, most literature on distributed control of manufacturing systems either uses CNP or some variation of it.

Once a negotiation protocol has been outlined, the control architecture of the system must be decided. There are three types of control architectures in relation to CPPS: hierarchical (centralized control), heterarchical (decentralized control) and hybrid (combination of hierarchical and heterarchical) control. One of the earlier MAS control architectures was "Yet Another Manufacturing System" (YAMS) that was developed by Paranuk (1987). YAMS was a factory control system based on contract net protocol. Paranuk (1987) employs a hierarchical control architecture in their model of a shop floor. Van Brussel et al (1998) developed a reference architecture for HMS called product resource order staff architecture (PROSA). Whilst this system consisted of three holons (order, product and resource holons), only the order holons had autonomy. Both these systems use a hierarchical control structure, and this sort of control structure has been found to have poor response to stochastic disruptions to the system. A real world manufacturing environment is subject to a number of stochastic disruptions and as such, would present a number of problems when utilizing this type of control structure. As a result, more recent literature appears to focus on heterarchical and hybrid control architectures. The earliest instance of heterarchical control was Duffie and Piper (1987). In their model they use agents as representations of parts and operators with the goal of scheduling the production of the parts. They found that it addressed some of the issues faced when using hierarchical control. With heterarchical control architectures, the agents typically represent entities within the manufacturing system (jobs, machines, parts, etc.) (Ouelhadj & Petrovic. 2009). These systems have each agent making local scheduling decisions with no consideration of the global objective.

This approach, whilst it has good performance, may not yield the optimal schedule with respect to the global objective and could prove problematic computationally when there is a large number of agents (Brennan & Norrie 2001; Shen & Norrie 1999; Bongaerts et al. 2000; Shen et al. 2001; Tharumarajah 2001). This led to the study of hybrid control structures which try to bridge the gap between hierarchical and heterarchical control. Brennan and Norrie (2001), Bongaerts et al. (2000), and Cavalieri et al. (2000) present comparative studies that suggest that hybrid control architectures provide better performance than heterarchical ones. One of the proposed hybrid control architectures for CPPS was by Leitão and Restivo (2006). Leitão and Restivo (2006) developed the ADAptive holonic COntrol aRchitecture (ADACOR), a holonic architecture for agile and adaptive manufacturing control. ADACOR does this by using a supervisor holon to regulate the activities of the operator holons. They found that the balance of having the features of a hierarchical and heterarchical systems allows for near optimal response when system is undisturbed, but a rapid response to disturbances when one occurs. This finding is inline with those of Barbosa et al. (2015).

Despite the research into CPPS control architecture, there are only a few industrial applications of it in literature. Bussman and Schild (2001) present the Production 2000+ project for producing cylinder heads in a DaimlerChrysler factory plant. They suggest that their proposed system was capable of robust manufacturing in the automotive industry in a high product flexibility environment whilst maintaining a nearly optimal production throughput. Also, note that whilst a number of works discuss CPPS implementation and design, most of these works present conceptual frameworks for implementation and do not quantify the benefits of CPPS implementation in tangible a manner such as improvements in factory performance measures (machine utilization, wait times, flowtime, etc). There is very little literature that addresses this issue. Gronau & Theurer (2016) were able to demonstrate using their hybrid simulation (a combination of virtual and physical models) that CPPS can result in significant reductions to wait times. Žapčević and Butala (2013) found that CPPS implementation can decrease process variance and increase system productivity. Kamble et al. (2020) found through their surveys of companies employing CPPS that they resulted in optimized productivity due to optimized use of labor, materials and machines. As such, there is some literature that indicates that CPPS are beneficial, but further research is warranted.

1.2 Designing a Smart Manufacturing System

Designing a smart manufacturing problem can be seen as two distinct problems. The first problem is determining the resources that will be available within the system. Specifically, which type of machines will be available in the system, as well as how many of each machine type should be in the system. This decision will influence the system's flexibility and robustness. A system with multiple duplicate machines will be better able to accommodate stochastic disturbances. However, having too many duplicate machines also means that the system would have low average machine utilization due to underutilized machines. Similarly, having many machines with different functional capabilities increases the number of operations available to the system, and, as such, adds more ways to produce parts. The selection of machines used in the system would have a significant impact on the planning process for the parts. Specifically, the machine routing options and operation sequences for each parts' process plan. This in turn would have a significant impact on the decisions available to the SMS agents which in turn would affect the system performance. As such, the selection of machines is quite important.

The second problem has to do with the facility layout design. The facility layout is an aspect of the physical layer of the SMS which refers to the arrangement of machinery, labor (operators) and materials in a manufacturing system within the spatial confines of the facility. Once the machines that constitute a manufacturing system are known, determining the facility layout design generally comes down to determining the type of layout to use, and then the location assignment for those machines. The selection of the best layout type is dependent on the variety of products, the volume of demand and the type of product (information relating to parts required and operations sequence required to produce these parts). The most common layout types are product lines, cellular and functional. In a mass customization environment, there is high demand and high product variety as well as volatility in demand and product mix. Conventional layout types are not designed for such a manufacturing environment. One way to address this need is to use a hybrid layout. A hybrid layout combines aspects of the other layout types into one layout. Determining the optimal configuration for the physical layout is important to maximizing the performance of the system. The layout of a facility will impact the flow of WIP through the facility. This would impact the transfer times for parts through the system, and, as such, influence the completion time for orders. As the volume of parts ordered increases, the significance of this decision should become more apparent as the effect of any inefficiency compounds. An efficient layout should reduce the overall flow time for parts in the system.

1.3 Research Objectives and Scope

The main purpose of this research is to develop a framework for the design of a smart manufacturing system for use in a dynamic manufacturing environment subject to multiple sources of uncertainty. To that effect, the objectives of this research are as follows:

1. To develop a model for a smart manufacturing system that is capable of scheduling in a dynamic manufacturing environment. This model will be designed to accommodate high product variety and high product demand. (Chapter 3)
2. To investigate the performance of the model we propose in comparison with alternative models using simulation. During this investigation we will examine the performance of these models under manufacturing environments subject to multiple, varying sources of uncertainty. (Chapter 4)
3. To develop a simulation-based optimization model for determining the machine resources to deploy when designing a smart manufacturing system. The objective here is to design a system that minimizes the mean order completion time without excessive variation in performance or exceeding an allocated budget. (Chapter 5)
4. To develop a simulation-based optimization model for determining the optimal machine location assignments. The objective is to design a layout that minimize the mean order completion time by minimizing the overall time parts spent in transit. This must be done within the spatial confines of the facility whilst maintaining adequate space between all machines. (Chapter 6)

1.4 Research Contributions

The main contributions of this thesis align with the objectives mentioned in the previous section. They are as follows:

1. The development and testing (via simulation) of a model for the operation of a smart manufacturing system (SMS). This model treats the SMS as a decentralized problem solver consisting of a set of agents that collaborate to schedule jobs. This requires

defining the elements of the SMS as well as the rules that govern the how these elements operate and interact with each other as parts of the system.

2. The investigation into how dynamic scheduling strategies perform when used in machine environments that are subject to both job-related and resource-related sources of uncertainty simultaneously. Typically, in literature, when dynamic scheduling problems are studied it is usually in environments subject to a single source of uncertainty (Ouelhadj, 2009). This is not representative of a real manufacturing system. Through this research, we aimed to study if the conventions in the literature held true when the model studied is more representative of the real system. From the literature, it is generally accepted that predictive-reactive scheduling should outperform completely reactive scheduling approaches. However, we were able to find conditions under which completely reactive scheduling outperformed predictive-reactive scheduling.
3. The development of a model for machine resource deployment using simulation-based optimization. The limitation with current models for solving this problem is that they are typically limited to small sized problems due to the complexity of these sorts of problems (Chtourou et al., 2005). Also, the models typically fail to capture complex aspects of the system such as the interaction between the machines and the system inputs, or the interdependence and competition between machines. By using simulation-based optimization, the model presented in this work should be able to address both of these limitations.
4. The development of a model for location assignment for smart manufacturing systems. With the increasing demand for customized products, we have high product variety and high demand volume alongside an uncertain demand mix. Conventional layout designs

have been either designed for high product variety or high demand (Tompkins et al., 2002). Also, these layouts have been designed with the assumption that job shop scheduling would be done by the shopfloor manager. There is little literature on designing layouts for systems that self-coordinate. As such, it is important to develop a model for determining the near-optimal layout for such systems. The layout has a direct effect on how work-in-progress (WIP) flow through the system. This directly impacts the completion time for orders. It is important to understand if the traditional layout design options are the best options for use in SMS's or if new layouts would results in better system performance. The results of our experiments present a case for hybrid layouts potentially outperforming traditional layouts.

1.5 Organization of the Thesis

This thesis is organized into seven (7) chapters. Chapter 1 introduces the research problem as well as our research objectives and motivation. In chapter 2, we present the literature review pertinent to designing our smart manufacturing system. The chapter begins by detailing various dynamic scheduling strategies and approaches. After which, we present how these approaches have been employed in problems involving manufacturing environments subject to different sources of uncertainty. We outline the advantages, limitations, and disadvantages of the dynamic scheduling approaches. Finally, the chapter concludes with a review of the literature on machine selection, and facilities layout design.

In chapter 3, we present a detailed description of our proposed model for the smart manufacturing system. The proposed system consists of two domains, a physical and agent domain. We define and describe the characteristics of the elements of each domain, as well as describe how the overall system functions. This chapter will also include a detailed description of the simulation model which we developed for use in our investigation into the performance and characteristics of our proposed SMS. As part of this description, we provide the algorithm used as well as the results of the demonstrative simulation experiments we conducted. This simulation model is intended to only require the demand mix, system machines, layout configurations and the information on the uncertainties to the system in order to estimate the order completion time. This relationship can be seen in Figure 1.1.

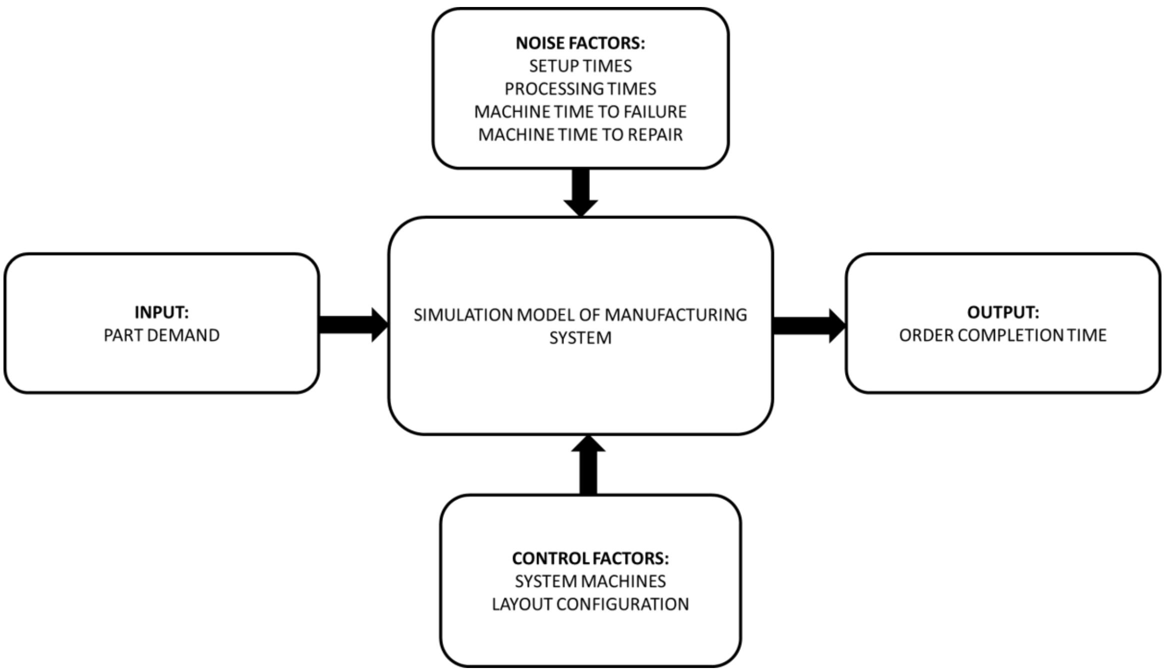


Figure 1.1 System Inputs and Outputs for Proposed Simulation Model

Chapter 4 contains our investigation into the performance of our proposed SMS when subject to different manufacturing environments. We will use simulation to evaluate the performance of manufacturing system using three distinct scheduling approaches; multi-agent systems (our proposed system), right-shifting rescheduling, and dispatching rule prioritization.

The chapter will contain a series of numerical experiments as well as the expected results from this investigation. The performance of the manufacturing system will be based on mean completion time and associated standard deviation after a set number of simulation runs. Chapter 4 concludes with a summary of our findings from all five sets of experiments and a comparison of the performance of the three scheduling approaches evaluated.

Chapters 5 and 6 focus on the specifics of designing the smart manufacturing system. Specifically, in chapter 5, we present a simulation-based optimization model for determining the type and number of each machine resource to have in the SMS. We refer to this problem as the machine deployment problem (MDP). In chapter 6, we present a simulation-based optimization model for determining the layout configuration for our SMS once machine selection has been finalized. The problem is referred to as the machine location problem (MLP).

Finally, in chapter 7, we present our conclusions and recommendations. This chapter begins with a summary of the overall thesis. It then proceeds to an outline of the main contributions of this thesis. After which, the chapter concludes with our recommendations for future studies.

Chapter 2

Design and Operation of Smart Manufacturing System: Background and Literature Review

The real-world manufacturing environment is highly dynamic. Unforeseen disruptions occurring are not uncommon. This results in the need to reschedule in order to minimize the impact of these disruptions on system performance. Industry 4.0 brings with it increased data velocity, availability, and processing which provides allows for improved capability to reschedule. There is a need to determine how best to integrate Industry 4.0 into the operation of manufacturing systems. This requires solving three distinct problems. Firstly, we must determine how the system functions. After which, we must decide how to select resources to design the system such that it is robust. Finally, we must decide how best to organize these resources into the space available.

This work will primarily involve determining how a smart manufacturing system should schedule work in highly dynamic manufacturing environments. These manufacturing environments are subject to various uncertainties that make scheduling difficult; new jobs constantly arriving, variable processing times, machines availability issues, etc. Also, with the increase in mass customization, the impact of these uncertainties is exacerbated by the higher volatility in demand mix and volume that comes with higher variability in products. All of which makes this scheduling problem difficult.

The problem of scheduling work for manufacturing systems is commonly referred to as the job shop scheduling problem. A schedule is the assignment of resources towards completion of a series of jobs. With job shop scheduling, it refers to the determining the work to be assigned to each machine at each time in a production run. The most common approach to job shop scheduling in literature involves assuming little to no uncertainty in the manufacturing system inputs when determining schedule. When the inputs are fixed and deterministic, the problem is generally referred to as the static job-shop scheduling problem. In the context of the real-world, the static job-shop scheduling problem is not representative of the needs of the practical manufacturing environment (MacCarthy & Liu, 1993). This approach fails to make use of real-time data (Cowling & Johansson, 2002) and, as such, cannot be easily adjusted. It assumes that there are no unforeseen events or randomness like new job arrivals, changes in demand, machine breakdown, variable processing and setup times, changes in due dates, changes in job priority etc. However, these events are part of the complexity of real systems and must be considered for the development of efficient schedules. This is the basis of dynamic scheduling, and the dynamic job-shop scheduling problem (DJSSP).

In our research, we propose a smart manufacturing system model for addressing the DJSSP. This solution involves the use of smart machines and smart parts to design a multi-agent system (MAS) for autonomously scheduling jobs that are requested of the system in real-time. There are multiple approaches to addressing the DJSSP. It is important to understand these approaches as well as their advantages and disadvantages. Through understanding these approaches, we came to our decision to employ MAS as the basis for scheduling in a smart manufacturing system.

Once a model for how the smart manufacturing system should operate has been decided, there is now a question of which resources the system requires and how to arrange those resources within the facility. These questions are important as machine selection and location assignment will affect system performance, and as such affect the scheduling decisions made within the facility. As such it is important to understand the various approaches for selecting machines as well as the approaches for designing facility layouts.

The upcoming sections of this chapter are intended to cover the literature relevant to designing a manufacturing system in a highly dynamic environment. We will cover the literature on dynamic scheduling, scheduling under uncertainty, machine selection, and facility layout design respectively.

2.1 Dynamic Scheduling Approaches

Dynamic scheduling approaches fall under three categories; (1) completely reactive scheduling, (2) predictive-reactive scheduling, and (3) Robust pro-active scheduling (Ouelhadj & Petrovic, 2008; Wang et al., 2020).

1. Completely Reactive Scheduling: With completely reactive scheduling no schedule is generated to begin with. Schedules are generated in real-time using priority dispatching rules or other heuristics. This approach is quick and easy to implement. However, solution quality may be poor, and it is difficult to predict the system performance as decisions are made locally without consideration for the global performance measures.
2. Predictive-Reactive Scheduling: This approach involves creating an initial schedule and rescheduling operations in response to real-time events. It is the most common dynamic scheduling approach used in manufacturing systems (Ouelhadj & Petrovic, 2008). These approaches involve making simple adjustments to the schedule whilst considering only shop efficiency and stability. Schedule efficiency is usually reflected in the makespan whilst schedule stability is measured in the starting-time deviation and sequence deviation (Wu et al., 1991; Abumaizar & Svestksa, 1997). The primary objective with

predictive-reactive scheduling is to minimize the impact of disruptions on the performance of the revised schedule due to the changes.

3. Robust Proactive Scheduling: This approach involves building predictive schedules in advance that anticipate the effect of disturbances on the system and maintain system performance within a predictable threshold (Mehta & Uzsoy 1999; Vieira et al. 2003).

Dynamic scheduling problems are typically solved using heuristics, exact optimization algorithms and artificial intelligence, or by instituting policies for addressing system disturbances. In this research, we use an agent-based scheduling approach. We will compare our proposed method against a predictive-reactive scheduling and a completely reactive scheduling approach. For predictive-reactive scheduling, we will make use of partial schedule repair, specifically right-shifting. Schedule repair is employed because schedule repair approaches are the main approaches used by predictive-reactive scheduling systems (Sun & Xue, 2001; Dorn et al.,1995; Abumaizar & Svetska; 1997). Also, schedule repair has been shown to be more effective in maintaining schedule stability than complete rescheduling and requires less computation time (Sabuncuoglu & Bayiz, 2000). For the completely reactive scheduling approach, we will be comparing the performance to that achieved using priority dispatching rules. This is because they are the most commonly employed in real manufacturing systems (Ouelhadj & Petrovic, 2008)

In the upcoming subsections, we will discuss the approaches to scheduling under uncertainty in more detail. We will provide the advantages and drawbacks of each approach as presented in literature. The latter sections of this chapter will address the common causes of uncertainty or disturbance to the manufacturing system that will be investigated in our study (uncertain processing times, new job arrivals, and machine breakdown).

2.1.1 Scheduling Under Uncertainty - Completely Reactive Scheduling

With completely reactive scheduling no firm schedule is generated in advance, all decisions are made locally in real-time (Ouelhadj & Petrovic, 2008). The most common approaches to completely reactive scheduling involve employing priority dispatching rules and agent intelligence (Ouelhadj, 2009; Aytug et al., 2005). Typically, with completely reactive scheduling, there is no consideration, context, or impact of the decisions being made. Work is scheduled for the immediate future using simple, readily available information on the system status. This approach might be better in regard to the global objective. However, they are rarely used due to the computation time and potential schedule instability that may result (Ouelhadj & Petrovic, 2008). Two commonly used approaches for completely reactive scheduling in the literature are dispatching rules, and multi-agent systems. In the upcoming subsections we will discuss the literature on both these approaches to completely reactive scheduling.

Dispatching Rules

Of the two approaches seen in literature, priority dispatching rules are the most prominent (Ouelhadj & Petrovic, 2008). When a machine becomes idle, dispatching rules are used to select the next job to be processed at a machine from a given set of jobs waiting for service. The jobs are selected based on a priority index calculated from job and machine attributes. This approach is usually quick, intuitive, and easy to explain to users and implement. This is why dispatching rules are the most common approach to job shop scheduling in real manufacturing environments

(Demirkol et al., 1997). However, it is important to note that this approach to dynamic scheduling is myopic (Ovacik & Uzsoy, 1997; Ouelhadj & Petrovic, 2008). It relies on local scheduling policies and does not consider future effects of the current decisions. As a result, global scheduling approaches have the potential to significantly improve performance of job shops when compared to dispatching rules (Ouelhadj & Petrovic, 2009; Demirkol et al., 1997). However, these global scheduling approaches are significantly more computationally taxing in comparison to priority dispatching rules.

Some of the earliest literature on dispatching rules are for their implementation in the semiconductor industry (Mohan & Clancy, 1990; Golovin, 1989). It is also roughly around this time that researchers began focusing on dispatching rules as we start seeing a lot of literature on the material (Bhaskaran and Pinedo, 1991; Haupt, 1989; Ramasesh, 1990). Most of this research explored their application in flexible manufacturing systems (FMS). Around the same time, the idea of dynamically selecting dispatching rules as the state of the job shop changes emerged as an extension to the dispatching rules. Wu & Wysk (1989) are some of the early pioneers of this extension. They separate the production cycle into short intervals. At the beginning of each interval, a variety of dispatching rules are simulated, and the one that yields the best performance is implemented in the next time period. There are numerous studies that expand upon this area of research in literature. Piramuthu et al. (1991) developed a decision tree to select dispatching rules when a change to the system state occurs. They used a simulation model to generate data to which they apply an inductive learning algorithm which allowed them to develop their decision tree. Aytug et al. (1994) employ genetic learning to select a population of rules for a given system state. Kim & Kim (1994) developed a scheduling method that varies dispatching rules dynamically in real-time based on information from discrete event simulations using candidate rules. Chen & Yih (1996) use a neural network to predict the appropriate dispatching rule to use under a given system state. Jeong and Kim (1998) expanded on the work done by Kim & Kim (1994). They presented a more systematic framework for simulation based real-time scheduling with eight scheduling strategies. Their results showed that system performance improved more by strategies that more directly respond to changes in system state during the planning horizon. However, it is important to note that these extensions do increase the computational power required to solve the dynamic job shop problem.

Multi-Agent Systems

The other common approach to completely reactive scheduling is the use of agent intelligence in the form of a multi-agent system (MAS). In our application, an agent can be described as an autonomous, decision-making entity within a system. The agent is self-contained with its own characteristics and behaviors. Each agent is able to interact with other agents and their environment (Kang & Choi, 2013). An agent receives inputs from other agents and the environment and decides on what actions to take based on its behavior pattern and then takes that action. A multi-agent system is one that consists of multiple, intelligent agents.

In the context of a manufacturing environment, machines, operators, transporters, and parts can be treated as agents. A manufacturing system comprised of all or even a subset of these agents collaborating with each other to produce parts is an example of a MAS. This collaboration and communication between these agents to produce parts is in the form of assigning jobs to machines at specific times (scheduling). This is the basis of MAS-based scheduling for manufacturing systems.

With a MAS-based scheduling system, each agent is designed with a local objective. For example, a part agent could have the objective of minimizing flow time. A machine agent could have the objective of maximizing its utilization. There is also a global objective for the system as a whole. This is typically minimizing order completion time or minimizing production cost. Each agent acts autonomously to fulfill its objective and, in the process (depending on the system design), should collectively fulfill the global objective.

Typically, with MAS-based scheduling approaches, there is a need to establish how negotiations between agents will occur. The most common approach in literature is to use a market-based negotiation approach, of which contract net-protocol (Smith 1980) and modified versions of this protocol are the most prevalent in the literature (Kang & Choi, 2013). This protocol requires agents to propose bids (and evaluate offers) and other agents to bid on these proposals. Through this auction process, tasks are allocated. With MAS-based scheduling approaches, the schedule is not planned a priori, however, it is generated in real-time by the dynamic interaction between the agents.

There are two architectures that are typically implemented for MAS-based scheduling: autonomous and mediator architectures. With autonomous architectures, the agents directly represent entities within the manufacturing system (jobs, machines, parts, etc.) (Ouelhadj & Petrovic. 2009). These systems are purely heterarchical with each agent making local scheduling decisions with no consideration of the global objective. This approach, whilst it has good performance, may not yield the optimal schedule with respect to the global objective and could prove problematic computationally when there is a large number of agents (Brennan & Norrie 2001; Shen & Norrie 1999; Bongaerts et al. 2000; Shen et al. 2001; Tharumarajah 2001). Brennan and Norrie (2001), Bongaerts et al. (2000), and Cavalieri et al. (2000) present comparative studies that suggest that mediator architectures provide better performance than an autonomous structure in these regards.

Mediator architectures are more of a hybrid control architecture. Similar to autonomous architecture, they have local agents (machines, parts, jobs, etc.) working towards their local objectives. However, there are also mediator agents that act as supervisors to the local agent. These agents' primary objective is the global objective. They can advise, intervene, or update decisions made by local agents in order to resolve conflicts and to ensure that the system tends towards to global objective. Whilst the local agent is concerned with local situations, the mediator agent sees the entire system. This hybrid control architecture results in a system that can react well to disturbances but tends towards to optimal schedule (Barbosa et al., 2015).

The literature on agent-based scheduling and MAS-based scheduling is quite extensive. Ranging from Yet Another Manufacturing System (YAMS) (Paranuk, 1987) to the various MAS-based scheduling for industry 4.0 (Barenji et al, 2017; Chao et al, 2021; Shi et al, 2021). The field is quite interesting, requiring that the agent intelligence be set to ensure good system performance. This process of setting agent intelligence is an area of major interest as it is unclear as to how to select agent intelligence for near optimal performance.

2.1.2 Scheduling Under Uncertainty – Predictive-Reactive Scheduling

Predictive-reactive scheduling is the most common approach to addressing the dynamic job shop scheduling problem (DJSSP) (Ouelhadj & Petrovic, 2009; Mohamed et al, 2018). Most of the definitions reported in the literature on dynamic scheduling refer to predictive-reactive scheduling (Ouelhadj & Petrovic, 2009). Predictive-reactive scheduling can be viewed as a two-step process (Aytug et al., 2005). It is a scheduling/rescheduling process in which a schedule is first created which represents the optimal schedule for the shop floor over a given time horizon assuming no disturbances. This schedule is then revised to account for real-time events/ disturbances to the system to minimize their impact on system performance. The final schedule that is executed (with all its revisions) on the shop floor is referred to as the realized schedule.

With predictive-reactive scheduling, as with any strategy that involves rescheduling, there are two main issues that need to be addressed. The first is when to initiate rescheduling, and the second is which strategy should be employed when rescheduling.

When to Reschedule – Rescheduling Policies

Literature presents three main rescheduling policies: periodic, event-driven, hybrid (Church & Uzsoy, 1992; Vieira et al. 2003; Sabuncuoglu & Bayiz, 2000). In this subsection we will discuss these policies and highlight their advantages and disadvantages.

With the periodic rescheduling policy, rescheduling actions occur at fixed intervals referred to as rescheduling points (Ayutug et al., 2005). These rescheduling points are at time kT, where k is an integer, and T is the time interval between. The schedule created at each of these rescheduling points is executed as planned and is only revised at the next rescheduling point. Any events that occur during or between rescheduling points are ignored. By rescheduling at fixed intervals, the DJSSP is broken up into a series of static problems each of which can be treated as a traditional job shop scheduling problem (JSP). Period rescheduling results in having more schedule stability and less nervousness (Ouelhadj & Petrovic, 2009). However, given that the schedule is followed regardless of events occurring, there is risk of poor system performance. As such, it is important to determine the appropriate time interval T for rescheduling, and this is difficult. Muhlemann et al. (1982), in their investigation of rescheduling frequency, found that the scheduling performance negatively correlates with rescheduling interval. Whilst this might imply that the best approach would be to decrease the interval between rescheduling points, doing so has been shown to have diminishing returns with increased schedule nervousness (Aytug et al. 2005). This is corroborated by a number of other researchers (Sabuncuoglu & Bayiz, 2000; Sabuncuoglu & Karabuk, 1998; Perry & Uzsoy, 1993; Fang & Xi, 1997).

With event-driven rescheduling policies, rescheduling is initiated when an event that could cause a disruption to the system occurs. Most approaches to dynamic scheduling use this approach (Ouelhadj & Petrovic, 2009). Given that rescheduling only occurs as a result of a disruptive event, this policy is less computationally taxing than periodic policy whilst outperforming it (Yamamoto & Nof, 1985). These findings are corroborated by other researchers in the field (Vieira et al. 2000a; Vieira et al., 2000b). The consensus is that event-driven rescheduling is better than periodic.

A hybrid rescheduling policy is one in which the system is rescheduling periodically and in the event of a specified disruption. Typically, these events are machine breakdowns, new job arrivals, job cancellations, and job priority changes. There is very little research into this area, Church & Uzsoy (1992) seems to be the most prominent work in this area. Their findings suggest that effectiveness of periodic scheduling decreases with increasing rescheduling periods. However, the event-driven policy achieved reasonably good performance.

How to Reschedule - Rescheduling Strategies

Most literature on rescheduling present three options for rescheduling: (1) right-shifting, (2) partial schedule repair, and (3) complete rescheduling (Sabuncuoglu & Bayiz, 2000; Cowling & Johanson, 2002; Vieira et al., 2003; Ouelhadj & Petrovic, 2009; Wang et al., 2020).

Right-shifting refers to delaying every subsequent operation affected by a disruption to a point in time where the disruption is resolved. For instance, if a machine in the system breaks down, all operations scheduled on that machine are delayed until the machine is brought back online. Right-shifting is a very particular subset of partial schedule repair.

Partial schedule repair refers to a local adjustment of the nominal schedule in response to real-time disturbances to the system. With this approach, only the operations directly or indirectly influenced by the disruption are rescheduled, all other operations in the schedule proceed as scheduled. For example, if a machine in the system breaks down in a system where there are alternatives or duplicates to the current machine, the operations previously assigned to that machine may be rescheduled to other machines in the system. Once the machine that broke down becomes available again, another rescheduling event is triggered. This approach has been shown to be less computationally taxing than complete schedule repair (Ouelhadj & Petrovic, 2009).

Complete schedule repair refers to the rescheduling of all operations that have not yet been initiated in response to a disturbance to the system. Complete rescheduling may yield the best results, but they are rarely achievable in practice and are very computationally taxing (Ouelhadj & Petrovic, 2009). This approach also results in schedule instability and increased schedule nervousness.

The consensus from the literature is that in most practical settings partial schedule repair or right-shifting are used (Sun & Xue, 2001; Dorn et al., 1995; Abumaizar & Svestka, 1997; Sabuncuoglu & Bayiz, 2000). In fact, there is literature that shows that the partial schedule repair approaches outperform complete scheduling. Yamamoto & Nof (1985) compared the performance of right-shifting against dispatching rules and complete schedule repair. Their experiments show that right-shifting outperforms these other approaches. The partial scheduling approach also offers more schedule stability and predictability (Mehta & Uzsoy, 1999; O'Donovan et al., 1999). It is for this reason that in our study, we will be using partial schedule repair to address disruptions to the nominal schedule we will be using as the basis of our comparison to our proposed MAS real-time scheduling approach.

2.1.3 Scheduling Under Uncertainty – Robust Proactive Scheduling

Robust proactive scheduling involves producing a schedule *a* priori that tries to anticipate the effect of disturbances on the manufacturing system and attempts to minimize its effect on (Wang et al., 2020; Ouelhadj & Petrovic, 2009; Mehta & Uzsoy, 1999; Vieira et al., 2003). The main issue with robust proactive scheduling is how to implement it.

One approach is to develop a schedule that has optimal performance under the worst possible scenario (Daniels & Kouvelis, 1995; Daniels & Carillo, 1997; Kouvelis et al., 2000). The objective of this approach is to determine a schedule with good performance over a wide range of possible disturbances. These studies show that this approach does, in fact, yield results with performance over several scenarios with little deterioration in performance.

Another approach to robust proactive scheduling is to develop predictive schedules by optimizing to maximize predictability of the realized schedule (Mehta & Uzsoy, 1999; O'Donovan et al., 1999; McKay et al., 2000). With this approach there is a primary measure (tardiness, lateness, etc.) and a measure for predictability. Despite using different measures to estimate schedule predictability, all these studies came to similar conclusions. By using this approach, a predictable schedule can be developed with minimal degradation of the primary performance measure.

The primary issue with the robust proactive approach to scheduling is that it is assumed that all the disturbances to the system that could occur are known *a* priori. This is the basis for the schedule development. The problem with this is that it is not as practical as with completely reactive or predictive-reactive approaches. Those approaches are more concerned with adjusting to disturbances. However, the robust proactive approach is subject to the accuracy and completeness of information regarding the possible disturbances that could occur within a manufacturing system. It is for this reason that we do not use robust proactive scheduling in our comparison study.

2.2 Scheduling in an Uncertain Manufacturing Environment

2.2.1 Scheduling Under Uncertain Setup and Processing Times

In real manufacturing environments, processing and setup times for operations are subject to some uncertainty. Operations are typically not completed within a fixed time frame every time they are repeated. This could be due to operator fatigue, differences in operator skill, tooling issues, etc. The uncertainty in these times can have negative effects on schedule efficiency and stability. As such, they should be accounted for in the scheduling process.

There are many approaches to dealing with stochastic processing times when scheduling. These approaches include the use of completely reactive methods, predictive-reactive methods, and proactive scheduling. Lawrence & Sewell (1997) compared completely reactive scheduling (dispatching rules) to other predictive and proactive scheduling methods (global optimization approach). They find that as processing time uncertainty increases, the difference between dispatching rules and global methods becomes less significant. In fact, they suggest that systems with high uncertainty in processing time, completely reactive algorithms can be confidently employed. They also question the benefits of global optimization approaches for these sorts of problems. The literature shows that the predictive approach yields the best solutions. In our research, we will be using a predictive-reactive approach as they yield the best solution. However, dispatching rules may be the most practical approach to handling uncertain processing times. Given that the different approaches to dynamic scheduling have been addressed in previous sections in this chapter, they will not be expounded upon further here. This section will focus on the approaches used to determine nominal schedules prior to production runs.

A fair amount of literature on uncertainty in processing time when job shop scheduling typically extends deterministic models with stochastic reformulations that solve the stochastic model to optimality (Lawler et al., 1999; Shaked & Shantikumar, 1994). With these approaches, uncertainty is incorporated into the model as random independent variable expressed in the form of some predetermined distribution. The intent behind these approaches is to develop a robust schedule by minimizing a given performance measure. For example, the “disjunctive graph” is modified and extended by Fortemps (1997) to deal with fuzzy durations whilst job shop scheduling with the objective of minimizing the makespan. Another such heuristic is the shifting bottleneck heuristic. This was originally developed for the $J_m || C_{\max}$ problem by Adams et al. (1988). It has since been modified and extended by numerous authors to address dynamic scheduling environments (Petrovic & Fraya, 2004; Mönch et al., 2007).

These stochastic models are technically complicated, relying on semi-Markovian decision theory and stochastic dynamic optimization. Also, the nature of this problem is NP-hard as it is an extension of the job shop scheduling problem which has been proven to be NP-hard. Given the complexity of these problems, researchers often rely on heuristics to solve them (e.g. genetic algorithms, tabu search, simulated annealing, etc) (Fortemps, 1997; Chang & Lo, 2001; Yun, 2002; Petrovic & Fraya, 2004). Each of these heuristics has its advantages and disadvantages depending on the nature of the problem being solved. Ishibuchi et al. (1994) demonstrated this in their comparison genetic algorithms, tabu search, simulated annealing, multi-start descent algorithm, and a hybrid genetic algorithm. They found that the performance of these techniques differs with problem size, with GA displaying more robust performance. Ishibuchi et al. (1994) also found that combining these techniques may yield significant improvement over either option. Their hybrid genetic algorithm yielded the best solutions in their simulation study. This has been corroborated by other researchers in this field (Adenso-Diaz, 1996; Liaw, 1998; Chang & Lo, 2001).

In this research, we will be performing a comparison of our MAS-based approach to a predictive-reactive approach to scheduling. As such, it is important to understand the determination of the nominal schedule against which our approach will be compared. Based on the literature, we have decided to use a normal distribution to represent the uncertainty in the processing and setup times. For our determining our nominal schedule, we are focusing on minimizing the makespan as this is the most common objective we have found in our review of the literature.

2.2.2 Scheduling for New Job Arrival

In a manufacturing environment, new job arrivals are unpredictable (Muris & Moacir, 2012), and can have significant impact on schedule stability and efficiency. A new job arrival is an order for a part being made during a production run. Prior to this order being made, there would usually be a nominal schedule being followed in the job shop. This schedule would now require rescheduling to accommodate the new job arrival. New job arrivals may occur randomly (Gao et al., 2015), continuously (Nie et al., 2013) or intermittently (Muhlemann et al., 1982).

There are very few specific surveys of the dynamic job shop scheduling problem that address new job arrivals (Wang et al, 2019). They do show that there are two common categories into which approaches to scheduling with new job arrivals fall under. These are completely reactive and predictive-reactive approaches (Moratori et al., 2010; Wang et al, 2019). Proactive approaches are not typically common in this area of research as it is difficult to account for the effect on random new jobs arriving on system schedule stability and performance.

With completely reactive scheduling, a common approach is to employ dispatching rules (Wang et al, 2019). They are the simplest, and the most frequently used approach for schedulers to address high uncertainty scheduling environments (Dominic et al., 2004). As there is no nominal schedule in place to follow, any new job arriving to the system does not cause a disruption to said system. New jobs arriving to the system would simply be scheduled based on the dispatching rule being followed in the job shop. The main issues to address with using dispatching rules for scheduling under uncertainty is which dispatching rule to follow. There currently is no single dispatching rule that minimizes the all the typical performance measures used for dynamic environments (Blackstone et al., 1982; Haupt, 1989). These measures are makespan, flowtime, tardiness, and tardiness variance (Fattahi and Fallahi, 2010; Moratori et al., 2010; Adibi et al.,2010; Wang et AL., 2019). Whilst no single rule is optimal, there are rules that can be employed that maximize system performance under most metrics. Dominic et al. (2004) performed a comparison of various dispatching rules and combinations of these rules as well. They found the at the combination of most work remaining + first-in-first-out (MWRK_FIFO) and most work remaining + shortest processing time (MWRK_SPT) were found to have good performance in minimizing the most performance measures they evaluated. It is important to note that more complicated rules may perform better but are less practical to implement than simple rules. This is because they require more comprehensive information about the total system in order to implement them. Regardless of which dispatching rule is used in a job shop, the problem remains that the approach, whilst practical, is not the optimal solution to the scheduling problem. This is the major issue with dispatching rules.

Under predictive-reactive scheduling, three approaches to dynamic scheduling are common in the literature when addressing new job arrivals. These are right-shifting, insertion in the end, and match-up algorithms (Moratori et al., 2010). With right-shifting, new operations are inserted into the schedule as required and any affected previously scheduled operations are moved downstream in the schedule. “Insertion in the end”, as the name implies, involves inserting the new job to the end of the nominal schedule. Match-up algorithms focus on trying to return to the nominal schedule as quickly as possible. A rescheduling window is created where the new job is inserted into the schedule. Before and after this rescheduling window, the schedule remains unchanged. Whilst these are three distinct approaches, the common thread between them is that they are partial schedule-repair approaches. This means that only the operations that are affected by the disruption are rescheduled. Complete schedule repair in an environment where new job arrivals occur would result in high schedule instability and floor nervousness (Aytug et al. 2005). The main concern when using predictive-reactive scheduling is balancing schedule stability and efficiency. Right shifting and “insertion in the end” are optimal with respect to stability but perform poorly with respect to efficiency. However, match-up algorithms have been shown to be comparable to right-shifting or “insertion at the end” with respect to stability. However, they have similar performance to complete schedule repair with respect to schedule efficiency when handling new job arrivals (Moratori et al., 2010). However, it is important to note that match-up algorithms have mostly been used for solving simple scheduling problems such as single machine problems (Bean & Birge, 1986; Birge & Dempster, 1995) and single stage parallel machine problems (Bean et al., 1991). Also, there is the question of where to insert these rescheduling windows as well as their duration.

Our proposed MAS-based scheduling approach falls under the category of completely reactive scheduling. Normally, completely reactive approaches are not as good as predictive-reactive approaches with respect to efficiency but are more robust in the sense that they are reactionary and adjust to incorporate disruptions to the system. However, we propose that the incorporation of the appropriate agent intelligence and hybrid control architecture can result in similar scheduling efficiency to predictive-reactive approaches whilst providing the robustness of the purely reactive approaches. This is particularly important and useful in a highly dynamic manufacturing environment where new orders are constantly flowing into the system.

2.2.3 Scheduling in Environment with Machine Breakdown

One common assumption with static job shop scheduling problems is that machines are always available throughout the production cycle. However, this assumption is unrealistic because machines may become unavailable due to preventative maintenance, breakdown or repair (Mehta & Uzsoy, 1999). Addressing machine availability is important as unexpected machine unavailability may result in higher costs due to delays in delivery time, machine repair and material waste (Fazayeli et al., 2016). As such, mitigating the effects of machine unavailability by employing effective scheduling strategies is important, especially in a mass customization environment where customer satisfaction with and, perception of the manufacturer can be adversely affected by delays in product delivery.

As previously mentioned, there are three main causes of machine unavailability. These are preventative maintenance, breakdown, and repair. For this study, we will focus solely on machine breakdown and repair. This is because preventative maintenance is typically deterministic as the event is usually planned based on the facility planner's knowledge of care for each machine. The start and end times for the machine unavailability period are known, and these periods occur at fixed intervals. As such, this information can be easily incorporated into the schedule by the planner. However, machine breakdowns can occur randomly and at random intervals. This makes them difficult to account for whilst scheduling. It can occur due to misuse of the machine or as a result of wear and tear even with proper use. Once a machine breaks down, it must be repaired before it can be used for further operations. This repair time is dependent on the type of damage to the machine as well as the resources available to direct towards the issue.

There is a question of how to incorporate machine availability into the model? Whilst machine breakdown is non-deterministic, it is possible to estimate the mean time to failure (MTTF) and the mean time to repair (MTTR) for each machine from reliability data from the original equipment manufacturer (OEM) and previous experience. In the literature, this data is incorporated into the model in the form of a bounded uncertainty, distribution description or fuzzy data (Fayazeli et al., 2016). In our research, we will be expressing this data in the form of an exponential distribution. We are using the exponential distribution as it is very commonly used in reliability engineering as it phenomenologically and empirically represents the time-to-failure distribution of components, and equipment that exhibit constant failure rates (Kececioglu, 2002).

Literature on scheduling under machine uncertainty commonly focuses on predictive-reactive or robust proactive scheduling strategies. With respect to proactive scheduling, a common approach is to insert idle times into the schedule (Mehta & Uzsoy, 1998; O'Donovan et al., 1999). This is inserting buffer space to the schedule for handling stochastic disruptions. The main issue with this approach is that you must decide how many idle times to insert and where in the schedule to insert them. Typically, surrogate measures of schedule predictability are developed to help determine the location and frequency of these idle time insertions. Other proactive approaches involve the use of heuristics or metaheuristics to solve multi-objective scheduling problems that attempts to maximize schedule stability and efficiency (Fazayeli et al., 2016; Wang et al., 2015; Aloulou & Portmann, 2005; Goren & Sabuncuoglu, 2009; Buddala & Mahapatra, 2019; Nouiri et al., 2017). The issues with these proactive approaches are twofold. The first is that they are computationally taxing and as such there are limitations on the problem sizes that can be solved feasibly. The other is that proactive schedules are developed assuming that all the information is accurate, any disruptions not accounted for in these assumptions may have significant negative impact on the schedule stability.

The other approaches to address machine breakdown when scheduling are predictive-reactive and completely reactive scheduling strategies. These strategies focus on scheduling policy but can also involve rescheduling (Sun & Xue, 2001). With completely reactive scheduling, there is no nominal schedule, and jobs are assigned to machines in real time. This approach usually involves establishing scheduling policies that govern work assignment using dispatching rules or some other artificial intelligence-based approaches (multi-agent systems, neural networks, etc.). With predictive-reactive approaches, a nominal schedule is first created, and rescheduling occurs in response to disturbances to the system. Rescheduling can be in the form of partial or complete schedule repair. Partial schedule repair involves rescheduling only affected tasks in the schedule whilst complete schedule repair involves rescheduling all tasks downstream of the disruption. Kutanoglu and Sabuncuoglu (2001) studied reactive scheduling policies based on rerouting jobs to their alternative machines when their primary machine fails. They find the best policy to employ is dependent on several factors such as machine utilization, mean times to failure and, mean repair times. They find that when downtimes are sufficiently long it is cost effective to reroute. However, if downtimes are short, it is best to wait at the primary machine. Merdan et al. (2011) use multi-agent system (MAS) simulation to assess the robustness of four different rescheduling policies. Like Kutanoglu & Sabuncuoglu (2001), they found that the best policy to employ is dependent on the MTTF and MTTR. However, they also found that when using MAS, the Complete Rerouting rescheduling policy outperformed all other rescheduling policies. Moratori et al. (2010), in their investigation into dynamic scheduling strategies, show that right-shifting is optimal with respect to schedule stability but comparable to total rescheduling with respect to schedule performance.

From our review of the literature, predictive-reactive scheduling appears to be the best approach for dynamic scheduling. This is because these approaches are designed with the potential uncertainties in mind but also have policies in place for handling unforeseen disruptions. As such, we will be comparing our chosen approach to predictive reactive approach. Our application of predictive-reactive scheduling will handle disruptions by right-shifting. We chose right shifting as it offers optimal schedule stability with good schedule performance whilst being easy and intuitive to implement on a real shop floor.

Regardless of the model or approach utilized, there is a need to establish a robustness measure as the basis of evaluating the effectiveness of their proposed solutions. Most of the literature uses a variety of approaches to scheduling with machine breakdown. However, they all seem to focus on similar objectives; minimizing makespan, tardiness, completion times or flowtime (Ahmadi et al., 2016; Xiong et al., 2013; Yuan & Hua, 2013; Leon et al., 1994; Wu et al., 1999; Liao & Chen, 2003; Safari et al., 2010; Hasan et al., 2011; Dong & Jang, 2012; Fayazeli et al., 2016). From our review of the literature, minimizing makespan appears to be the common objective used for this problem. However, our research will focus on the minimization of both the total flowtime and the variability in the flowtime. This is because the total flowtime includes the makespan.

Our proposed solution can be classified as under a reactive scheduling strategy. It is based on using an MAS to schedule jobs in the facility in real-time based on current system state and the global objective of minimizing order completion time or total flow time for the order. As such, our focus is to determine the objectives of the individual agents within the system that will result in the best system performance for any given input scenario. We will compare our approach against a predictive-reactive scheduling strategy that uses right shifting. Our comparison will be against right shifting the schedule by the machine downtime as it has been shown outperform dispatching rules as well as rescheduling by partial or complete schedule repair (Yamamoto & Nof, 1985; Abumaizar & Svestka, 1997).

2.3 Layout Design – Machine Selection

One of the major facilities design activities is deciding the selection and number of machines to use in a system from a given set of alternatives. It has been shown that these decisions can have a significant impact on the quality, flexibility, and productivity of manufacturing systems (Arslan et al., 2004). Determining the optimal number of machines to use in a manufacturing system has been explored as an aspect of multiple different problems in the literature. Early studies referred to this as the "machine requirements problem" (Miller & Davis, 1977). It was modeled and solved using mixed integer programming techniques (Miller & Davis, 1978; Kusiak, 1987). These models typically accounted for machine setup and processing times, costs, and availability. Behnezhad and Khoshnevis (1988) extended this model by incorporating a machine's production rate over time, known as the manufacturing progress function. However, Miller and Davis (1977) highlighted the limitation of these models, as they tend to oversimplify manufacturing systems by disregarding the interrelationships between the number of machines, and the system inputs, as well as ignore the interdependence between machines.

Another related problem is the "machine selection problem," which involves evaluating and selecting the optimal number of machines using decision-making methods such as fuzzy multi-attribute decision-making (Wang et al., 2000). Chtourou et al. (2005) developed an expert system to systematically add and remove machines to optimize machine numbers, while Karim & Karmaker (2016) used analytic hierarchy process (AHP) to determine weighting factors for machine selection criteria and ranked the machine options using TOPSIS.

In literature, the "machine duplication problem" also commonly arises when considering the number of machines in a manufacturing system. This problem involves decision-makers purchasing additional units of existing machines in a cell as a decision variable. Agnetis and Oriolo (1995) provided an early formulation to analyze optimal solutions for machine duplication in a manufacturing system with two jobs, and subsequent research considered multiple considerations such as subcontracting bottleneck parts (Logendran & Puvanunt, 1997), group layout with unequal area requirements (Kia et al., 2015), operator assignment and cellular layout (Mehdizadeh & Rahimi, 2016), integration of cellular and distributed layouts (Defersha & Hodiya, 2017), and alternative process routings (Mohammadi & Forghani, 2017). To account for the uncertainty of cost, machine capacities, and part demands when determining machine duplication, fuzzy programming has been applied (Arikan & Güngör, 2005; Safaei et al., 2008). Additionally, research has considered the integration of duplicate machines with cell formation, layout, and scheduling (Feng et al., 2018; Feng et al., 2019; Rahimi et al., 2020).

The computational complexity of the machine duplication problem was analyzed by Agnetis and Oriolo (1995), who demonstrated its NP-hardness when dealing with two jobs. Consequently, heuristic/meta-heuristic approaches have been proposed to solve various formulations of machine duplication problems, such as Tabu search (Logendran & Puvanunt, 1997), simulated annealing (Kia et al., 2015; Mehdizadeh & Rahimi, 2016; Defersha & Hodiya, 2017), genetic algorithms (Feng et al., 2019), and vibration damping optimization (Mehdizadeh & Rahimi, 2016; Rahimi et al., 2020). Hybrid procedures that combine heuristic/meta-heuristic algorithms with exact methods, such as cluster analysis + integer programming (Bortolini et al., 2011), genetic algorithm + dynamic programming (Mohammadi & Forghani, 2017), and genetic algorithm (or simulated annealing) + linear programming (Feng et al., 2018), have also been proposed to improve the efficiency of algorithms. Additionally, the Taguchi method has also been used to determine the parametric settings for meta-heuristic algorithms (Mehdizadeh & Rahimi 2016; Feng et al. 2018; Rahimi et al. 2020).

In this work, we will be using a combination of simulation and a metaheuristic (genetic algorithm) as the means to solve the machine selection problem. As such we will be presenting the mathematical model for evaluating the results of the simulation. This model will guide the decision of the genetic algorithm. In our model, there are three metrics by which we will evaluate the performance of the system; the cost of the system, as well as the order completion time and its associated standard deviation. The system being considered will be a manufacturing environment subject to various sources of uncertainty; machine availability, demand uncertainty, variable setup and processing times, etc. Our model is intended to select the machines for a robust manufacturing system given a set of possible machine options.

2.4 Layout Design – Facilities Layout Design

Facilities layout design is a crucial aspect of manufacturing and has been extensively studied in literature (Sun et al., 2018). A vast body of literature has been surveyed by Balakrishnan & Cheng (1998), Meiler & Gau (1996), and Kusiak & Heragu (1987). However, despite the vast amount of research in this area, it is still considered to be in its early stages (Hosseini-Nasab et al., 2018). The emergence of Industry 4.0, which introduces smart manufacturing and cyber-physical production systems (CPPS), coupled with the increased availability and velocity of data, has led to a more demanding market (Perez-Gosende et al., 2021). This necessitates a shift towards more flexible manufacturing systems that can accommodate greater product variety while utilizing smaller lots. As such, facilities layout design remains a critical research area for manufacturers to optimize their operations and meet the demands of Industry 4.0. Facility layout can be broken down into three main problems: departmental planning, the facility layout problem (FLP), and the machine location problem (MLP).

Departmental planning is concerned with determining the appropriate layout type for the manufacturing facility based on demand volume and product variety. It involves taking into consideration the flow, space and activity relationships. (Tompkins et al., 2001). There are four conventional layout types; product, fixed location, functional, and cellular (group technology) layouts. Each of these types are designed to perform best when subjected to specific levels for product demand mix and production volume. The most common of these layout types studied in literature are cellular layouts and functional layouts. A functional layout consists of groupings of machines that serve similar functions, for example, a grouping of cutting machines into a cutting department coupled with grouping of drill presses into a drilling department. These sorts of layouts are good for low volume-high variety manufacturing environments. However, a functional layout is usually subject to poor material handling efficiency and high scheduling complexity which can adversely affect system performance (Sarper & Greene 1993; Montreuil, 1999). A cellular (group technology) layout is based pairing machines into groups based on part families. Part families are a grouping of parts that share a number of similar operations that require the same set of machines. With group technology, these machines are grouped into departments that are dedicated to a specific set of parts. These layouts tend to become inefficient when the demand mix for products changes (Heragu et al., 2000).

The facility layout problem (FLP) and the machine layout problem (MLP) are both critical tasks in manufacturing system design. These problems involve determining the optimal physical arrangement of facilities and machines to efficiently and effectively utilize available resources, while maximizing productivity and minimizing costs (Ku et al., 2011; Altuntas & Selim, 2012). Efficient layout solutions can significantly reduce material handling costs (MHC), while an inefficient layout can result in congestion and increased MHC (Saraswat et al. 2015).

Therefore, the proper design of manufacturing layouts is crucial for ensuring efficient and cost-effective production systems.

There is an extensive body of literature on the FLP dating back to the 1950s. Koopmans and Beckman (1957) were the first to model the FLP as a quadratic assignment problem (QAP). This QAP model has frequently been used to model the FLP (Kusiak & Heragu, 1987; Perez-Gosende et al., 2021). Lawler (1963) was the first to formulate the FLP as a linear integer programming problem with modifications to the common QAP. The second formulation for the general FLP is a quadratic set covering problem (QSP) (Bazaraa, 1975). Kaufman & Broeckx (1978) further extended the QAP model for FLP. They developed a linear mixed integer program with the smallest number of variables and constraints amongst all the other inter programming formulations of the QAP at the time. All these models focus on minimizing the cost of placing a facility in a specific location as well as minimizing the flow between facilities. There are other models developed focusing on different measures. Some other measures that appear in the literature are transportation cost of materials, and closeness rating measure. Rosenblatt (1979) and Dutta & Sahu (1982) developed heuristics to solve FLP by optimizing these two measures. More recently, the most widely used approaches for modelling the FLP involve mixed integer non-linear programming (Gulsen et al., 2019; Vázquez-Román et al., 2019; Yang et al., 2019) and mixed integer linear programing (Allahyari & Azab, 2018; Ejeh et al., 2018; Kia et al., 2014; Klausnitzer & Lasch, 2019; Xiao et al., 2017).

The literature on MLPs is not as extensive as that for FLPs (Perez-Gosende et al., 2021). However, similar to FLP's, the earliest works relating to machine layout model the problem as a QAP (Hassan, 2007). The QAP is suggested for all machine layout types that are typical in MLP's (Sarker et al., 1991; Kaklu & Rachamadugu, 1992). The MLP shares a number of common objectives with the FLP. Some objectives are the minimization of backtracking (Sarker et al., 1991, Kouvelis et al., 1992), minimization of material handling, operating and fixed costs (Kouvelis & Kiran, 1990), minimization of travel time (Sarin & Wilhelm, 1984, Heragu & Kusiak, 1988), maximization of throughput (Co et al., 1989). However, there are some objectives that are unique to the MLP such as minimization of the maximum number of transfers between machines (Leung, 1992), and minimization of the average number of machines that are visited by the parts (Kouvelis & Kim, 1992).

FLPs and MLPs are often modeled as Quadratic Assignment Problems (QAPs), which have been shown to be NP-complete (Sahni & Gonzalez, 1976). As a result, various heuristic and metaheuristic methods have been proposed to tackle these problems. These methods include branch and bound (Gilmore, 1962; Lawler, 1963; Bazaraa, 1975), cutting plane algorithms (Bazaraa & Sherali, 1980), genetic algorithms (Kulturel-Konak & Konak, 2013), and simulated annealing (Allahyari et al., 2018). Hybrid algorithms have also been developed recently to enhance the solution quality for these problems. For example, Kulturel-Konak & Konak (2015) proposed a large-scale local search (LSLS) based on simulated annealing (SA) hybridization and MILP, which they named LS-HSA. Kulturel-Konak (2017) developed a metaheuristic solution approach called VNSAM that combined variable neighborhood search (VNS) and SA with an MINLP model. Additionally, Feng et al. (2018) introduced two hybrid approaches, GALP and SALP, to solve an MINLP model by combining GA and SA, respectively, with LP.

New layouts have been proposed to overcome the drawbacks of traditional layout types. Overlapping cells were suggested by Irani et al. (1993), while Suresh & Meredith (1994) proposed machine sharing between cells, and Montreuil (1999) proposed fractal cells. With the advent of Industry 4.0, it is now possible to create novel layouts that are more efficient but less intuitive. These layouts are typically a blend of conventional layouts and are referred to as hybrid layouts.

Choosing the appropriate layout is crucial as the efficiency of the system can decline when there are fluctuations in product volumes, mix, or routings (Afentakis et al., 1990; Norman & Smith, 2001). To address this issue, it may be desirable to develop robust layouts. Early research on the robustness of layouts was carried out by Rosenblatt & Lee (1987), who proposed a model for designing layouts that can handle uncertainties. Palekar et al. (1992) continued this line of research by incorporating uncertainties into facility layout design. They employed dynamic programming to solve the model, which can handle both small and large problems.

Studying the various layouts configurations and their impact on system performance is crucial. This is especially true with the SMS manufacturing environment where new, hybrid layouts may be better suited. As manufacturing environments increasingly shift towards high product variety with high production volume, the need for new types of facility layouts has been highlighted (Benjaafar et al., 2002). Hybrid layouts have been proposed as a solution to this issue (Ariafar et al., 2011; Irani & Huang, 1989). However, the suitability of each layout depends on the production conditions. After all, Zolfaghari & Roa (2006) demonstrated that the system inputs can significantly impact which layout (functional, cellular or hybrid) performs better. Therefore, it is imperative to develop a model to determine the optimal layout configuration.

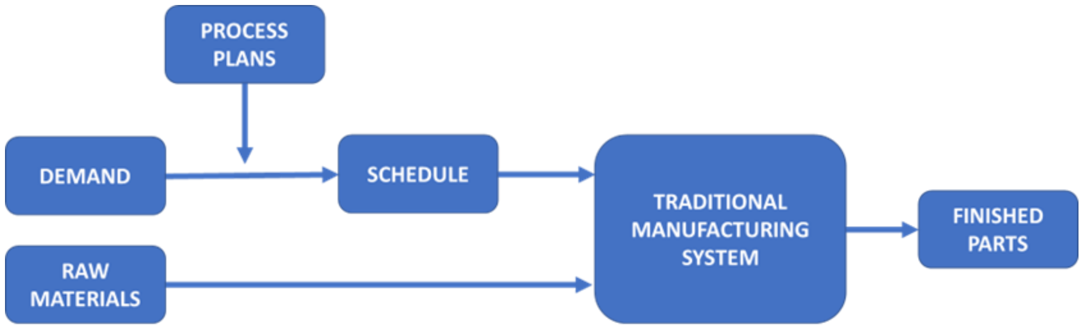
The present design criteria fail to fully account for how layout impacts crucial performance measures, such as cycle time, throughput rate, and congestion (Benjaafar et al., 2002). Instead, these criteria rely on proxy measures, which may not be effective under a range of operating conditions. Therefore, there is a demand for a new category of layouts, and corresponding design models and solution methods for determining them.

Chapter 3

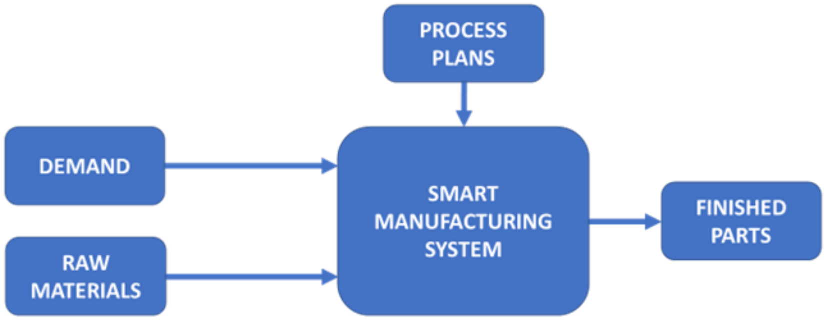
Smart Manufacturing System Model

A traditional manufacturing system consists of machines and transporters that exist within the spatial constraints of a manufacturing facility. Within that facility, operators are tasked with the production of a predetermined set of parts using this system's resources. They make use of the system's machines to perform value-adding operations on work-in-process (WIP), taking them from being raw materials to finished parts. Transporters move the WIP between machines for processing. The WIP flows through the system based on a schedule or some pre-established scheduling principle. These schedules are determined prior to production based on processing plans for each part are developed given the machines available in the system.

The smart manufacturing system (SMS) we propose is an extension of the traditional system. The primary difference between the SMS and traditional manufacturing system is that scheduling decisions are made in real-time by the systems parts and machines self-organizing whereas in the traditional manufacturing system, scheduling decisions are made by the job shop manager. This difference is illustrated in Figure 3.1.



a) Traditional Manufacturing System Overview



b) Smart Manufacturing System Overview

Figure 3.1 Comparison of SMS to Traditional Manufacturing System

The SMS we propose is provided with the processing plans for the parts required of the system. It uses this information to decide the best processing route options for fulfilling the current demand to the system for parts in real-time. The system can make these decisions as the main components of the system (parts and machines) are given agency. Machines are given agency with respect to deciding when to accept parts. Each part ordered has agency in deciding which available machine will be performing value-adding operations to produce it. The system's parts and machines negotiate with each other to produce the finished. As such, the SMS we propose can be viewed as a distributed problem solver tasked with solving the problem of producing parts as efficiently as possible.

The ability to schedule (and adjust the schedule) in response to real-time events (dynamic scheduling) is the major benefit of our SMS. Mass manufacturing environments are subject to numerous sources of uncertainty and by dynamic scheduling we reduce the impact of these uncertainties on the system efficiency is reduced. Our proposed SMS falls under the category of completely reactive dynamic scheduling. Specifically, our SMS employs multi-agent system-based (MAS-based) scheduling. This approach does not require schedule development prior but offers more intelligence in its scheduling than scheduling using dispatching rules.

The SMS can be viewed to consist of two interconnected domains; the physical domain, and the agent domain. A breakdown of the model's domains can be seen in Figure 3.2. The physical domain encompasses the machines, transporters, and parts that exist within the system. The agent domain encompasses the intelligence of the system's physical components. The upcoming sections of this chapter will describe our SMS model in detail. We will begin with an overview of the SMS as a whole. This will be followed by detailed descriptions of the physical domain of the model and the agent domain of the model. The chapter will conclude with the verification of the model through simple numerical experiments.

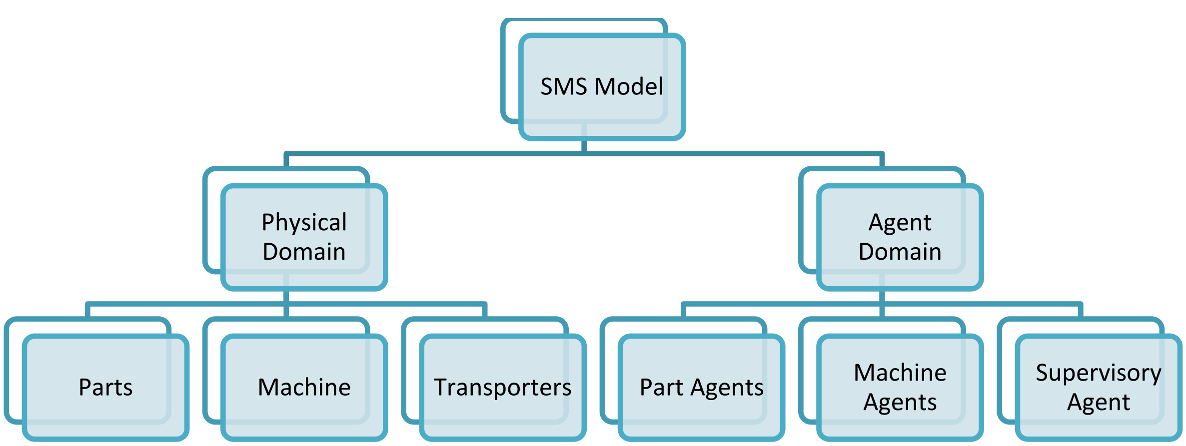


Figure 3.2 Breakdown of the SMS Model

3.1 Smart Manufacturing System Overview

The SMS we propose is a type of cyber-physical production system (CPPS). Wang et al. (2016) describes a CPPS as consisting of four layers: a physical layer, network layer, cloud layer and, supervisory and control layer. A layer can be viewed as a category to which a resource or entity belongs based on its function or characteristics. Each of these layers communicate and coordinate between each other to facilitate the manufacture of a set of products. They allow for the CPPS to be self-organizing and autonomous. We have chosen to represent the same concept in the form of two domains that communicate with each other; the physical and agent domains.

The physical domain consists of the physical resources that exist within the system. These are the parts, transporters, machines, operators, and the facility floorspace. The facility houses the machines, parts, buffer spaces, operators, and transporters. The parts are transported by the transporters between machines to have value-adding operations performed on them. These physical resources are assumed to be “smart”. For example, machines are equipped with sensors with receiving and transmitting capabilities. These sensors allow them to communicate their current state to the system as well as other requested information related to projected operation completion times for certain tasks. The machines are also equipped with a microprocessor for localized computation of projected operation completion times. Similarly, parts in the system each have an RFID tag to communicate their current location in the facility as well as their current stage in the processing route. This information is communicated to the cloud which uses it to make decisions for the part. The system’s transporters are assumed to be automated guided vehicles (AGV’s). These AGV’s have limited autonomy, they simply receive information from the system as to where to pick up parts, and where to deliver them to.

The agent domain consists of all of the system's data storage and data processing capabilities which can be scaled on demand. These include things such as software-as-a-service (SaaS), infrastructure-as-a-service (IaaS), and platform-as-a-service (PaaS). It is necessary to handle the vast amount of data produced by a smart manufacturing system. The system is assumed to have the capacity to store data on the current status of all components of the system in real-time. This provides each agent within the system with an overview of the entire system to guide the agent's decision-making. When a change occurs in the system, the agent responsible for that change makes updates to the cloud. The agent domain also includes the rules by which the system components negotiate with each other and their environment. This is each machine's and part's decision-making process. Each part and machine in the system has a representative agent that guides its decision-making in real-time. For the purpose of resolving system conflicts between agents, the system also includes a supervisory agent.

To facilitate communication between the agent and physical domain we require an industrial network. The industrial network consists of the infrastructure that allows all the system components to transmit information between each other. This is typically in the form of an industrial local area network (LAN) or industrial wireless area network (WAN). This network is also the means by which each agent in the system would update its status on the cloud. In our model, it is assumed that all parts, machines, operators, and transporters can communicate and coordinate with each other in real-time over this network. As such, this aspect of the model is not explicitly modelled as is implicitly modelled in MAS agent interaction.

It is also assumed that there are system terminals in the form of computers, tablets and phones through which people can access the performance and diagnostic data from the system and make adjustments to the system as necessary. In the upcoming subsections, we will discuss the components of the smart manufacturing system as well as provide an overview of how the system works.

In the SMS, decisions are primarily made in the cloud and then executed by the physical resources. Communication between the cloud and physical resources occurs over an industrial network. The cloud is provided with information relating to the parts in the systems (current stage in production and process plan network) as well as machine status (busy or idle as well as capacity and location). It uses this information to determine the schedule for jobs in real-time. This dictates what machines have been assigned to work on each part that requires service. This information is then relayed back to the physical system, which then executes the schedule (transports parts to the queue of respective assigned machines). Upon any change to the physical system (i.e. machine completes/starts a job, part exits system, etc.), the physical system relays this information back to the cloud. The supervisory and control layer primarily observes the system performance and adjusts the objectives that govern the system performance. An overview of the interaction between these layers can be seen in Figure 3.3.

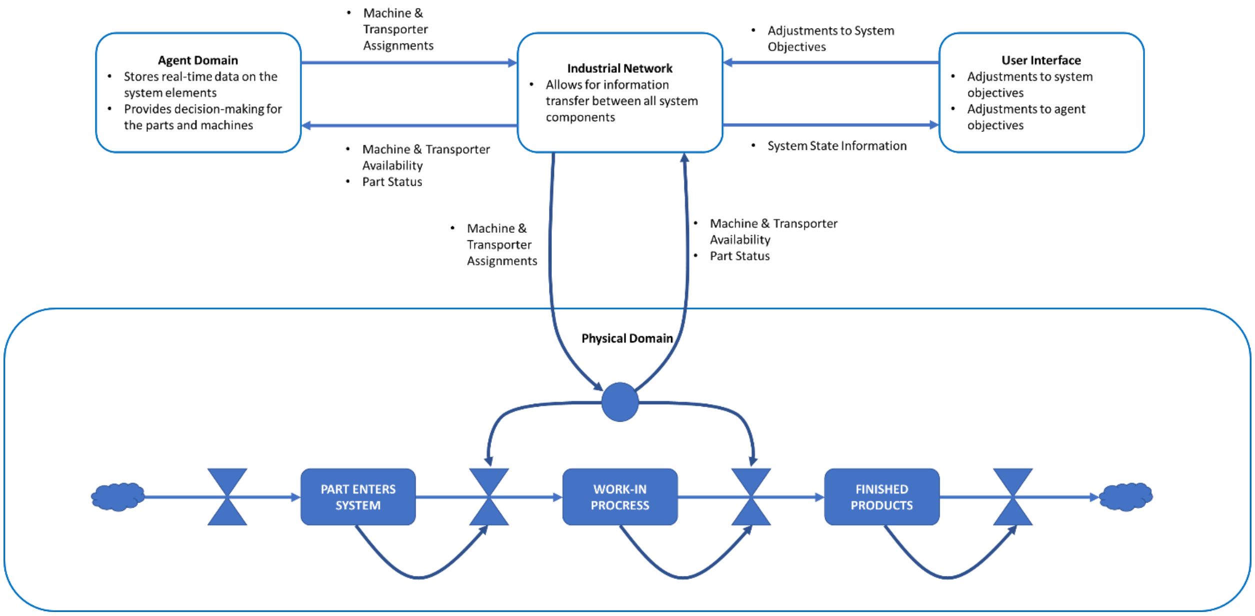


Figure 3.3 Interaction Between Cloud and Physical Layer of Resources

3.2 Physical Domain of the Smart Manufacturing System

In this research, we model the machines and parts as well as the spatial specifications of the facility. We consider these three elements (parts, machines and facility grid) to be the "core" elements of physical domain of the SMS, and as such provide more detail on their representation is provided. In this section, we will provide a description of how we modelled these elements in this research.

3.2.1 Machines

Machines are resources that perform value-adding operations to WIP. In our model, we use $M = \{m_1, m_2, \ldots, m_{nm}\}$ to represent the set of nm distinct types of machines in the system. Similarly, we use $O = \{o_1, o_2, \ldots, o_{no}\}$ to represent the set of all no value adding operations that can be performed within a manufacturing system. The distinction between machines and operations is important as the number of operations that a system can perform does not need to match the different types of machines in the system.

Our model allows for duplicate machines and similar machines to exist within the system. Duplicate machines are machines that can perform the same operations for the same set of parts and have the same set up and processing times for each operation they can perform. Similar machines may vary in the set of common operations they can perform or have varying set up and processing times or both. It is important to note that in our model, the notation for operation refers specifically to the operation being performed and makes no inference to the machine being used to execute the operation. For example, if o_1 refers to the operation drilling, and $\{m_1, o_1\}$ is the notation for perform drilling at machine m_1. Then, similarly, $\{m_2, o_1\}$ would be the notation for perform drilling at machine m_2.

An operation is an action that transforms the WIP and adds value to it. These are actions such as cutting, drilling, milling, planing, etc. The number of operations that can be requested of the system does not have to equal the number of distinct machines in the system ($no \neq nm$). Whilst each machine in the system is capable of at least one operation, they may be capable of multiple different operations. The machine, however, can only perform one operation on one part at a time.

A machine's ability to execute an operation is part specific. A machine being capable of performing an operation does not mean it should be able to service all parts that require that operation. They can be capable of performing an operation for one part but not the other. For example, one drill press may only be able to fit a fixed set of bit sizes. Therefore, any part requiring a hole larger or smaller than the bits this drill press can hold cannot be processed on this machine.

In our proposed system, there is no queueing in front of machines for work. Machines can only accept parts' requests when they are free to do so. Rather, any parts that have not been assigned to a machine for its next operation once it has been processed by the current machine simply waits in a buffer space adjacent to the current machine for its machine assignment. By doing this, the current machine is freed up to receive new part requests and is not forced to be idle by being occupied by completed work with nowhere to go. However, this requires that adequate buffer space be provided to hold parts between operations.

The overall system dynamics for a machine are shown in Figure 3.4. A part waits to be assigned to a machine for work. Once this part is assigned to a machine in assigned a part, the part is transported to the machine. It waits in for the machine to be set up to service it. After which, the part is serviced by the machine. Once the part is serviced by the machine, it is released from the machine to transported to its next machine in its processing path or held in a buffer space. This process continues until the production cycle ends.

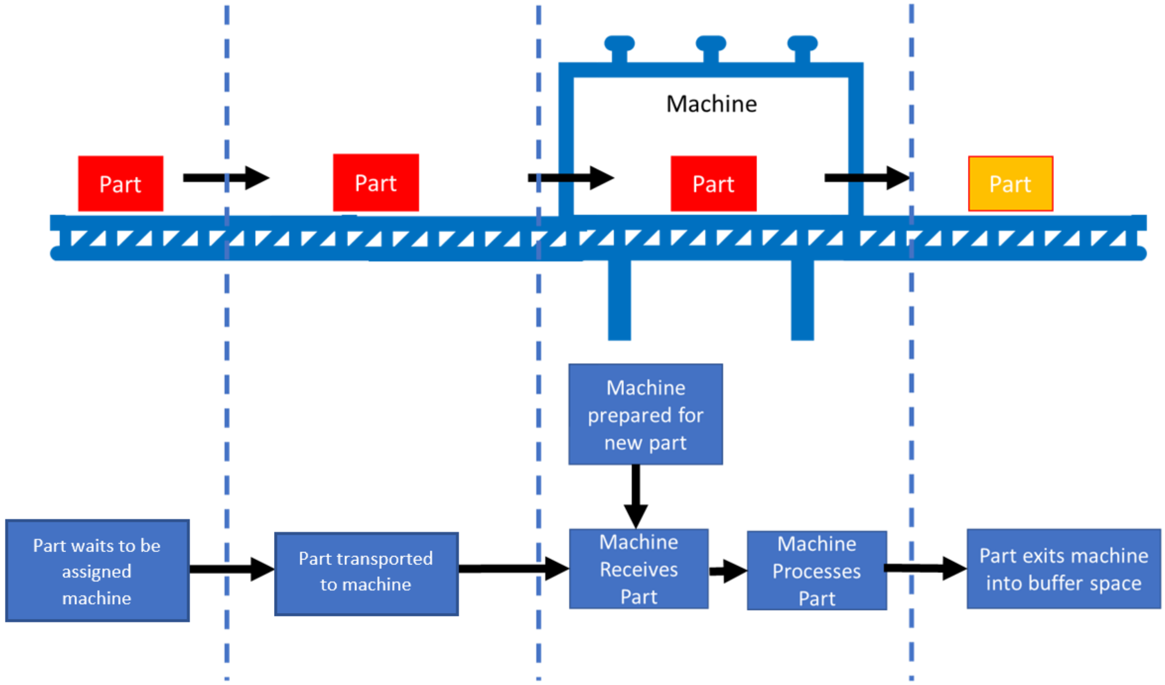


Figure 3.4 System Dynamics for Machines

Machine Model Representation

The key attributes that determine how well a machine performs an operation are the set up and processing times, and these are part specific for each machine. Machine reliability information is also very important. This is information relating to mean time to failure (MTTF) as well as mean time to repair (MTTR) for each machine. In our representation of machines, we focus on the collection of these pieces of information when describing machines. Setup time is the time spent preparing a machine to perform an operation on a part. Once, set up is completed, the operation can be executed on the part. The time required to complete the operation on a part using a given machine is the processing time on that machine. Both set up and processing times are specific to the machine the operation is being performed on, and the type of part being operated on. Whilst set up and processing times are machine and part specific, it is important to note that two machines may be capable of performing the same operation for the same part but not have the same set up and processing times. This is because different machines may have different physical specifications. For example, two different drill presses may have different securing mechanisms and as a result require different setup times to perform the same operation on the same part type.

In real life manufacturing environments, the set up and processing times can be uncertain. This could be due to the differences in operator capabilities or other stochastic factors that influence operator performance (fatigue, errors, skill difference, experience, shift changes, etc.). Also, machine breakdown occurs unexpectedly, and the time required for repair depends on the damage done to the machine. Given the uncertainty, these times are best represented in the form of a distribution. In this research, we have decided to represent the setup times them in the form of a normal distribution. Similarly, the MTTF and MTTR are represented in the form of an exponential distribution. This is based on the convention observed in the literature.

In our model, machines are classified according to the operations they can perform and the parts they can service and their reliability information. We represent this using two different matrices. One of these is a machine-operation-part (MOP) relationship matrix (seen in Figure 3.5a), and the other is the machine reliability information (seen in Figure 3.5b). Each machine has a corresponding MOP that defines its specific capability. The MOP lists the operations the entire system can perform as well as the different parts that the system can produce. A sample MOP matrix can seen in Figure 3.5a. This figure shows the mean (μ) and standard deviation (σ) for the setup and processing times for a three-part system. The MOP depicts the operations the a given machine can perform and which part it can perform it for. If an operation is possible for a given part on a given machine, it has numerical values assigned to its distribution, else, it is assigned an entry of ‘inf’. In Figure 3.5a we see that the given machine is capable of performing operation o_1 for part p_1, operation o_2 for part p_2, and operation o_3 for part p_3. The “inf” indicates that the machine is not capable of performing the operation for the given part. In this example, setup time for operation o_1 would take an average of 2 time units. Processing time for the same operation would take an average of 5 time units.

	O1		O2		O3	
	S	P	S	P	S	P
P1	µ: 2 σ: 1	µ: 5 σ: 2	inf	inf	inf	inf
P2	inf	inf	µ: 2 σ: 1	µ: 4 σ: 1	inf	inf
P3	inf	inf	inf	inf	µ: 3 σ: 1	µ: 6 σ: 2

a) Sample MOP

MTTF	MTTR
Distribution: Exponential Mean: 10	Distribution: Exponential Mean: 5

b) Sample Machine Reliability Data

Figure 3.5 Machine Information Representation

3.2.2 Parts

Parts are the end product of performing a series of value adding operations on WIP to transform it from an incomplete part to a finished part. We represent the set of distinct parts (or different part types) that can be produced by a system using $P = \{p_1, p_2, \ldots, p_{np}\}$. By distinct parts, we mean that the parts are not perfect duplicates of each other. In our model, we have chosen to treat variants of parts as different part types that the system can produce. For example, parts p_1 and p_2 can both be tabletops. However, p_1 could have a different surface finish than p_2 but otherwise be exactly the same. In our model, they would be treated as two different types of part.

Producing a part requires the execution of a subset of the operations that the system's machine can perform. This subset of operations must be executed in a specific sequence and each operation in this subset must be completed. This specific order in which a series of different operations must be performed in to produce a specific part is the operation sequence. For example, creating a wooden tabletop may require cutting a piece of wood into the appropriate size, planing the surface and treating the surface. In this example, the WIP is the wood we begin with, and the part is the tabletop. The part requires three operations; cutting, planing and surface treatment. Each of the operations performed on the WIP transforms it into a form closer to that of the desired part. Figure 3.6 shows the possibilities for WIP flowing through a six-machine system. WIP enters the system and is assigned a route to follow, at each machine an operation is performed that alters the WIP (depicted in Figure 3.6 by a change in color).

The operations that make up the operation sequence required to produce a part are not defined by the specific machine performing the operation. By this we mean, if a part requires drilling to be performed, the requirement does not specify which machine must be used to perform this operation. As such, any machine in the system that can perform drilling can be used to perform the operation. Knowing the operation sequence required to process a part does not convey the information on how long each operation would take. This information can only be obtained by knowing the machine used as well as the part and operation required. We will also need to know the machine route being used. In this research, we distinguish between the operation sequence a part follows and the machine route path through which it flows through the system. For each operation sequence, there may be multiple machine route options within a given system. This is the case with the example depicted in Figure 3.6 which depicts a system that produces one part. There are three possible machine route paths and two possible operation sequences. Two machine routes use operations o_1 and o_5, and the other uses operations o_3 and o_4. The machine routes using operations o_1 and o_5 use entirely different machines to perform the same operations. These machines may be duplicates (same processing and setup times) or similar (different processing or setup times).

Each part may have multiple operation sequences that can be used in their production. These operation sequences may consist of permutations of the same operations. For example, if there are *n* operations required to produce a part, and the order in which they are executed is unimportant, then there are *n*! potential operation sequences for that part. Alternatively, each operation sequence may consist of sets of operations that are distinct from each other. The set of all possible operation sequences that can be used for manufacturing of a specific part using a given set of machines is represented in the process plan network for that part. The total set of operation sequences for each part represents the system's overall flexibility with respect to the production of the given part. Each part has its own distinct process plan network. A sample process plan network for the system depicted in Figure 3.6 can be seen in Figure 3.7. In Figure 3.7, we see that the part can be produced using two routes. One route o_2 and o_4, and the other using o_3 and then o_5.

In our model, we assume that all operation sequence options in a given part's process plan network are available to be used interchangeably during the production period. This means that the specific operation sequence chosen to make a specific type of part may be different at different times during the production cycle depending on the system status (machine availability, part arrival times, etc.). It is assumed that during the production period, we can use alternate sequences if the system machines' statuses make it viable.

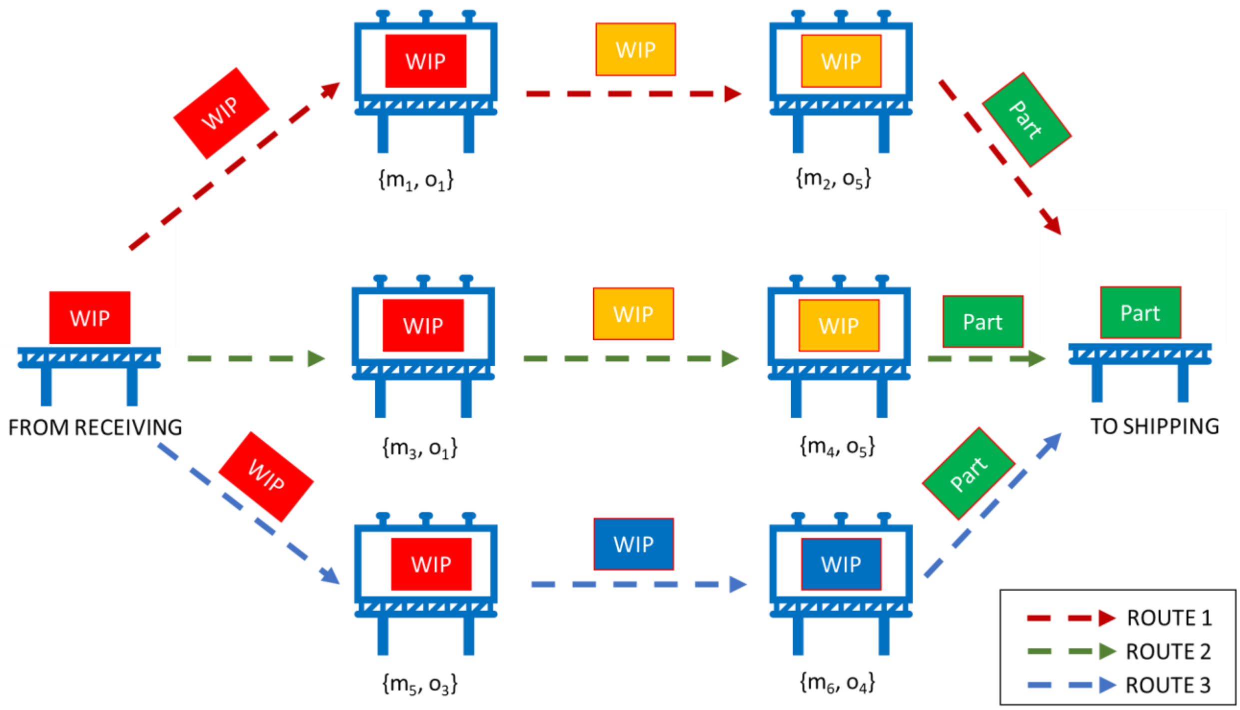
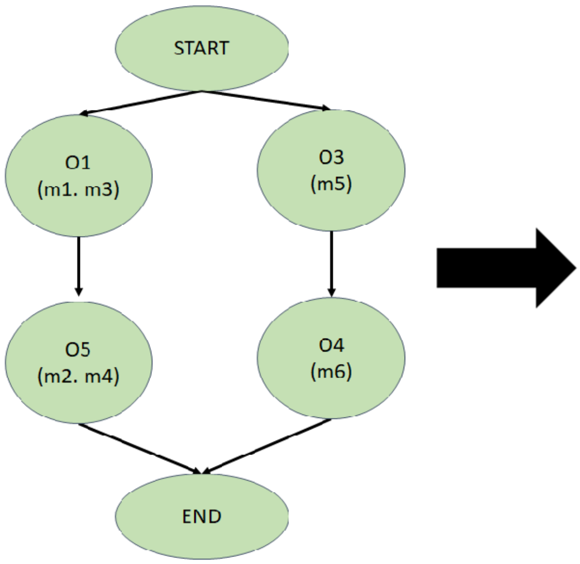


Figure 3.6 Machine Route Options for Sample Part In 6 Machine System

Part Model Representation

In our model, we represent the information from the process plan network in the form of an adjacency matrix. A sample of this matrix can be shown in Figure 3.7. The matrix depicts every operation sequence that can be used to produce a part. It begins from a start node and ends with an end node. Between the two, each column shows the next operation that can be executed given the preceding operations. For example, for the part depicted in Figure 3.7, we have a system capable of 5 operations. For this specific part, we can start its production using operation o_1 or o_3. Assume operation o_5 is selected. From o_1 the only available next processing step is o_5.

This is the final operation along the sequence and as such, it leads to the end node. For this part, there are two possible operation sequences (o_1, o_5) or (o_3, o_4). Either option is allowed in our model.



	S	O1	O2	O3	O4	O5	E
S		0	0	0	0	0	0
O1	1		0	0	0	0	0
O2	0	0		0	0	0	0
O3	1	0	0		0	0	0
O4	0	0	0	1		0	0
O5	0	1	0	0	0		0
E	0	0	0	0	1	1	

Figure 3.7 Sample Process Plan Network (left) and Operation Sequence Adjacency Matrix (Right)

3.2.3 Facility Floorspace - Facility Grid

The facility floorspace refers to the spatial dimensions of the facility from the top view. All resources (machines, work-in-process, raw materials and transporters) required to produce all parts must be able exist within the facility floor space. We assume that the facility has enough space to accommodate these resources. The positioning of these resources has a direct influence on the flow distance and flow path for each part and thus, the material handling costs. As such, the facility must be organized such that the flow distance can be effectively minimized whilst having feasible flow paths for all parts. All machines are assigned fixed positions within the facility. Parts (in the form of work-in-process) are transported through the machine stations by the use of transporters.

For a flow path to be feasible, the path along which a part is processed must be unobstructed by other machines. By this, we mean that there must be a route to get to each machine required that does not run through another machine. Flow path feasibility also requires that each machine on the route be accessible. For example, if all the machines in a facility are all tightly packed into a corner of the facility, all machines closest to the corner would effectively be inaccessible. This would be a situation where the distance would be minimal, however, flow paths effectively do not exist. To have feasible flow paths, there must be space around each machine to allow access to it. There must also be space (paths through the facility) allowed for transporting work-in-process (WIP). The optimal location assignments are highly dependent on the machine positions and the routing options for each part as they dictate the possible flow paths and distances through the facility.

Facility Grid Model Representation

In our model, we have chosen to represent the facility floorspace in the form of an *a* by *b* rectangular grid of spaces we are calling a facility grid. The facility grid is a grid representation of possible locations where machine workstations can be set up. Each space in the grid is assumed to be sufficiently large enough to contain only one machine. Once a machine is assigned to a location on the grid, that position is its fixed position for the production period (no reconfiguration). To allow for access to a machine occupying a given space, we do not allow for any two machines to be in direct contact (all adjacent spaces to an occupied space must be empty). This allows for multiple flow paths to coexist as well as for freedom to approach each machine from any direction.

We represent the facility grid in matrix form. Let $G = [G_{ij}]$ be an a by b matrix that represents a grid of possible positions within the facility that a machine can be located (See Figure 3.8). Each location is assigned a number that identifies that location in the grid. This number is used in the encoding of the problem to indicate where the machine is assigned. For example, in Figure 3.8, $G_{33} = 9$. Meaning that location (3,3) on the grid is labeled location number 9. The labeling structure is used in place of cartesian coordinates to simplify the encoding of the problem by reducing the number of variables that need to be considered and with that, shortening the chromosome length.

G is used to establish a labeling structure for solving the problem. However, we need to establish a machine location assignment matrix, L. This allows us to see exactly where each machine is in the facility. Let $L = [L_{ij}]$ be an a by b matrix that contains information on where each specific machine is located in a grid of possible locations within the (See Figure 3.8). Such that $L_{11} = 2$ indicates that machine m_2 has been assigned to position (1,1) on the layout grid. Note, L_{11} corresponds with G_{11}.

One of the constraints is that each machine does not have any other machine adjacent to it. For the sake of simplicity, we created a matrix of occupied locations within the grid. Let $Y = [Y_{ij}]$ be an a by b matrix that indicates if a machine is located in a space of a grid of possible machine locations within the facility (See Figure 3.8). $Y_{ij} = 1$ indicates that a machine has been assigned to position (i,j) and $Y_{ij} = 0$ indicates the space is unoccupied.

Having established the locations of each machine, we can construct a distance matrix given that each spatial unit on the grid is a 1m^2 square. The distance is the number of squares it would take to move from one node to the next. Diagonal movements are not permitted. An example distance matrix for the grid shown in Figure 3.8 can be seen in Figure 3.9. Let $D = [D_{ij}]$ be an *nm* by *nm* matrix that depicts the distance between machines, where *nm* is the number of machines in the system. $D_{ij} = 1$ indicates that machine m_i is 1m away from machine m_j.

$$G = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \quad L = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 3 \end{bmatrix} \quad Y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Figure 3.8 Matrix Representations of Layout

$$D = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 3 \\ 1 & 3 & 0 \end{bmatrix}$$

Figure 3.9 Distance Matrix Example

3.2.4 Operators

Operators perform operations on the WIP using the machine. In our model, we do not explicitly model operators. However, we allow for operators to exist if needed as well as for the system to be fully autonomous. When operators are present, it is assumed that when a machine is in use (during set up and operation), that an operator is present. Also, in that scenario it is assumed that there are sufficient operators in the system such that there are no delays in work.

The operator notifies the system when they are free or busy and the system tells the operator what to do.

3.2.5 Transporters

Transporters are physical resources that move WIP from one location to another as required by the system. The transporter capabilities determine the transfer time. This is the time required to transport parts from one location to another within the facility. It is assumed that there are sufficient transporters to move all parts that require it between machine locations. These transporters are assumed to never encounter obstructions or delays in their path and will always take the optimal path between locations (transfer time is always the shortest time). Transporters, in this model, are assumed to have a fixed, constant, preassigned speed.

3.2.6 Buffer Space

Buffers are spaces where WIP are held when they have finished being served by a machine but not yet in assigned to a new machine. This happens when every machine that could be used to execute the next operation in a part's processing route is busy serving another part. They allow for the current machine to be freed to accept new work even when WIP cannot be moved to its next processing step. It is assumed that there is always sufficient buffer space to hold parts in the event that they must wait for the next available machine that can perform the next operation in their operation sequence.

3.2.7 Overview of Physical Domain of SMS Operation

In our model, a part order enters the system and immediately initiates a job that needs to be scheduled. The system determines which machine is assigned the first operation in this job based on the current system state. The system transports the raw materials or WIP to its assigned machine for the operation to be executed. Once the operation is completed on the machine, a new operation is requested of the system and the process repeats again. This continues until the job is complete. At this point, the completed part is transported out of the system. Figure 3.10 depicts an overview of the operation of the physical aspect of the SMS.

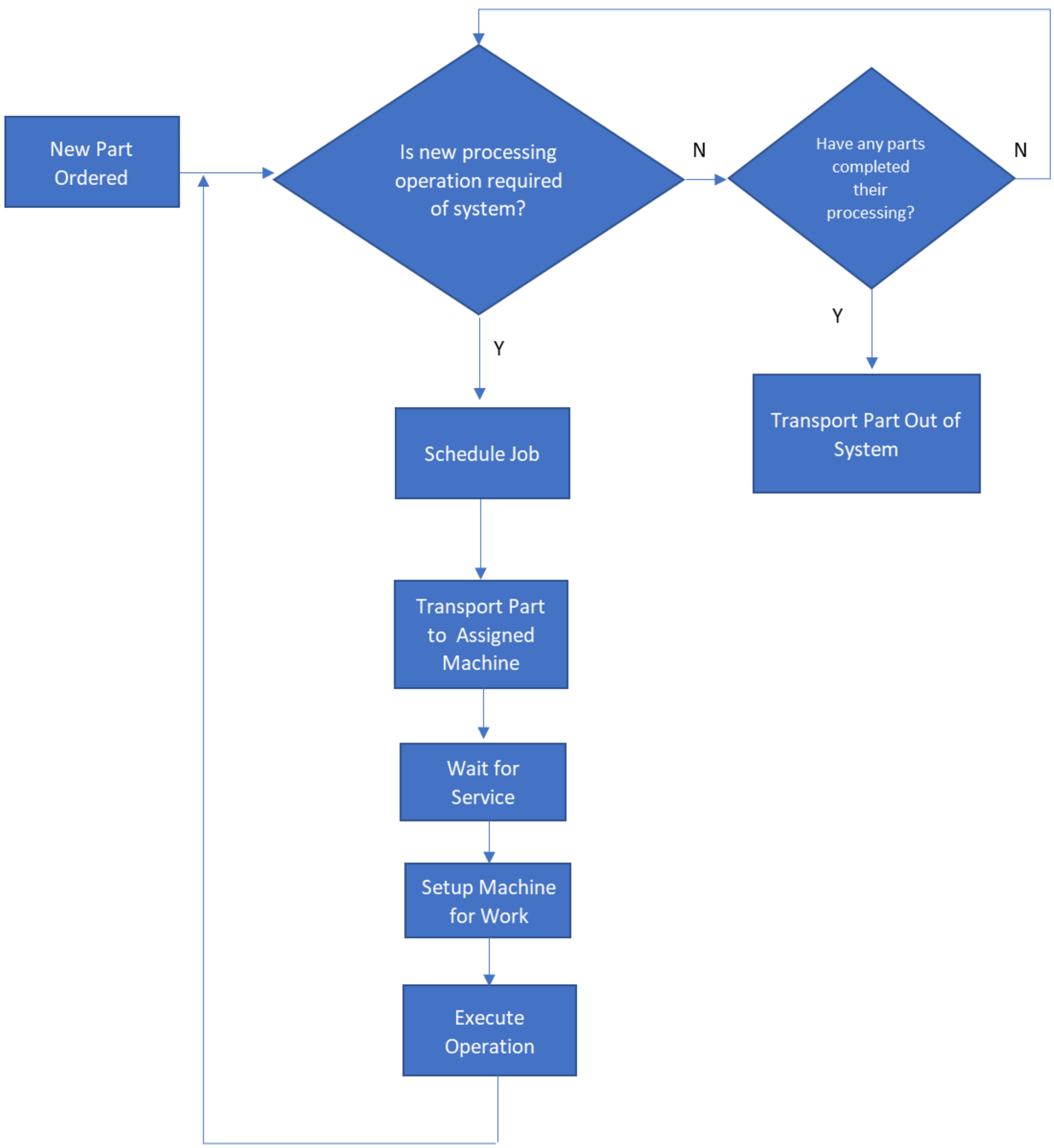


Figure 3.10 Operation of the Physical Aspect of the SMS

3.3 Agent Domain of the Smart Manufacturing System Model – Multi-Agent System

Determining how to schedule work in a mass customization manufacturing environment is difficult. This environment is subject to high levels of uncertainty which makes scheduling difficult. Cyber-physical production systems (CPPS's) provide a solution for scheduling in such an environment. With the SMS, there is a large volume of data being generated in real-time. The system has the infrastructure for collecting information (sensors) and communicating (an industrial network, transmitters, receivers, etc.) the real time status of the system is available. This status information includes the real-time status of the system's machines (idle or busy), the current location of all WIP and the current processing stage each WIP is at. The CPPS has the capacity to store and process this information is available (SaaS, IaaS, PaaS). This begs the question of how to effectively use this data and processing capability? We propose using a multi-agent system-based approach. The approach allows us to break the complex problem of scheduling in real-time into a series of smaller, less complex problems. The result is a less computationally taxing and more scalable solution.

The SMS we propose is a multi-agent system (MAS). This means that it consists of multiple autonomous agents which interact with each other whilst acting towards their individual goals. With MAS, there are three important things to establish. One is the rules that govern the interaction of the agents (part agents and machine agents). Particularly, what information is exchanged between the agents of the system. The second is the objective that guides the behaviour and decision-making of each agent within the system. Lastly, the control architecture for the system (the hierarchy of the agents within the system with respect to decision-making) must also be established.

In our SMS model, we have two major components; (1) parts and (2) machines. We have chosen to give both these components autonomy in the scheduling process. Due to literature suggesting the benefits of a hybrid control structure, we also incorporate a supervisory agent (Barbosa et al, 2015). As such, there are three types of agents that exist in our model: (1) part agents, (2) machine agents, and (3) the supervisory agent.

In this section, we will describe each agent type within the system, the control architecture of the system, and how the parts and machines interact with each other within the system. More specifically, we will discuss how decisions are made within the system.

3.3.1 Agent Descriptions

In our model, we have three (3) types of agents within the system. These are part agents (PAs), machine agents (MAs), and one supervisory agent (SA). This subsection will provide a description of these agents. More specifically, we will describe their function and objective within the system.

Parts Agent

The part agent (PA) is a representative for a specific part that was ordered from the system. Each part requested from the system is assigned its own respective part agent. This PA solely focuses on scheduling the production of the part it represents, and that part alone. It competes with the other part agents in the system for time at the system's machines.

Each part agent is like an auctioneer. It auctions operations required to produce the part that it represents to the systems machines. When the part it represents is not being processed at a machine, the PA has work that it needs done and so sets up an auction for this work for the machine agents to bid. The bids provided by the machine agents are the time required to complete the request the part agent made. This is the summation of the transfer time (*TT*) to machine from the part's current location, the setup times (*ST*) and processing times (*PT*) at a given machine. The PA selects the best available bid.

The part agent's objective is to minimize the flow time (*FT*) for the specific part, *p*, it represents. More simply put, its objective is to produce the part it represents as quickly as possible. This is done by selecting the best machine and operation combination at each processing step that advances the part to the next processing step the fastest. The objective function is as follows:

$$\min FT_P\,(M, L) \tag{3.1}$$

$$FT_P(M, L) = TT_P(M, L) + \sum_{i=1}^{m} \left(ST_{ip} + PT_{ip}\right) \tag{3.2}$$

Once a part agent is generated, it operates as follows:

1. If part is not being serviced by machine
    i. ***Request*** operation(s) from system
2. ***Receive*** information
    i. Transport time from each viable machine agent
    ii. Setup time from each viable machine agent
    iii. Processing time from each viable machine agent
3. ***Determine*** the combination of machine and operation that yields shortest time to next the operation
4. ***Create and Send*** ranked list of machine and operation combinations for use in event of conflict with other part agent
5. ***Assign*** work to the machine
6. ***Wait*** for confirmation of lack of conflict from supervisory agent
    i. if there is a conflict receive new machine and operation assignment from supervisory agent based on ranked list sent

7. ***Wait*** for work to be completed
8. ***Repeat*** process until there are no more operations
9. ***Delete*** part agent

Machine Agent

A machine agent (MA) is a virtual representation of a machine in the system. Each machine in the system has its own respective MA. This agent has the ability to make decisions on behalf of the machine it represents. The specific decision it makes is whether or not to look for new jobs for the machine it represents. It competes for these jobs with other MAs for similar machines.

The MA is aware of the capabilities of the machine it represents. This means the MA has the information on which operations its machine can perform. It also has information on the parts this machine can service as well as the processing and setup times associated with servicing those parts. It is also aware of the real-time status of the machine (whether the machine is idle or busy). The machine agents use this information to make decisions on which PA requests to bid on as well as to prepare estimates to submit as their bid.

The objective of the machine agent is to maximize the utilization of a specific machine (as shown in Equation 3.3). To do so, the machine agent sends bids on all available work if, and only if, the machine it represents is currently idle. The machine agent is like a contractor. It decides on whether or not to bid on work and then proceeds to prepare a bid for the auctioneer (the PA). This bid is an estimate of the amount of time required to complete the process requested by the PA. All MA's that are able to bid on a job do so. The MA that sends the best bid is awarded with that work. If two or more bids are equivalent, the work is randomly assigned to any of the winning MAs.

$$\max MU_m \tag{3.3}$$

The machine agent functions as follows;

1. ***Receive*** request to execute operation
2. ***Assess*** if machine is free or busy (if busy ignore request from part agent)
3. ***Estimate*** processing and setup time for requested operation
    i. Given distribution for the operation
4. ***Estimate*** transfer time
5. ***Send*** transfer, set up and processing time to part agent
6. ***Wait*** for operation to be assigned
7. ***Complete*** assigned operation
8. ***Repeat***

Supervisory Agent

There are bound to be job assignment conflicts with our MAS scheduling approach. Perhaps there is a bottleneck machine that all parts need processing on. In such situations, there needs to be a means to resolve these conflicts. The PAs cannot do this as they are not concerned with the global objective. They represent their part's best interest and that alone. There is no incentive for any agent to defer to another. That is why a supervisory agent (SA) is necessary. The SA acts as a referee in the auctioning process. It observes the interactions between machine and part agents and only ever intervenes when a machine wins multiple bids simultaneously. The supervisory agent intervenes by reassigning work in such a way as to best align with what is best for the system as a whole. In doing so, the SA ensures that the MAS scheduling solution tends towards the global objective of the system.

The SA's objective is to minimize the largest observed flow time (FT) for all parts involved in the scheduling conflict. The SA is responsible for determining the machine assignments for parts whose PA's assigned them to the same machine. This decision should result in shorter flow times overall for completely processing the total order of parts. This could involve determining which parts should be scheduled on an alternate machine. It could also involve determining which part should be given priority, and which should wait in buffer space. The objective function is as follows:

$$\min\left(\max\left(FT_1(M,L), FT_2(M,L), \dots FT_p(M,L)\right)\right) \tag{3.4}$$

The supervisory agent functions as follows;

1. ***Receive*** machine assignments
2. ***Receive*** ranked list of alternate assignments
3. ***Assess*** if intervention is needed. Check if there are conflicts in machine assignments (if none exist do nothing)
4. ***Reassign*** machine jobs based on ranked list such that global objective is fulfilled
5. ***Send*** new machine assignment information to machines and parts agents
6. ***Repeat***

3.3.2 Multi-Agent System Control Architecture

The MAS approach we present employs a hybrid control architecture (combining elements of heterarchical and hierarchical control) and is developed based on contract net protocol (Smith 1980) and the extension to contract net protocol presented by Wei et al. (2007). This approach has been shown to provide the best compromise between system performance of hierarchical control and the reduced sensitivity to stochastic disturbances exhibited by heterarchical control structures (Barbosa et al. 2015). In our model, PAs and MAs exist on the scheduling layer, and the SA exists on the supervisory layer. The hierarchical representation of the system can be seen in Figure 3.11.

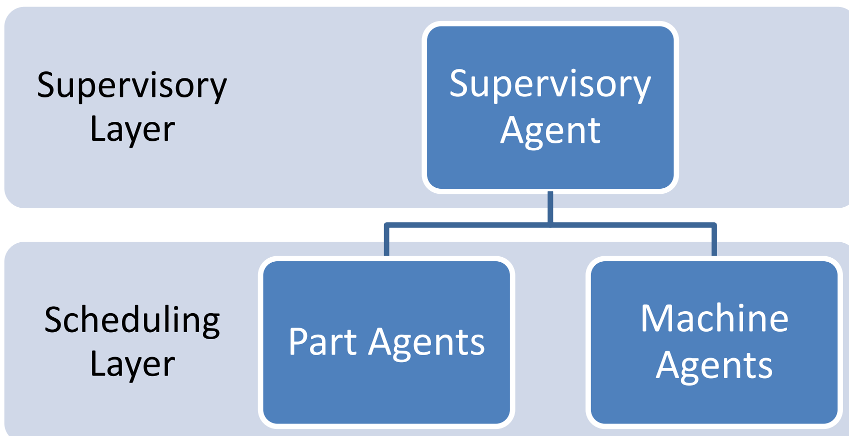


Figure 3.11 Hierarchical Representation of System's Agents

3.3.3 Overview of Agent Relationships Within the System

In our model, we assume that the infrastructure for collecting information (sensors) and communicating (an industrial network, transmitters, receivers, etc.) the real time status of the system is available. This status information includes the current status of the system's machines (idle or busy), the current location of all WIP and the current processing stage each WIP is at. We also assume that the capacity to store and process this information is available (Software-as-a-Service, Infrastructure-as-a-Service, Platform-as-a-Service). The question is, how should we use this information to schedule work? We are using a MAS for this scheduling. In the previous subsection, we have presented the three types of agents that exist in our model: (1) part agents, (2) machine agents, and (3) the supervisory agent. This subsection will focus on how these agents communicate with each other during their interaction. To that effect, an overview of each agent's functions, inputs and outputs can be seen in Figure 3.12. Figure 3.12 also shows the information flow between each agent.

With each part ordered from the system, a PA is generated. Before each processing step, each PA announces to all machines the work available to be done for their respective part it represents. They request bids from machines in the form of estimates of time required to get each respective part to the next processing step. The MAs review all requests for work from the PAs in the system. If any MA in the system can execute the operation, and is available to do so, it returns a bid. This bid consists of three pieces of information, the estimated transfer time, setup time and processing time for the specific part and operation combination. The PAs review their bids then select winners to assign work to. After which, each PA ranks the remaining machines based on their bids as potential alternates. All of this information is then communicated to the SA. If there is a conflict (i.e. two PAs awarding work to the same machine), the SA intervenes. It reviews the ranked list(s) of alternate machines provided by the PAs and then assigns work based on minimizing the maximum flowtime (*FT*) for all parts (*p*) currently in the system. If no MA bids on a PAs work request, then the PA must wait and re-announce the work. In the meantime, the part is held in storage until it can be processed. It is assumed that there will always be sufficient storage capacity for work-in-process (WIP) in the system. Similarly, if no work is available to bid on, the MAs simple wait idly for work to be requested. A comprehensive illustration of the behavior of and the interaction between the agents in the system is represented in the UML-Sequence diagram shown in Figure 3.13. These series of interactions continue until all parts requested of the system are produced and the production cycle ends.

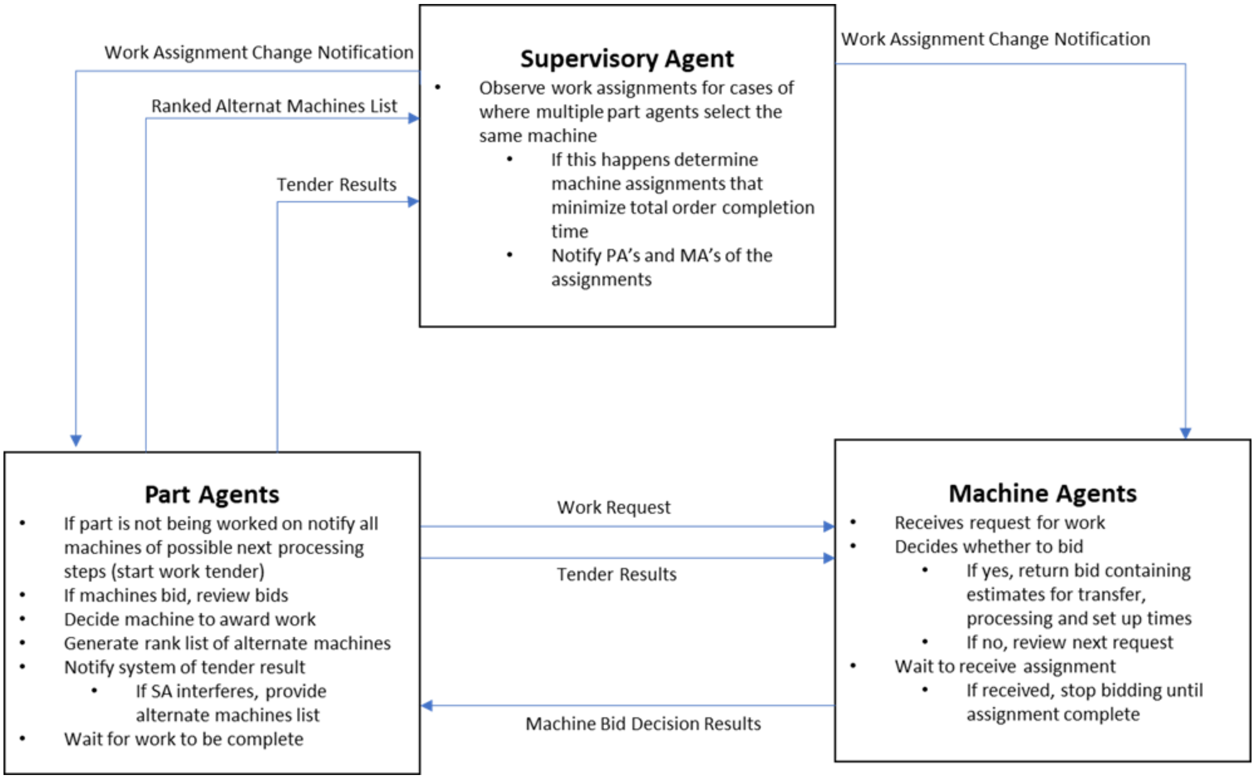


Figure 3.12 Information Flow Between Multi-Agent System Agents

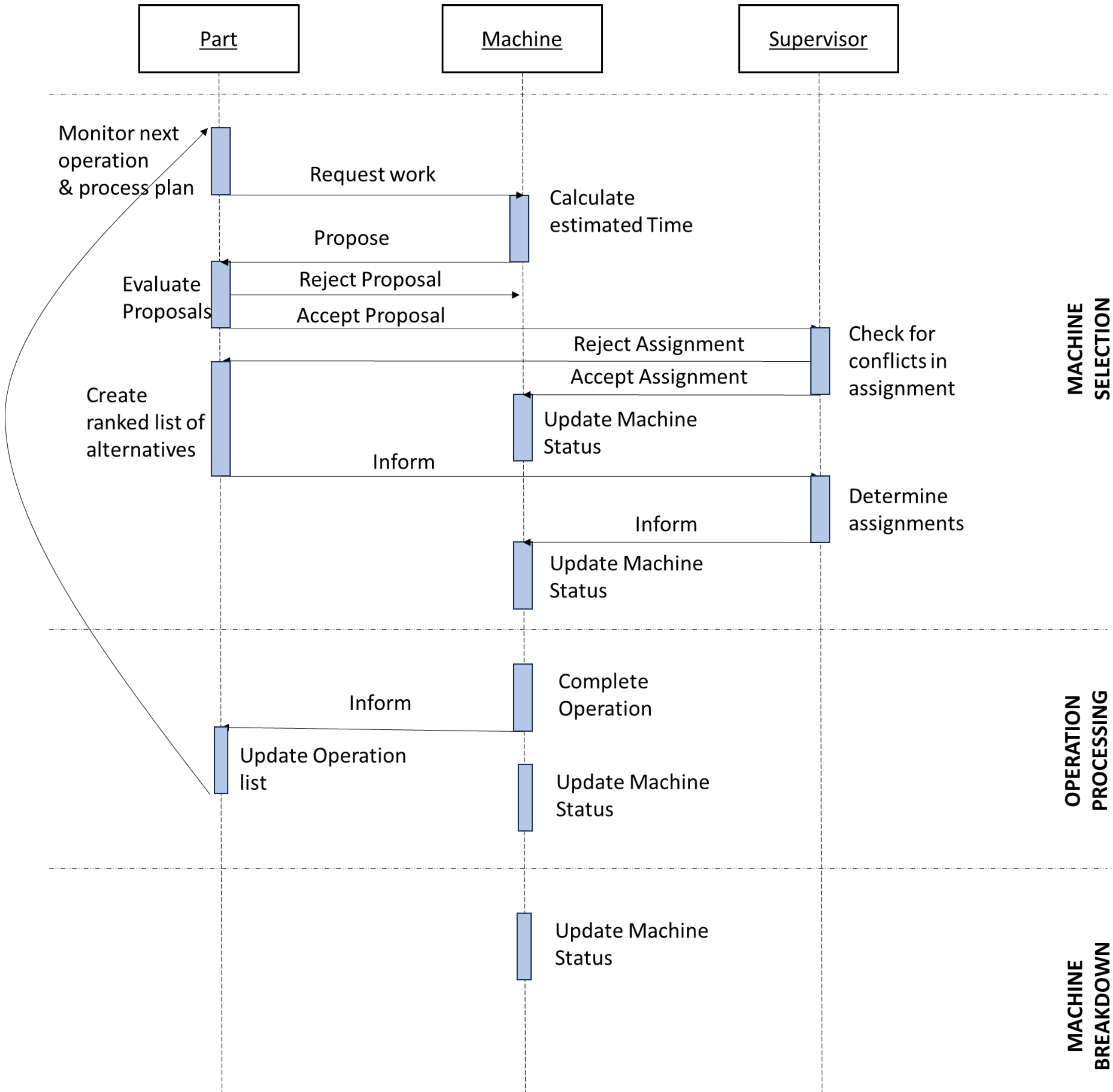


Figure 3.13 UML-Sequence Diagram for Multi-Agent System Model

3.3.4 Illustrative Example of MAS Agent Decision-Making Process

In this section, we present an illustrative example to demonstrate the agent decision-making process in our MAS model. For this example, we have a system consisting of eight (8) machines which can be used to produce two (2) different parts. Figure 3.14 depicts the layout of the system for this example. Each part has 2 possible machine routes that they can follow. The routing information for each part can be seen in Figure 3.15. For this example, we will demonstrate the decisions made when trying to produce part 2. We will assume that it is the only part in the system.

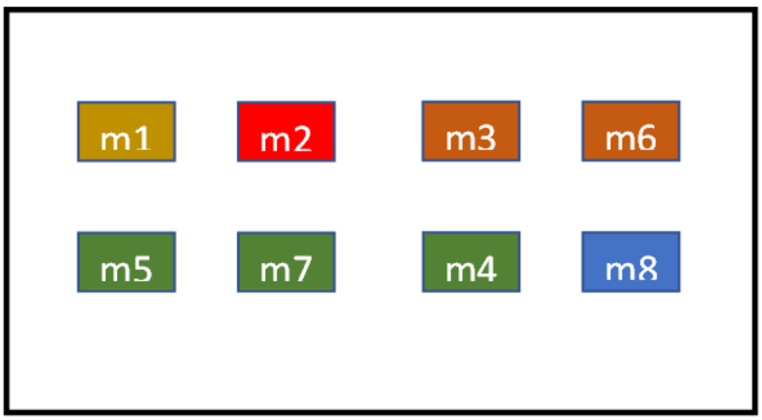


Figure 3.14 Example Layout Design

	O1		O2		O3		O4		O5		O6		O7		O8	
	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P
P1	μ: 1 σ: 0	μ: 4 σ: 0	μ: 1 σ: 0	μ: 4 σ: 0	μ: 2 σ: 0	μ: 7 σ: 0	μ: 1 σ: 0	μ: 5 σ: 0	μ: 1 σ: 0	μ: 2 σ: 0	μ: 1 σ: 0	μ: 3 σ: 0	μ: 1 σ: 0	μ: 3 σ: 0	inf	inf
P2	μ: 1 σ: 0	μ: 4 σ: 0	μ: 1 σ: 0	μ: 2 σ: 0	μ: 2 σ: 0	μ: 3 σ: 0	inf	inf	μ: 2 σ: 0	μ: 7 σ: 0	μ: 1 σ: 0	μ: 4 σ: 0	inf	inf	μ: 3 σ: 0	μ: 9 σ: 0

	S	O1	O2	O3	O4	O5	O6	O7	O8	E
S										
O1	1									
O2						1				
O3							1			
O4								1		
O5	1									
O6		1								
O7			1							
O8										
E				1	1					

Part 1

	S	O1	O2	O3	O4	O5	O6	O7	O8	E
S										
O1	1									
O2		1				1				
O3			1							
O4										
O5	1									
O6				1						
O7										
O8			1							
E							1		1	

Part 2

Figure 3.15 Part Routing Information (bottom) and Processing and Setup times (top)

MAS Agent Decision-Making for Production of Part 2:

The following steps represent the MAS decision-making process for the production of part p_2:

Step 1*:* Request bid from m_1 or m_5

Step 2: Wait for bids on work

- m_1 returns bid (job takes: 5 hrs; remaining flowtime: 13 hrs)
- m_5 returns bid (job takes: 9 hrs; remaining flowtime: 15 hrs)

Step 3: choose m_1 (see Figure 3.16)

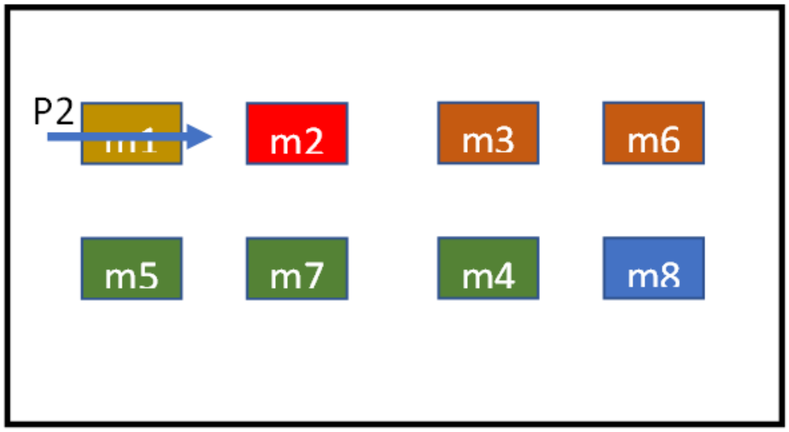


Figure 3.16 First Routing Decision

Step 4: Wait for work to complete on m_1

Step 5: Request work for m_2

- m_2 returns bid (job takes: 3hrs; remaining flowtime: 10 hrs)

Step 6: choose m_2 (see Figure 3.17)

Step 7: Wait for work to complete m_2

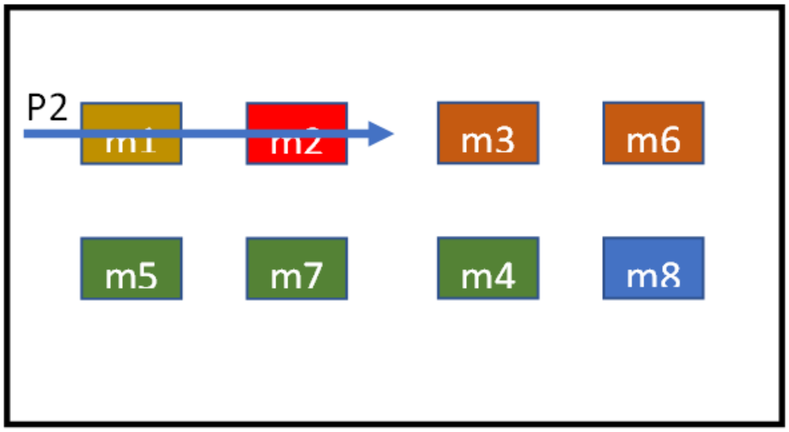


Figure 3.17 Second Routing Decision

Step 8: Request work from m_3 and m_6

- m_3 returns bid (job takes: 5hrs; remaining flowtime: 5 hrs)

- m_8 returns bid (job takes: 12hrs; remaining flowtime: 0 hrs)

Step 9: choose m_3 (see Figure 3.18)

Step 10: Wait for work to complete in m_3

Step 11: Request work for m_6

- m_6 returns bid (job takes: 5hrs; remaining flowtime: 0 hrs)

Step 12: Choose m_6 (see Figure 3.18)

Step 13: Wait for work to complete in m_6

Step 14: Done. End simulation

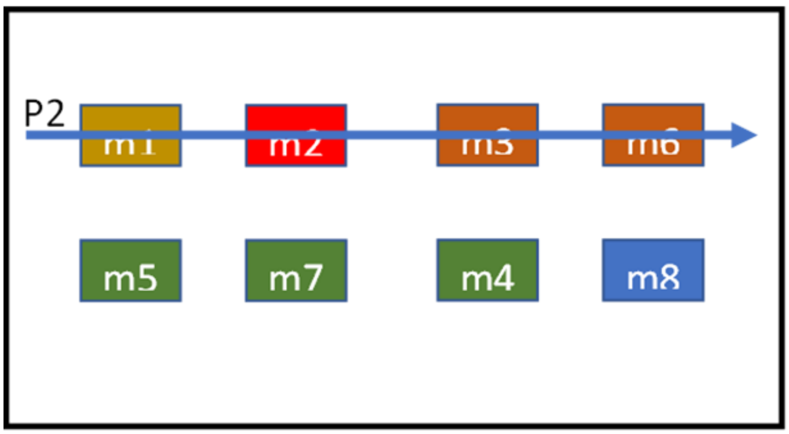


Figure 3.18 Final Routing Decision

3.4 System Properties and Performance Measures

There are a number of properties that can be used to evaluate the performance of different manufacturing systems. Some of these include the throughput, completion time, machine utilization, flowtimes, and wait times. In our model, we primarily focus on three measures; (1) completion time, (2) wait times, and (3) transfer times. In this section, we will describe these properties.

3.4.1 Order Completion Time

The order completion time is the time required to completely process all orders for parts placed to the system. Processing orders completely for a part requires converting the raw material to the finished good through a series of operations.

The order completion time is the key metric by which the performance of the SMS will be evaluated against the MS. This is because shorter order completion times indicates that the system is deploying its resources effectively to minimize instances of machines being underutilized, parts waiting unnecessarily in the system, or longer than required time in transit for the parts.

3.4.2 Wait Time

The wait time is the time WIP spends in the system waiting to be served by a machine. This does not include the time the WIP spends in transit. Higher part wait times indicate that the volume of parts entering into the system exceed the capacity for the system's machines to serve them or that the system's resources were being poorly utilized. Conversely, low wait times suggest that system's resources were sufficient to service parts in the system, or that the system's resources were being effectively utilized.

3.4.3 Transfer Time

Transfer time refers to the total time that a part spends being transported from one location in the facility to the other. Transfer times are dependent on two factors; (1) the machine route through which the part flows, and (2) the relative distance of machines in the system.

Transfer times provide an indication of how efficiently the machine locations within the facility are assigned whilst considering the processing network plans for the various parts that the system must produce. A shorter transfer time is better; however, it is not a direct indicator of the effectiveness of a system.

3.5 Model Verification

To verify that the presented model works as expected we have executed a number of verification experiments. In this section, we will describe the numerical experiments that were used for model verification, the intent of these experiments, and the results of these experiments.

3.5.1 Verification Scenario

As verification, we present a scenario in which we have twelve machines used in the production of one part (p_1). Part p_1 requires three operations to produce it, operations *C*, *D* and *E*. There are 4 sets of machines (m_5, m_6, m_7, m_8) that can perform operation *C*. There are 4 sets of machines (m_1, m_2, m_3, m_4) that can perform operation *D*. There are 4 sets of machines (m_9, m_{10}, m_{11}, m_{12}) that can perform operation *E*. Each machine that can perform the same operation is a duplicate of the other. The processing and setup times for each operation at their respective machine type can be seen in Table 3.1. At time $t = 0$, 20 units of part p_1 are requested from the system.

Table 3.1 Setup and Processing times For Part p_1 (in Time Units)

		Operation C	Operation D	Operation E
Minimum Time Required	Setup	0.009	0.009	0.009
	Processing	0.090	0.090	0.090
Average Time Required	Setup	0.010	0.010	0.010
	Processing	0.100	0.100	0.100
Maximum Time Required	Setup	0.011	0.011	0.011
	Processing	0.110	0.110	0.110

In this scenario we will examine the performance of two different layouts. These layouts are shown in Figures 3.19 and 3.20. The purpose of this scenario is to illustrate that distance has a significant effect on the completion time of an order. The primary difference between both layouts is the positions of machines m_1, m_2, m_3, and m_4. The difference in these machines' positions should result in an increase in the transfer times and order completion time. This is due to the increase in the distance that must be travelled with the repositioning of these machines. There should also be an increase in wait times seen in layout 2 as there will be longer queues due to part agents favoring specific machine routes due the shorter transfer times estimations. Transfer time is one of three deciding factors being used by the part agents in their decision-making (the others are set up and processing times).

	1	2	3	4	5	6	7	8	9	10	11	12	13
1													
2		m5						m7					
3				m1		m3							
4		m9						m10					
5				m2		m4							
6		m6						m8					
7													
8				m11		m12							
9													

Figure 3.19 Layout Type 1

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	m1												m3
2													
3				m5						m7			
4													
5				m9						m10			
6													
7				m6						m8			
8													
9	m2					m11		m12					m4

Figure 3.20 Layout Type 2

This scenario should demonstrate that the agent intelligence works as expected and also demonstrates that the output of the simulation is reflective of the logic used in its development (longer travel distances for the same machine route should result in having the same processing and setup times but different completion times). In our numerical experiment, we performed 10 simulation runs each using layouts 1 and 2. The results of the simulation run are shown in Table 3.2.

Table 3.2 Mean Results for Scenario Facility Performance Metrics in Time Units

	Completion Time	**Max. Observed Part Wait time**	**Part Wait time**	**Total Transfer Time**
Layout 1	5.70	3.60	1.61	1.65
Layout 2	10.71	6.95	3.15	2.94

The result of the numerical experiment serves as verification of the simulation model. As expected, when relative distances between machines in a machine route are longer, the completion time is increased due to the increased travel time. The wait times indicate the agent intelligence is functioning as expected. When the transfer time increases, there should be greater inclination for certain machines to work together. This means that when a part starts on a machine, it is more restricted in the machine options that it has for processing.

3.6 Model Comparison

In Table 3.3 we present our review of literature related to multi-agent systems for dynamic scheduling. From our review, most literature focuses on scheduling new arrivals and order cancellations. Works seldom examined alternate routing or operation sequences. The most common objective for the MAS was related to the makespan for the order. The MAS used in literature typically either utilized a hierarchical or heterarchical control architecture. Our model provides a unique contribution in that it uses hybrid control architecture as well as addresses not only multiple sources of uncertainty but also alternative routing and operation sequence options.

Table 3.3 Classification of Previous Works Related to MAS for Job Shop Scheduling

Source	Sources of Uncertainty Examined			Production Flexibility		Approach	
	Demand	Random Machine Breakdown	Processing Times	Alternative Routes	Alternative Operation Sequences	Objective Function	Control Architecture
Our Model	✓	✓	✓	✓	✓	Total Flow Time	Hybrid
Kang et al. (2007)	✓	✓	×	×	×	Makespan & Tardiness	Hierarchical
Madureira et al. (2007)	✓	×	✓	×	×	Makespan	Hierarchical
Wei et al. (2007)	✓	×	✓	✓	×	Several	Heterarchical
Xiang & Lee (2008)	✓	✓	✓	✓	×	Several	Heterarchical
Yu & Ram (2006)	✓	×	✓	×	×	Several	Heterarchical
Li et al. (2005)	✓	×	×	×	×	Makespan	Hybrid
Zhang & Wong (2017)	✓	✓	×	✓	✓	Makespan	Heterarchical
Rajabinsab & Masour (2011)	✓	✓	✓	✓	✓	Several	Heterarchical
Xiong & Fu (2018)	×	×	×	×	×	Makespan	Heterarchical
Hsu et al. (2016)	×	×	×	×	×	Makepsan	Heterarchical
Nie et al. (2022)	✓	✓	×	×	×	Several	Hybrid
Li et al. (2014)	×	×	×	×	×	Completion Time	Hierarchical
Li et al. (2022)	✓	×	✓	✓	×	Several	Hierarchical
Pal et al. (2022)	✓	×	×	✓	×	Completion Time	Hierarchical

Chapter 4

Investigation of the Multi-Agent System Based Manufacturing System Subject to Uncertainty

In any manufacturing environment, there are several uncertainties that can make scheduling difficult. For example, completing operations may take longer or shorter durations than estimated depending on the skill and fatigue level of the operator. The actual demand for parts may be different than the forecasted demand. Machines may break down during the production cycle and need to be repaired. Some combination of all these sources of uncertainty typically exist simultaneously in most manufacturing systems. As such, the choice of which dynamic scheduling strategy to employ is crucial. It is important to understand how different scheduling strategies perform when faced with different types of uncertainties.

In the previous chapter, we introduced a model for scheduling in a dynamic manufacturing environment. We proposed a using multi-agent system (MAS) to schedule jobs in real-time. It is a form of completely reactive dynamic scheduling. However, there are multiple approaches for dynamic scheduling that could have been explored as well. There is a need to investigate how the system we propose will perform when subject to different conditions as well as compare that performance to alternative solutions. There are two key objectives in this chapter. The first is to investigate the performance of our proposed MAS-based manufacturing system when subjected to different sources of uncertainty. We will investigate the performance of our system when used in a manufacturing environment that is subject to different, singular sources of uncertainty. We will then proceed to study the MAS system's performance when used in an environment subject to multiple sources of uncertainty simultaneously. Our intent is to study how the proposed MAS responds to varying levels of only one source of uncertainty before studying the performance when subject to multiple sources of uncertainty simultaneously. The other objective of this chapter is to compare the performance of our MAS against other dynamic scheduling approaches. Specifically, another completely reactive approach (using dispatching rules) and a predictive-reactive approach (right-shifting). We will be evaluating performance of these approaches based on the mean time for completing an order as well as the standard deviation in that completion time.

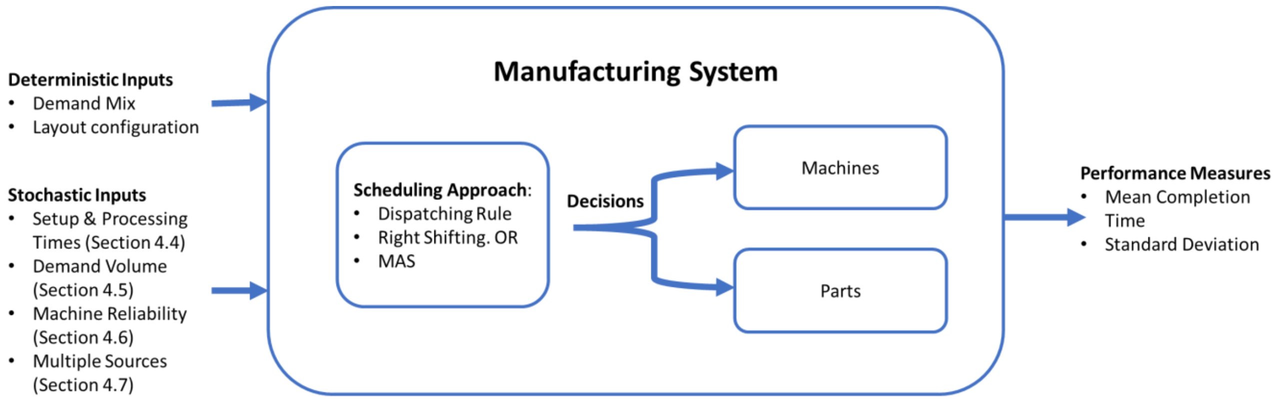


Figure 4.1 Overview of System Inputs and Outputs for Each Set of Numerical Experiments

This chapter will begin by introducing the case study problem that will be the basis of the simulation experiments conducted during this investigation. This case study provides the manufacturing system we will be simulating. The inputs used in the simulation model are separated into deterministic inputs (demand mix, layout configuration) and stochastic inputs (setup and processing times, demand volume, and machine reliability). Decision-making in the simulated manufacturing system is done using three different scheduling approaches for comparison purposes. From the simulation we get the completion time for the order. The manufacturing system's performance is evaluated based on the mean completion times and their associated standard deviation. This information is obtained from multiple simulation runs. An overview of the system as well as its inputs and outputs can be seen in Figure 4.1.

In our simulation experiments, we will explore multiple scenarios under which the manufacturing system will be operated. There will be five distinct input conditions we will be considering for the manufacturing environment. For the first set of experiments, we assume that the manufacturing environment is subject to no uncertainty. In the second set of experiments, we assume that the manufacturing environment is subject to uncertain processing and setup times. In the third set of experiments, we assume that the manufacturing environment is subject to uncertainty in demand volume. In the fourth set of experiments, we assume that the manufacturing environment is subject to random machine breakdown and repair. Finally, in the fifth set of experiments, we assume that the manufacturing environment is subject to all the afore-mentioned sources of uncertainty simultaneously. This chapter concludes with a summary of our findings from our investigation. This summary will provide details on the conditions in which the MAS we propose will perform best as well as our findings on how the system responds to the different input conditions studied.

4.1 Introduction to the Problem - Furniture Manufacturing Problem

In this study, the manufacturing system used in our problem is a modified version of a furniture manufacturing facility presented by Suzic et al. (2012). The facility we are examining consists of eleven (11) machines that are used in the production of sixteen (16) different parts. These 16 parts are used in the manufacture of five (5) products. These products are shelves, wardrobes, horizontal dressers, vertical dressers, and computer tables. The products are denoted as $PD = \{pd_1, pd_2, pd_3, pd_4, pd_5\}$ respectively. Similarly, $P = \{p_1, p_2, \ldots, p_{16}\}$ represents the 16 parts that the system can produce. Each product in *PD* is composed of a subset of parts from *P*. The total number of each part required to produce each product can be seen in Table 4.1.

Table 4.1 Composition of Furniture Parts

	Product pd_1	Product pd_2	Product pd_3	Product pd_4	Product pd_5
Part p_1	0	0	2	2	0
Part p_2	2	0	3	0	1
Part p_3	0	0	0	2	4
Part p_4	0	3	0	3	3
Part p_5	0	2	0	1	3
Part p_6	0	4	0	0	0
Part p_7	0	0	1	2	0
Part p_8	0	2	0	0	0
Part p_9	0	0	3	0	0
Part p_{10}	0	2	0	0	0
Part p_{11}	0	0	0	0	2
Part p_{12}	0	1	0	1	0
Part p_{13}	2	0	1	1	2
Part p_{14}	2	0	0	0	1
Part p_{15}	0	0	0	2	2
Part p_{16}	0	0	0	0	4

There are four different types of machines in this system; (1) cutting, (2) edging, (3) drilling, and (4) computer numerical control (CNC) machines. The system contains some duplicate machines (machines that have the same capabilities with the same processing and setup times) as well as machines that are capable of multiple operations (the CNC mills). In our study, we assume that each of the CNC machines is capable of performing drilling, cutting, edging and CNC-specific operations. The CNC can perform these operations with the same proficiency as the other machines that uniquely perform each function. The breakdown of each machine's operation capabilities can be seen in Table 4.2.

Table 4.2 Machine Types Within the System

Machine Type	Machines	Machine Operation(s)
Cutting Machine	m_1, m_2	Cutting
One-sided Edging Machine	m_3	One-sided edging
Two-sided Edging Machine	m_4, m_5, m_6	Two-sided edging
Drill Press	m_7, m_8, m_9	Drilling
CNC Multi-Purpose Machining Center	m_{10}, m_{11}	CNC milling, Drilling, Cutting, One-sided Edging, 2-sided Edging

We assume that the facility has been designed in a functional layout with the machines being separated into cutting, drilling, edging and CNC departments. Figure 4.2 depicts the allocation of the machines within the facility and Figure 4.3 depicts the relative machine distance matrix. Each machine is at least 1m away from the next machine. This allows transporters to move freely between all machines. It is assumed that the system always has transporters available and that the transporters within the system have fixed and constant speeds.

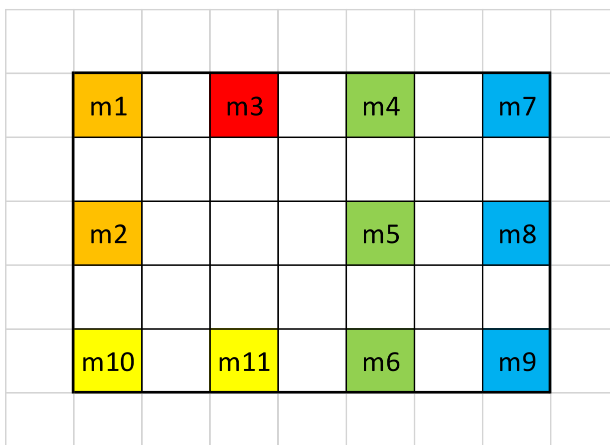


Figure 4.2 Furniture Manufacturing System Layout

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11
m1	0	2	2	6	6	8	8	8	10	6	6
m2	2	0	4	6	4	6	8	8	8	2	4
m3	2	4	0	2	4	6	6	6	8	6	4
m4	6	6	2	0	2	6	2	4	6	8	6
m5	6	4	4	2	0	2	4	2	4	6	4
m6	8	6	6	6	2	0	6	4	2	6	2
m7	8	8	6	2	4	6	0	2	6	10	8
m8	8	8	6	4	2	4	2	0	2	8	6
m9	10	8	8	6	4	2	6	2	0	8	6
m10	6	2	6	8	6	6	10	8	8	0	2
m11	6	4	4	6	4	2	8	6	6	2	0

Figure 4.3 Furniture Manufacturing Facility Relative Distance Matrix

This problem consists of eleven total machines but five distinct machine types. This means that certain machines have duplicates. In our study, we assume that all duplicate machines can perform equally. This means that all machine alternatives are equally capable of performing the same operation for the same set of parts. Processing and setup times for operations are part specific. The part-specific processing and setup times for each operation are assumed to be the same across all alternative machines for this problem. The part-specific set-up and processing times are provided in Table 4.3. For these experiments, the times are provided in time-units.

Table 4.3 Part Specific Set-up (ST) and Processing (PT) Times with Standard Deviations in Brackets (in time units)

	Cutting	1-sided Edging	2-sided Edging	Drilling	CNC
Part p_1	ST: 0.5 PT: 5	ST: 0.3 PT: 3			
Part p_2	ST: 0.4 PT: 4		ST: 0.4 PT: 4		
Part p_3		ST: 0.5 PT: 5	ST: 0.4 PT: 4	ST: 0.1 PT: 1	
Part p_4	ST: 0.4 PT: 4		ST: 0.3 PT: 3	ST: 0.1 PT: 1	
Part p_5			ST: 0.5 PT: 5	ST: 0.1 PT: 1	
Part p_6	ST: 0.4 PT: 4		ST: 0.3 PT: 3	ST: 0.1 PT: 1	
Part p_7	ST: 0.5 PT: 5	ST: 0.2 PT: 2	ST: 0.4 PT: 4	ST: 0.1 PT: 1	
Part p_8	ST: 0.4 PT: 4		ST: 0.5 PT: 5	ST: 0.1 PT: 1	
Part p_9			ST: 0.4 PT: 4		ST: 0.5 PT: 5
Part p_{10}	ST: 0.4 PT: 4		ST: 0.3 PT: 3		ST: 0.5 PT: 5
Part p_{11}		ST: 0.6 PT: 6	ST: 0.6 PT: 6	ST: 0.1 PT: 1	
Part p_{12}			ST: 0.4 PT: 4		ST: 0.5 PT: 5
Part p_{13}			ST: 0.4 PT: 4	ST: 0.1 PT: 1	ST: 0.5 PT: 5
Part p_{14}	ST: 0.4 PT: 4		ST: 0.5 PT: 5	ST: 0.1 PT: 1	ST: 0.5 PT: 5
Part p_{15}			ST: 0.3 PT: 3	ST: 0.1 PT: 1	ST: 0.5 PT: 5
Part p_{16}	ST: 0.3 PT: 3		ST: 0.3 PT: 3		ST: 0.5 PT: 5

Table 4.4 Part Operation Sequences and Machine Route Options

	Operation sequence	Machine Routing Options
Part p_1	cutting → 1-side edging	$(m_1, m_2, m_{10}, m_{11}) \rightarrow (m_3, m_{10}, m_{11})$
Part p_2	cutting → 2-side edging	$(m_1, m_2, m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11})$
Part p_3	1-side edging → drilling → 2-side edging	$(m_3, m_{10}, m_{11}) \rightarrow (m_7, m_8, m_9, m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11})$
Part p_4	cutting → 2-side edging → drilling	$(m_1, m_2, m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11}) \rightarrow (m_7, m_8, m_9, m_{10}, m_{11})$
Part p_5	2-side edging → drilling	$(m_4, m_5, m_6, m_{10}, m_{11}) \rightarrow (m_7, m_8, m_9, m_{10}, m_{11})$
Part p_6	cutting → 2-side edging → drilling	$(m_1, m_2, m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11}) \rightarrow (m_7, m_8, m_9, m_{10}, m_{11})$
Part p_7	cutting → 1-side edging → drilling → 2-side edging	$(m_1, m_2, m_{10}, m_{11}) \rightarrow (m_3, m_{10}, m_{11}) \rightarrow (m_7, m_8, m_9, m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11})$
Part p_8	cutting → 2-side edging → drilling	$(m_1, m_2, m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11}) \rightarrow (m_7, m_8, m_9, m_{10}, m_{11})$
Part p_9	CNC → 2-side edging	$(m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11})$
Part p_{10}	cutting → 2-side edging → CNC	$(m_1, m_2, m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11}) \rightarrow (m_{10}, m_{11})$
Part p_{11}	1-side edging → drilling → 2-side edging	$(m_3, m_{10}, m_{11}) \rightarrow (m_7, m_8, m_9, m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11})$
Part p_{12}	CNC → 2-side edging	$(m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11})$
Part p_{13}	drilling → 2-side edging → CNC	$(m_7, m_8, m_9, m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11}) \rightarrow (m_{10}, m_{11})$
Part p_{14}	cutting → drilling → 2-side edging → CNC	$(m_1, m_2, m_{10}, m_{11}) \rightarrow (m_7, m_8, m_9, m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11}) \rightarrow (m_{10}, m_{11})$
Part p_{15}	drilling → 2-side edging → CNC	$(m_7, m_8, m_9, m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11}) \rightarrow (m_{10}, m_{11})$
Part p_{16}	cutting → 2-side edging → CNC	$(m_1, m_2, m_{10}, m_{11}) \rightarrow (m_4, m_5, m_6, m_{10}, m_{11}) \rightarrow (m_{10}, m_{11})$

Each part required of the system has a distinct set of operations (in specific operation sequences) that must be performed to produce it. Each operation in the sequence can be performed by at least one machine in the system. This means that for a part to be produced, it must follow a route of machines through the facility. However, as there are duplicate and similar machines in the system, there are also multiple machine routings that can be used in their production. Any of these routing options can be used in the production of parts they can service (unless the scheduling approach used imposes restrictions). Table 4.4 depicts the operations sequences for each part as well as the possible machine routings that a possible within the facility.

4.1.1 Simulation Experiment Demand Scenarios

In our study, we will primarily be examining four (4) scenarios. Each of these scenarios involve orders with their own distinct demand mixes for parts in the system. The demand mix for orders requested from the system for each scenario is shown in Table 4.5 below:

Table 4.5 Demand for Each Part for Each Scenario

Scenario	Demand Mix
1	$\{p_1, p_2, p_3, p_4, p_5, p_6\}$
2	$\{p_1, p_3, p_5, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}\}$
3	$\{p_1, p_2, p_5, p_7, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}\}$
4	$\{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}\}$

Table 4.6 Machine Utilization for each Machine in the System for each Scenario

Scenario	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{12}	Mean
1	0.40	0.40	0.80	0.80	0.30	0.50	0.20	0.10	0.10	0.80	0.50	0.44
2	0.33	0.20	0.53	0.53	0.60	0.53	0.27	0.07	0.00	0.93	1.00	0.45
3	0.47	0.26	0.58	0.74	0.42	0.84	0.16	0.11	0.05	1.00	0.74	0.50
4	0.54	0.50	0.67	0.92	0.88	0.71	0.21	0.17	0.04	1.00	1.00	0.60

As previously mentioned, we will be focusing on four distinct scenarios. The scenarios represent different levels of complexity in the problem. Each scenario has its own mean machine utilization (see Table 4.6). With each subsequent scenario, the mean utilization of the system increases with the changing demand mix. The different demand mixes result in the different operation assignments for each machine "optimal" schedule. As such, each subsequent scenario represents a more complex problem than the previous. This is because there is more competition for the system's resources between the requested parts. In exploring these scenarios, we intend gather more insight on the performance of the dynamic scheduling strategies as the interrelationships between the parts and machines change, and more of the system's capacity is utilized.

4.2 Dynamic Scheduling Approach Benchmarks

In the upcoming sections of this chapter, we will present five distinct sets of experiments relating to scheduling under varying levels of uncertainty from different sources. Each experiment will involve a comparison of the MAS model that we proposed in chapter 3 against a predictive-reactive scheduling approach (right-shifting), and a completely reactive scheduling approach using dispatching rules (longest remaining processing time). This subsection will provide information on how these alternative dynamic scheduling approaches are implemented.

4.2.1 Dynamics Scheduling Using Right Shifting Rescheduling

Right-shifting begins with determining a nominal schedule which would be optimal assuming there are no disturbances to the system. This is the schedule which will be followed until a disturbance forces a deviation from it. In the event of a disturbance to the system, all jobs that are directly affected by the disturbance are moved downstream in the schedule to a point when the disturbance has been resolved. For example, if a machine breaks down during a production cycle and part p_a requires work that needs to be scheduled on that machine, that work is postponed until the machine is repaired and operational again. Also, any subsequent work for part p_a is delayed as well. All other unaffected jobs follow the nominal schedule.

Right-shifting requires that we determine a nominal schedule to follow. To determine the nominal schedule, we use a genetic algorithm (GA) based on the model developed by Zhang et al. (2011) for solving the flexible job-shop scheduling problem. This model is used as it allows for alternate machines and processing routes. Primarily, we will be using the structure for encoding the chromosome presented by Zhang et al. (2011). Figure 4.4 depicts a sample of how the chromosome is encoded. The chromosome is separated into two halves, machine selection and operation sequence. Each gene in the machine selection half of the chromosome corresponds to a gene on the operation sequence half. For example, Figure 4.4 indicates that the first operation for part p_2 is performed on machine m_4. Similarly, the second operation of part p_2 is performed on machine m_1, and so on. The order of the operation sequence half of the chromosome represents the precedence or priority of the operations. This comes into consideration determining the schedule for all parts scheduled for processing on the same machine. For example, parts p_1 and p_3 are both processed first at machine m_2. However, since part p_1 is first in the operation sequence, it would be scheduled first for the machine. The GA parameters used for solving our JSP are presented in Table 4.7.

	Machine Selection					Operation Sequence				
Chromosome	4	1	2	2	3	2	2	1	3	2

Figure 4.4 Chromosome Encoding for Flexible JSP

Table 4.7 GA Parametric Settings

Operator	Setting
Crossover Probability	0.55
Crossover Operator (Machine Selection)	Uniform Crossover
Crossover Operator (Operation Sequence)	Preserving Order-based Crossover
Mutation Probability	0.20
Mutation Operator (Machine Selection)	Random Resetting
Mutation Operator (Operation Sequence)	Random Resetting
Population Size	$10 \times n$
Maximum Number of Iterations	100
Termination Condition	ΔFitness < 0.01
Selection Process	Tournament Selection

The objective function used to evaluate the schedule solutions was the minimization of the completion time (*CT*) for a given order of parts, *s*. The problem formulation is as follows:

$$\min_{X} CT_s(X) \tag{4.1}$$

where

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1j} \\ \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{ij} \end{bmatrix} \tag{4.2}$$

s.t.

$$\sum_{j=1}^{np} y_{ij} = no_i \qquad \forall i = 1,2,\dots,np \tag{4.3}$$

Let $X = [x_{ij}]$ represent the order in sequence operation required for part p_j falls in the schedule for machine m_i. For example, in Equation 4.4, $x_{11} = 2$, this indicates that part p_1 is processed second on machine m_1. Similarly, $x_{12} = 0$ indicates that part p_2 is not processed on machine m_1. As such, X represents the overall schedule for the job-shop.

$$X = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 3 & 2 & 1 \end{bmatrix} \tag{4.4}$$

The only constraint this optimization problem is subject to is that all of the operations required to produce the demanded parts must be scheduled on a machine. We use Y = [y_{ij}] as a binary representation of parts that a machine is scheduled to process. It indicates that a part is processed at a machine without any indication of its order in the sequence. As such, the sum of the elements in the column represents the total number of operations required to produce the specific part (no_i). For example, Equation 4.5 indicates the total number of operations required for part p_1 (represented by no_1) is 3.

$$Y = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{4.5}$$

Nominal Schedules Determined Using GA

For each demand scenario presented in section 4.2.1, a nominal schedule was determined. These nominal schedules are to be used when employing right-shifting as the scheduling approach. The schedules for scenarios 1-4 can be seen in Figures 4.5 to 4.8. With each schedule depicted in Figures 4.5 to 4.8 we see the parts that have been assigned to each machine, as well as the time in which each part is scheduled to arrive and exit each respective machine. For example, in Figure 4.4, we can see that part p_2 is assigned to machine m_1 that at time T = 0. It is processed at this machine and exits at time T = 5.5. Part p_2 then is scheduled to be processed on machine m_4. Overall, the schedule presented in Figure 4.5 should be completed in 11 time units.

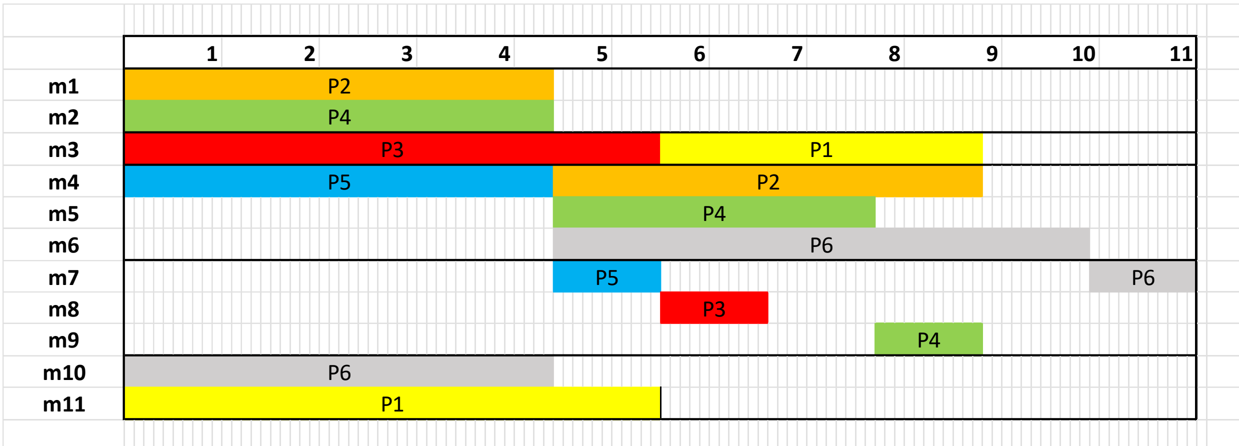


Figure 4.5 GA Nominal Schedule for Scenario 1

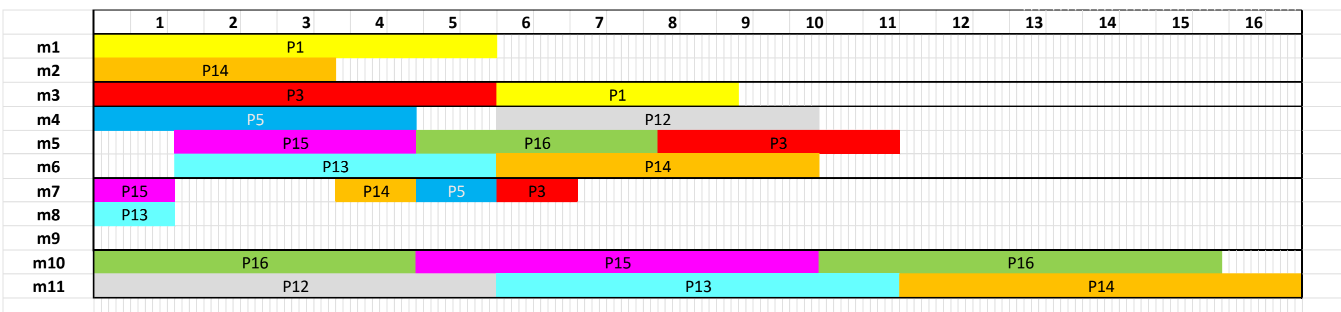


Figure 4.6 GA Nominal Schedule for Scenario 2

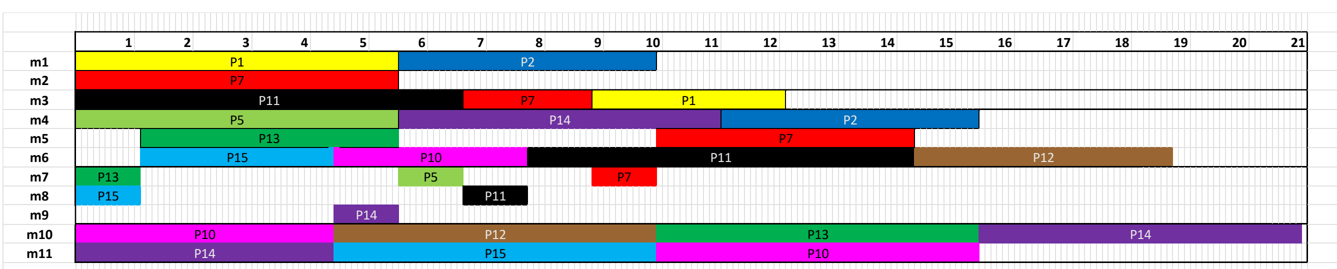


Figure 4.7 GA Nominal Schedule for Scenario 3

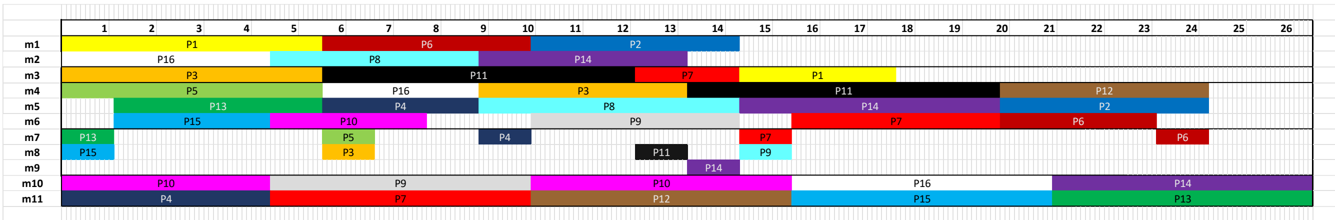


Figure 4.8 GA Nominal Schedule for Scenario 4

4.2.2 Dynamic Scheduling Using Dispatching Rules (Largest Remaining Processing Time)

Our proposed MAS model for dynamic scheduling will also be compared against a completely reactive dynamic scheduling approach using dispatching priority rules. The rule we have chosen is largest remaining processing time + shortest processing time (LRPT). This rule prioritizes jobs with the longest remaining time left along their given processing route. We have chosen LRPT as the benchmark, as it has been shown to have good performance in minimizing the most performance measures typically evaluated in job shop scheduling (Dominic et al., 2004). The algorithm used in the implementation of this dispatching rule is as follows:

1) For machine m_1
   a. if machine m_1 is available to service new part (currently not busy or in need of repair)
      i. determine parts that currently need to be serviced by this machine ($P_{m1} \in P$)
      ii. If no parts currently need to be serviced by this machine
         1. Continue to next machine
      iii. Else if there are parts that need service on this machine
         1. for each of part that is an element P_{m1}

a. determine remaining processing time for the part

iv. schedule part with longest remaining processing time on machine m_1

1. select the operation with the shortest processing time, if two or more tasks have the same remaining processing time

b. if machine m_1 is unavailable, continue to next machine

2) Repeat step 1 for machines m_2 to m_{nm}

3) Repeat steps 1 and 2 until no parts need to be serviced on any machine

4) End

4.3 Scheduling Under Fixed Product Demand with Fixed Processing and Set-up Times

In this section we will present a comparison of MAS-based scheduling approach to both dispatching rules and using near-optimal schedule determined *a* priori. For this set of experiments, it is assumed that there is no uncertainty, and all system inputs are purely deterministic.

4.3.1 Numerical Experiments

We compared the optimality of our MAS scheduling approach to the nominal schedule designed to have the minimum total completion time as well as against scheduling using LRPT as its dispatching priority rule. For these experiments, RS is not used as there would be no need to right-shift the schedule if there is no uncertainty.

Our objective was to determine if the proposed approach yielded a schedule that is comparable in terms of completion times when alternate scheduling approaches are employed. In this comparison, the optimality of two schedules is considered to be the same if there is no statistically significant difference between their means at the 95% confidence interval. To that effect, we conducted an analysis of variance (ANOVA) for the simulation experiments.

Experiment Conditions

The numerical experiments were only run for the MAS-based and LRPT rule-based approach to real-time scheduling. The simulations are run using a script developed in MATLAB r2021a. The conditions for the experiments are as follows:

1) A set of experiments is run for each of the four distinct demand mix scenarios (outlined in Table 4.5)
2) All variables in the experiment are fixed. Completion times for each operation are assumed to the mean time provided (i.e. $\sigma = 0$)
3) For each experiment run, the order for each demanded part enters the system at time zero ($T = 0$)
4) There were 20 replications for each simulation experiment conducted

4.3.2 Results

Table 4.8 below contains the results of the numerical experiments conducted for the four different scenarios presented in section 4.1.1. This table contains the mean completion time for processing a given order of parts as well as the associated standard deviation using three distinct scheduling approaches. We also present the P-values for hypothesis testing we conducted on the results of the experiments.

Table 4.8 Simulation Results for Performance Comparison between Nominal Schedule and MAS

Scenario	Nominal Schedule		LRPT Scheduling Approach		MAS Scheduling Approach		P-Value GA vs MAS	P-Value LRPT vs MAS
	Mean (μ_1)	STD (σ_1)	Mean (μ_2)	STD (σ_2)	Mean (μ_3)	STD (σ_3)	(H_0: $\mu_1 = \mu_3$)	(H_0: $\mu_2 = \mu_3$)
1	11.00	n/a	11.01	0.02	11.03	0.02	<0.001	<0.001
2	16.50	n/a	17.61	0.01	18.62	1.34	<0.001	<0.001
3	20.90	n/a	25.31	0.01	23.25	1.75	<0.001	<0.001
4	26.40	n/a	34.14	1.41	28.58	1.24	<0.001	<0.001

4.3.3 Discussion

The results of the experiments show that in very simple problems (such as in scenario 1) where there are sufficient machine resources to process all the parts simultaneously, there is little difference in the performance of the three scheduling approaches. In scenario 1, we see that all three approaches yield solutions that are equivalent (11.00, 11.01, and 11.03 time units when using the nominal schedule, LRPT and MAS respectively). With the MAS approach, we see that there is little variation in the results. This is to be expected as this scenario allows for each part to be processed in parallel given the machines in the system (As we can see in Figure 4.5). The processing routes of each part requested of the system do not overlap. As such, there is no need for a part to wait before moving to its next processing step as the machine it requires will be available. The results suggest that the MAS solution quality will be comparable to that of the near-optimal in situations where there is excess machine capacity to allow for parallel part processing.

As the problem complexity increases and parts can no longer be processed in parallel, we see the MAS approach does not yield the optimal result. It yields solutions that result in mean completion times within 8-12% of the optimal completion time. In scenarios 2, 3, and 4, we see more demand for the CNC and 2-sided edging machines than in scenario 1. This results in a situation where there is a need for some parts to wait before their next processing step. As such, the sequence of parts scheduled for a machine becomes more critical. For these set of experiments, we see that the MAS approach yields completion times approximately 8%-13% greater than the times expected using the nominal schedule. The variation in the results from the MAS approach is greater than that from the previous scenario.

Interestingly, when comparing the MAS approach to the LRPT-based, we see that the LRPT approach outperforms the MAS approach when there is lower demand for parts required of the system. However, as the demand for parts increases, we see the MAS approach begins to outperform the LRPT-based approach. In scenario 2, we see that using LRPT results in a 5% reduction in completion time when compared to the MAS approach. However, as we continue to scenarios 3 and 4 the MAS approach outperforms the LRPT approach by approximately 8%-10%. This observation is to be expected as there are fewer parts with longer processing times in the earlier scenarios. As such, by starting with them, we can better nest the processing times for the quicker to process parts in the time required to process the longer to process parts. This is because the system has enough capacity to allow for parts with longer processing times to be processed in parallel with parts with shorter processing times. However, as the number of parts demanded increases without an increase in the machine resources, we see the flaw in the LRPT approach. With the increase in the demand, there are more parts with long processing times. This takes up the machine resources that were free with the smaller, less complex problems. As such, parts with longer processing times can no longer be processed in parallel with parts with shorter processing times. Now, we have the longer processing time parts all being processed first before beginning the shorter processing time parts when using LRPT-based approach.

Overall, we can conclude that without any source of uncertainty, the MAS based approach will perform less optimally than the nominal schedule. However, MAS can outperform scheduling using a rule of thumb approach as with dispatching rules (LRPT). MAS outperforms dispatching rules as the volume of different parts demanded increases. It is also important to note that whilst the MAS approach is outperformed by the GA-derived, nominal schedule, it manages to remain within 8-13% of the lowest estimated completion times that we were able to solve for. It was expected that the MAS approach would underperform in comparison to the optimal in this set experiments. This is because the MAS is designed to act in real-time without considering downstream implications of each scheduling decision. It is analogous to a greedy algorithm. Without any source of uncertainty, there is no drawback to solving for the near optimal schedule, as there would be no need to reschedule or deviate from it.

4.4 Scheduling Under Fixed Product Demand with Uncertain Operation Execution Durations

In the typical job shop scheduling problem, we are presented with a set of n jobs to be completed on a set of m machines. It is assumed that there is a fixed demand for each part being requested from the system and this demand is known *a priori*. The setup and processing times are fixed and deterministic. However, this is not representative of real-life setup and processing times. Setup and processing times are uncertain, they can vary every time a task is repeated even when completed by the same operator. As such, it is important to understand how introducing a degree of uncertainty to these times would impact the system performance when using different scheduling strategies. In this section of our study, we compare the performance of our proposed MAS system to two other dynamic scheduling strategies using four scenarios involving different demand volumes for each part the system can produce. We will be referring to this source of uncertainty (variable setup and processing times) as operation execution duration (OED) uncertainty.

In this section, we will present a description of the numerical experiments performed as well as the results of the experiments and the associated analysis.

4.4.1 Numerical Experiments

With these experiments, we will use the four (4) scenarios outlined in subsection 4.2. The experiment inputs are listed in the next section.

Experiment Conditions

The numerical experiments were run using our own in-house simulation code written on MATLAB r2021b. The same set of experiments were conducted using each dynamic scheduling approach. We ran experiments for all four (4) distinct demand mix scenarios presented in section 4.1. In our design for our experiments, we assume that each of the setup and processing times for operations executed on any machine in the system follow a normal distribution. The times provided in Table 4.3 are used as mean values. For the standard deviation, we consider three level settings. These are (1) 10% of the mean time, (2) 20% of the mean time, and (3) 30% of the mean time. We ran a full factorial (3^1) experiment with 20 repetitions for each of the four scenarios. A total of 240 experiments were run.

4.4.2 Results

Table 4.9 below contains the results of the numerical experiments conducted for the four different scenarios presented in section 4.2. The table contains the mean completion times for processing a given order of parts as well as their associated standard deviations using three distinct scheduling approaches right-shifting (RS), dispatching rules (LRPT), and our MAS approach. We also present the P-values for hypothesis testing we conducted on the results of the experiments as well as the main effects of the uncertainty in OED. Tables 4.10 and 4.11 contain the main effects for the varying OED on the mean completion time and standard deviation respectively. This information is also depicted in Figure 4.9.

Table 4.9 Performance of MAS Scheduling in Comparison to Conventional Dynamic Scheduling Approaches (for 20 Repetitions)

Scenario	Uncertainty Level	RS Schedule		LRPT-based Scheduling		MAS Scheduling Approach		P-Value	P-Value
		Mean (μ_1)	STD (σ_1)	Mean (μ_2)	STD (σ_2)	Mean (μ_3)	STD (σ_3)	(H_0: $\mu_1 = \mu_3$)	(H_0: $\mu_2 = \mu_3$)
1	10%	11.84	0.76	10.89	0.56	10.95	0.47	<0.001	0.71
	20%	12.02	0.96	11.35	0.99	11.22	0.95	0.03	0.67
	30%	12.43	1.38	12.30	1.66	12.30	1.17	0.75	1.00
2	10%	16.66	0.70	18.72	1.37	19.80	1.53	<0.001	0.02
	20%	17.76	1.19	19.37	1.56	20.47	1.81	<0.001	0.04
	30%	18.79	1.92	19.09	2.28	20.48	1.87	0.62	0.04
3	10%	22.05	1.09	23.80	0.93	23.40	2.01	<0.001	0.42
	20%	21.85	1.28	23.91	2.02	23.04	1.68	<0.001	0.14
	30%	23.32	2.29	24.62	2.77	23.41	1.88	0.05	0.11
4	10%	28.29	1.15	34.61	1.73	29.37	1.57	<0.001	<0.001
	20%	28.77	1.73	34.61	2.11	30.61	2.99	<0.001	<0.001
	30%	30.86	2.76	34.48	2.33	30.72	2.21	<0.001	<0.001

Table 4.10 Main Effects of Uncertainty in Operation Execution Duration on Mean Completion Time Using Each Solution Approach

	10%	20%	30%	EFFECT
RS	19.71	20.10	21.35	1.64
LRPT	22.01	22.31	22.62	0.61
MAS	20.88	21.34	21.73	0.85

Table 4.11 Main Effects of Uncertainty in Operation Execution Duration on Standard Deviation of the Mean Completion Time Using Each Solution Approach

	10%	20%	30%	EFFECT
RS	0.93	1.29	2.09	1.16
LRPT	1.15	1.67	2.26	1.11
MAS	1.40	1.86	1.78	0.38

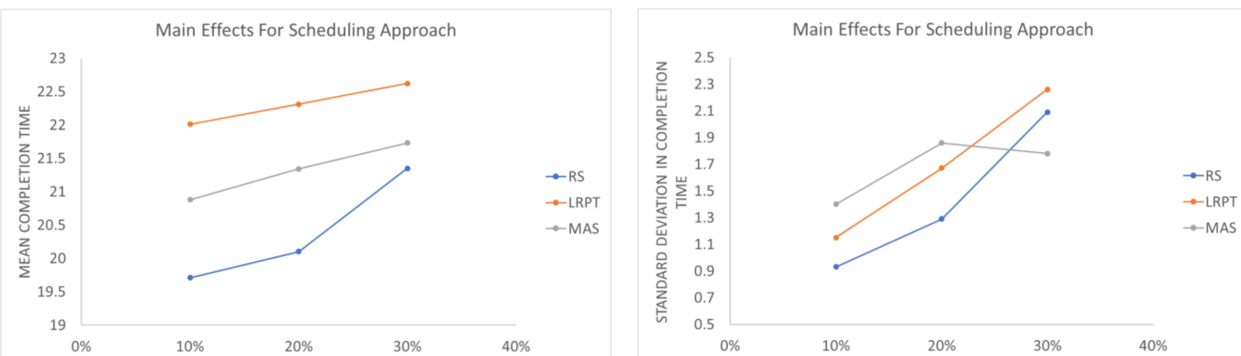


Figure 4.9 Main Effect Plots

4.4.3 Discussion

There are two key observations from our experiments. The first is that the completely reactive approaches were less sensitive to changes in the level of uncertainty in OED than the RS approach. This is important as it could reduce the impact differing operator proficiency could have on the system performance. The second is that the completely reactive scheduling approaches had lower standard deviations in their completion times. This means that there was more consistency and reliability in their performance.

The general trend observed in the results from Table 4.9 is that the mean completion time increases with increasing levels of uncertainty in the processing and set up times regardless of the dynamic scheduling approach used and the scenario being investigated. This is also confirmed by looking at the main effects plots in Figure 4.9. The figure shows all of the main effects plots have positive slopes.

The MAS and LRPT based approaches are less sensitive to OED uncertainty levels than the right-shifting approach. This is shown by the main effects for both mean completion time and the associated standard deviation as seen in Table 4.10 and Table 4.11. The main effects show that the MAS approach is more insensitive to changes in the level of uncertainty than the RS approach but more sensitive than the LRPT approach. The main effect of MAS on the completion time is 0.84 in comparison to 1.64 for the RS, and 0.61 for the LRPT. The difference between the mean completion times at low and high settings is ~4% for the MAS approach, and ~8.3% for the nominal schedule. We also see that there is a near linear relationship between the level of OED and the standard deviation in the mean completion time for the RS results. However, the MAS appears to plateau at 20% OED.

The RS approach, on average, has the lowest mean completion times regardless of the uncertainty levels. This can be seen by the main effects as seen in Table 4.10 and Table 4.11. These tables show that the mean completion times fall between 19.71 to 21.35 time units for RS, 22.01 to 22.62 for the LRPT, and 20.88 to 21.73 for the MAS approach. Similarly, for the standard deviation in the completion times, values fall between 0.93 to 2.09 for RS, 1.15 to 2.26 for the LRPT, and 1.40 to 1.86 for the MAS approach. However, at the highest uncertainty level, we see that there is no statistically significant difference between the performances of the MAS-based and the RS-based approaches. This can be seen for scenarios 1, 2 and 3 in Table 4.9, where the results of the ANOVA show p-values greater than 0.05. This suggests that the two sets of times do not have statistically significant difference. This is also corroborated by the main effects. The main effects show that the right-shifting approach and LRPT schedule marginally outperform the MAS approach at the low settings but are not distinguishable from the MAS approach when the uncertainty in the processing and setup times are at 30% of the mean. This observation makes sense, as when the demand volume is low relative to available machine resources there is greater benefit in waiting for the disruption to resolve than to reschedule (deviate from the near-optimal schedule). However, as the demand volume increases, it begins to become more advantageous to use an alternative route as opposed to waiting for the uncertainty to resolve.

Looking at the difference in the performance of the three approaches examined, we see that for scenario 1, the MAS approach yields the lowest completion times. The MAS approach yields times 1%-8% lower than the approaches using right-shifting the GA schedule. This is true regardless of the level of uncertainty in the OED. However, we see that the dispatching rule approach and our MAS approach perform comparably. This suggests that there are scenarios in which completely reactive scheduling is the best approach when faced with uncertainty in processing and setup times. In this scenario, the CNC machines which serve as multi-purpose machines are more available as we progress more downstream in the schedule. As such, they are more able to absorb uncertainty resulting from the deviations from the expected operation completion times. This gives the MAS an advantage as it can explore alternate, potentially more advantageous scheduling options whereas with right-shifting using a nominal schedule we do not explore these options.

Overall, the results suggest that MAS approach outperforms right-shifting approaches when there is excess machine capacity (lower utilization of duplicate machines) as is the case with scenario 1 or in instances with high uncertainty. However, when there is low uncertainty in the times, then right-shifting from a nominal schedule is the best approach. This observation makes sense, with little uncertainty, the scenarios examined are similar to the problems from section 4.3 and as such their results are inline.

4.5 Scheduling Under Uncertain Demand and Fixed Part Set-up and Processing Times

Scheduling under uncertain demand is a problem that has not been extensively examined in literature. To the best of our ability, we could not find much literature that focuses on scheduling in such scenarios. Literature involving scheduling under demand uncertainty commonly focuses on effective lot-sizing and/or batching (Petrovic et al., 2008) not specifically on scheduling. However, it is important to examine scheduling under uncertain demand as job shop scheduling typically utilizes a forecasted demand (Tompkins et al., 2004). This forecast is a prediction of the volume of each part that would be requested from the system. This implies that there is already a degree of uncertainty relating to the exact demand when scheduling decisions are being made. It is unclear what effect any deviation from the projected demand would have on schedule efficiency and stability. This uncertainty is exacerbated by the mass customization environment which has volatile demand due to the high number of product variants available to demand. Aside from the issue of having scheduling using forecast demand data, there is also the possibility of extenuating circumstances that can affect ability to fulfill demand. For example, there could be delayed delivery of raw materials. This can result in the actual volume of parts that can be produced varying from the projected demand. This would significantly impact the schedule efficiency and require changes to the schedule to address.

In this research, we will be employing our MAS-based approach for completely reactive scheduling to address this issue. To demonstrate its efficacy, we will be comparing this approach to two alternative approaches for scheduling under uncertain demand. The first approach used in our comparison is batching parts together. For this approach, the schedule is created assuming that orders for each part would be batched into lots of fixed sizes and pass through the job floor as one unit. By batching, we allow for a nominal schedule to be created despite uncertainty in demand information as the batch-sizing process should result in fixed number of batches for each part which can then be used to plan the schedule. For example, suppose the demand for part p_B is projected to fall between 8 and 10 units. We can plan for this by assuming that the part will be batched into groups of 5. A schedule would then be created for 2 batches of 5 for the part. In the worst-case scenario, the second batch would take less time to complete than expected. However, schedule stability is maintained. Batching effectively turns the uncertain demand problem into an uncertain processing time problem. The second approach we will be using is scheduling only after demand is confirmed for the production cycle. With this approach, there is no ambiguity in the demand, we are simply comparing how the near optimal schedule given the actual demand would compare to the performance of the MAS-based manufacturing system proposed.

In this subsection, we will present two different sets of comparison experiments involving our MAS-based smart manufacturing system. The first is a comparison between our proposed and scheduling using batch-sizes. The second is a comparison of how the MAS-based system working with real-time information on demand would perform against the near optimal schedule if the demand was known. For this set of experiments the MAS is compared to the near optimal schedule for each possible demand mix.

4.5.1 Numerical Experiments

In this research, we conducted two sets of experiments. Both experiments stem from the same problem. In this problem, we have the same furniture manufacturing system outlined previously. This system consists of 11 machines capable of producing 16 distinct parts. The problem is scheduling the for production of order of product, PD_P, where the makeup of PD_P = $\{p_1, p_2, p_3, p_4, p_5, p_6\}$. The demand for the product is uncertain with possible ranges for the product being specified.

Experiment Conditions

The numerical experiments are run using our own in-house simulation code written on MATLAB R2021b. The experiment conditions are as follows;

1) Number of repetitions for each experiment is 20
2) For experiment (1), demand for each part is randomly sampled at the beginning of each replication
3) For experiment (2) demand is fixed and deterministic for all experiment repetitions
4) The order (irrespective of experiment number) arrives to the system at time $T = 0$
5) Simulation terminates when the total order is completed

Experiment 1 Input Settings (Using Batches)

The system input settings for experiment 1 are as follows:

1) The four demand scenarios outlined in section 4.1 are investigated
2) Demand for each part follows the same uniform distribution of integers. There are two sets of demand distribution scenario inputs are to be examined
    a. Set 1: Minimum Demand 1, Maximum Demand 3
    b. Set 2: Minimum Demand 5, Maximum Demand 10
3) The manufacturing system is the same system outlined in section 4.2
4) The processing and setup times for each part on the given machines are fixed and deterministic (outlined in section 4.2)

Experiment 2 Input Settings (MAS Performance compared to Near-Optimal Schedule)

The input settings for experiment 2 are as follows;

1) The demand consisted of orders of six (6) distinct parts (p_1, p_2, p_3, p_4, p_5 and p_6).
    a. This order is scenario 1 as outlined in section 4.2
2) Five (5) demand scenarios are examined:
    a. $1 \times (p_1, p_2, p_3, p_4, p_5$ and $p_6)$
    b. $2 \times (p_1, p_2, p_3, p_4, p_5$ and $p_6)$
    c. $3 \times (p_1, p_2, p_3, p_4, p_5$ and $p_6)$
    d. $4 \times (p_1, p_2, p_3, p_4, p_5$ and $p_6)$
    e. $5 \times (p_1, p_2, p_3, p_4, p_5$ and $p_6)$
3) The near-optimal schedules for each scenario can be seen in Figures 4.10 to 4.14
4) The manufacturing system is the same system outlined in section 4.1.1

5) The processing and setup times for each part on the given machines are fixed and deterministic (outlined in section 4.1)

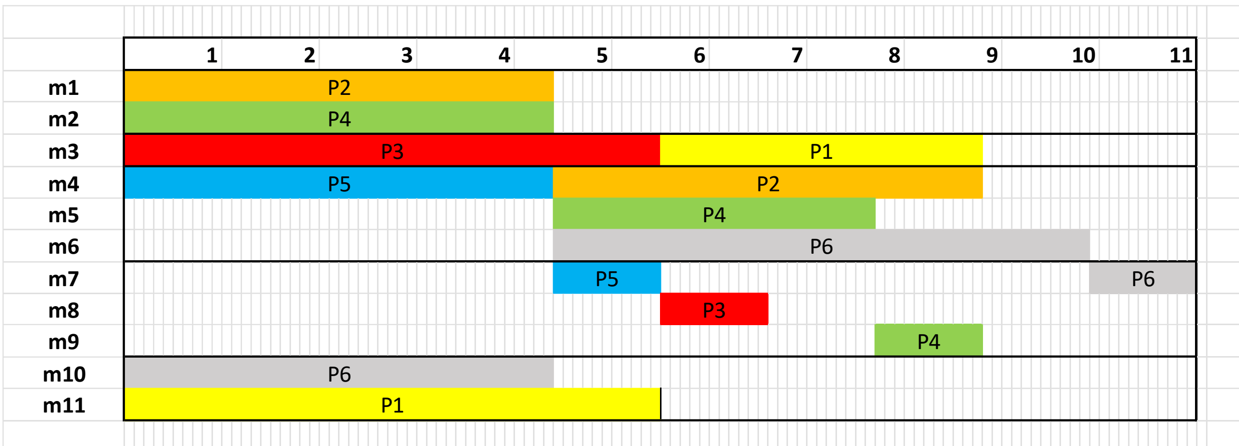


Figure 4.10 Schedule for Producing One Unit of the Product

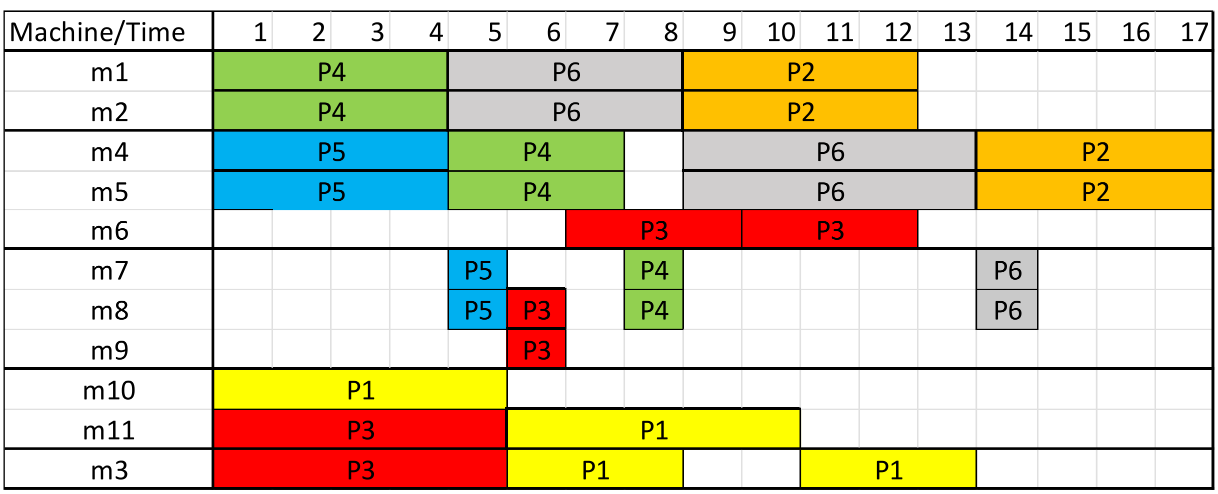


Figure 4.11 Schedule for Producing Two Units of the Product

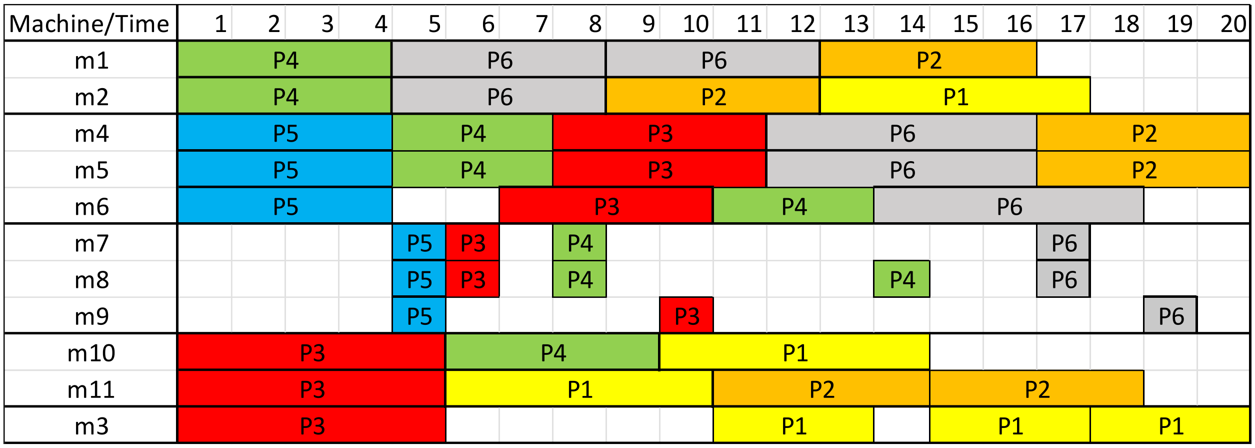


Figure 4.12 Schedule for Producing Three Units of the Product

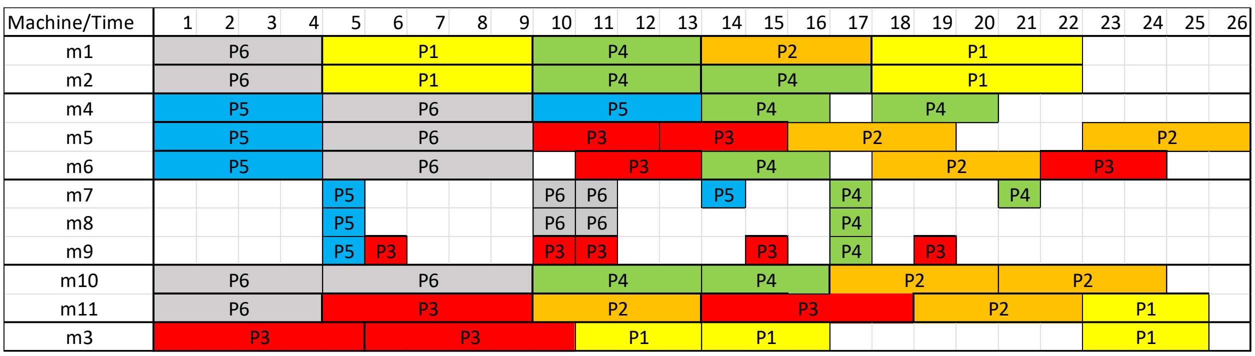


Figure 4.13 Schedule for Producing Four Units of the Product

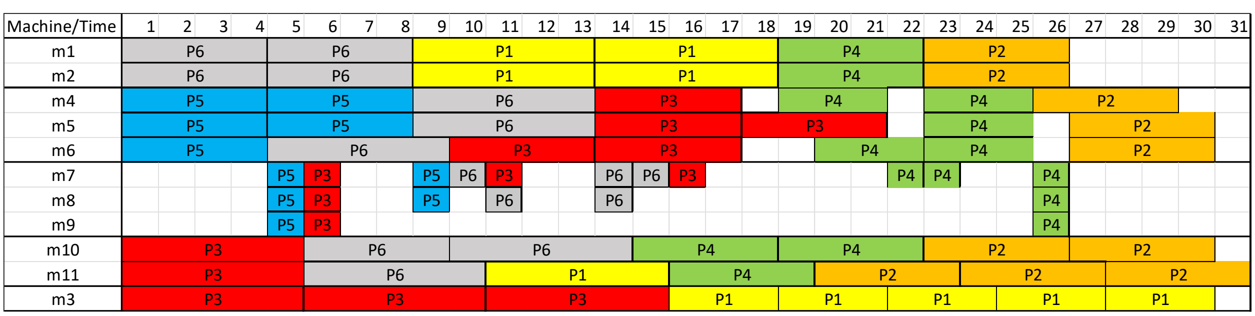


Figure 4.14 Schedule for Producing Five Units of the Product

4.5.2 Results

Table 4.12 and Table 4.13 below contain the results of the numerical experiments conducted for the demand orders presented in 4.5.1. These tables each contain the mean completion time and standard deviation of those completion times for processing a given order of parts using three distinct scheduling approaches for the varying levels of demand. We also present the P-values for hypothesis testing we conducted on the results of the experiments as well as the main effects for the different demand volumes.

Table 4.12 Completion Times Using Right-Shifting, Dispatching Rules. And MAS Approaches for Experiment 1(20 Repetitions)

Scenario	Level	RS Schedule		LRPT Scheduling		MAS Scheduling Approach		P-Value (H_0: $\mu_1 = \mu_3$)	P-Value (H_0: $\mu_2 = \mu_3$)
		Mean (μ_1)	STD (σ_1)	Mean (μ_2)	STD (σ_2)	Mean (μ_3)	STD (σ_3)		
1	Low	27.46	3.58	18.35	2.49	16.24	2.58	<0.001	<0.001
	High	89.98	9.38	54.63	3.39	48.38	4.41	<0.001	<0.001
2	Low	38.16	5.49	20.97	0.18	17.70	1.78	<0.001	<0.001
	High	124.41	10.95	86.34	0.97	50.49	3.91	<0.001	<0.001
3	Low	43.76	5.27	38.27	0.99	25.04	4.39	<0.001	<0.001
	High	150.37	14.54	111.50	1.33	66.02	3.11	<0.001	<0.001
4	Low	59.34	4.65	70.19	0.92	49.20	12.09	<0.001	<0.001
	High	200.82	14.33	238.64	2.64	151.36	12.67	<0.001	<0.001

Table 4.13 Completion Times Using Right-Shifting, Dispatching Rules. MAS Approaches for Experiment 2 (20 Repetitions)

Demand Volume	RS Schedule		LRPT-based Scheduling		MAS Scheduling Approach		P-Value	P-Value
	Mean (μ_1)	STD (σ_1)	Mean (μ_2)	STD (σ_2)	Mean (μ_3)	STD (σ_3)	(H_0: $\mu_1 = \mu_3$)	(H_0: $\mu_2 = \mu_3$)
1	11.00	n/a	11.01	0.01	11.45	0.75	0.01	0.01
2	17.60	n/a	18.72	0.01	18.96	1.11	<0.001	0.33
3	22.00	n/a	24.24	0.01	22.12	0.50	0.28	<0.001
4	27.60	n/a	31.02	0.88	28.67	0.93	<0.001	<0.001
5	34.00	n/a	36.54	0.39	35.31	1.20	<0.001	<0.001

4.5.3 Discussion

The results of the experiments show that the completely reactive scheduling approaches outperform RS when demand is uncertain. Of the two completely reactive approaches examined, the results indicate that the MAS approach performs the best of the scheduling approaches examined.

In the first set of experiments, we see that our proposed MAS approach significantly reduces the time required to complete the order in comparison to using a fixed nominal schedule and batching parts. MAS has a mean completion time that is consistently lower than the alternative approaches explored regardless of the level of uncertainty in the demand. This observation is corroborated by the results of the ANOVA we performed. We tested the null hypothesis that the means were equal, and in both cases, we found that $P < 0.05$. These results suggest that in situations where demand is difficult to accurately predict, it is better to use our proposed MAS or even dispatching rules.

From the first set of experiments, we see that any deviation in the demand from the planned volume results in a significant impact on the schedule efficiency. We draw this conclusion based on the results from this section as well as the observations from section 4.3.3. From section 4.3.3, we saw that the MAS approach we propose yields a close to optimal, but suboptimal solution in comparison to the RS solution when there is no uncertainty in the system inputs or conditions. As such, we can infer that since the MAS-based approach is significantly outperforming the RS solution, it is the result of the source of uncertainty in the demand and the approach to handling this uncertainty *a* priori.

The reduction in completion time when switching from using the RS approach to our proposed MAS shows that there is a benefit to treating parts as individual entities over batching them into groups of the same part. We see a 17.1% to 53.6% reduction in the completion time when demand falls between 1-3 units. This value increases to 24.6% to 59.4% reduction in the completion time when demand falls between 5-10. As such, we see the benefit of not grouping parts into batches may increase as the amount of demand uncertainty increases. Normally, treating parts as individual entities in a mass customization manufacturing environment would increase the difficulty of scheduling to the point of infeasibility. However, the MAS-based manufacturing system we propose is completely reactive, and there is no need to schedule ahead of time. This makes treating each part as an individual entity in the system practical.

The results of the second set of experiments suggest that our MAS approach is comparable to solutions derived by RS or built using LRPT as demand increases. Note, with these experiments, MAS performance is compared against the RS and LRPT based schedules whilst assuming that it was demand was known *a* priori. As such, we have insight into how the MAS results from the first set of experiments would compare to their optimal counterpart. The results show that the MAS performance was slightly worse for all demand volumes than the optimal schedule. Its completion times are 0.5% to 7.7% times higher. This is also supported by the results of the hypothesis testing conducted suggest that performance of all three approaches cannot be distinguished from each other ($P < 0.05$). However, this is to be expected as this set of experiments simply compares the performance of the MAS approach to alternate scheduling approaches without any uncertainty (only increasing demand).

Overall, we propose that our MAS approach is more practical than predictive-reactive dynamic scheduling in situations where there is any level of uncertainty in demand. It yields similar solutions with performance close to the near-optimal in situations without needing to know demand *a* priori.

4.6 Scheduling in Environment Subject to Machine Breakdown

Machine breakdown is the one of the two main causes of machine unavailability (the other is maintenance) (Ouelhadj & Petrovic,2009). However, of the two main causes of machine unavailability, machine breakdown is the one that occurs randomly. Machine breakdown is disruptive to system performance for two reasons. It results in uncertain downtime time for a given machine when breakdown occurs. This is because the time required to repair the machine is dependent on the type of damage and availability of tools, parts, and expertise required to repair said machine. This adds two additional sources of uncertainty to problems involving machine breakdown. The first is the mean time to failure (MTTF). This is a measure of how long it takes for a machine to fail again once brought online. The second is the mean time to repair (MTTR). This is a measure of how long it takes to repair the machine once it fails.

It is important to understand how the MAS model we propose performs in an environment that is subject to machine breakdown. It is also important to understand how that performance compares to that of other dynamic scheduling approaches when faced with the same conditions. In this subsection, we will present numerical experiments we have conducted to compare the performance of our MAS model against predictive-reactive approach (right-shifting with nominal GA derived schedule) and against dispatching rules-based approach (using LRPT). We will also present the results of the experiments as well as our analysis of the results.

4.6.1 Numerical Experiment

For the numerical experiments, we have modified the previously presented furniture problem to include mean time to failure (MTTF) and mean time to repair (MTTR) for each machine. The MTTF and MTTR for each machine are represented by an exponential distribution with a given mean. Both MTTF and MTTR have been assigned low, medium, and high settings. These settings are provided in Table 4.14. For the experiments we conduct, we assume that each machine has the same breakdown characteristics.

Table 4.14 High, Medium and Low Settings for Mean Time to Failure (MTTF) and Mean Time to Repair (MTTR)

LEVEL	MTTF	MTTR
Low	9	1
Medium	6	7
High	3	14

For each of the four (4) scenarios outlined in section 4.2, we conducted a full factorial experiment (3^2) with 100 replications each. The distinct sets of experiments conducted for each scenario can be seen in Table 4.15.

Table 4.15 Simulation Experiments Executed During Study

MTBF	LO	MED	HI	LO	MED	HI	LO	MED	HI
MTTR	LO	LO	LO	MED	MED	MED	HI	HI	HI

Experiment Conditions

The numerical experiments were run for the right-shifting, LRPT rule-based, and the MAS-based approaches for dynamic scheduling. The simulations are run using a script developed in MATLAB R2021b. The conditions for the experiments are as follows;

1) Each experiment is run for four distinct demand mix scenarios (outlined in Table 4.5)
    a. For each scenario all 9 combinations for levels of MTTF and MTTR are run
2) The order for all parts is assumed to arrive at the system at time zero, $T = 0$.
3) Each simulation experiment for each setting for each scenario is repeated 20 times
4) The mean completion times and the associated standard deviations are compared

4.6.2 Results

Table 4.16 below contains the results of the numerical experiments conducted for the four different scenarios presented in section 4.2. The table contains the mean completion time and standard deviation of those completion times for processing a given order of parts using three distinct scheduling approaches. We also present the P-values for hypothesis testing we conducted on the results of the experiments as well as the main effects of the machine breakdown variables. Tables 4.17 to 4.20 present the numerical values for the main effects for varying MTTF and MTTR levels. The main effects are visually represented in Figure 4.15.

Table 4.16 Simulation Experiment Results for Machine Breakdown

Scenario	MTTF Level	MTTR Level	RS		LRPT		MAS		P-Value (H_0: $\mu_1 = \mu_3$)	P-Value (H_0: $\mu_2 = \mu_3$)
			(μ_1)	(σ_1)	(μ_2)	(σ_2)	(μ_3)	(σ_3)		
1	LO	LO	13.26	1.80	12.80	1.69	12.79	2.23	0.46	0.45
	MED	LO	14.47	2.23	13.80	2.02	13.65	2.70	0.30	0.28
	HI	LO	16.53	2.59	16.15	2.28	16.27	3.17	0.78	0.77
	LO	MED	36.88	17.27	25.54	9.42	31.21	11.81	0.23	0.09
	MED	MED	46.48	17.11	33.58	14.29	33.86	11.52	0.01	<0.001
	HI	MED	60.78	13.57	50.41	18.11	51.24	16.53	0.05	0.08
	LO	HI	63.11	33.07	48.53	22.04	49.26	24.53	0.13	0.06
	MED	HI	74.72	29.36	75.81	31.29	64.01	30.82	0.26	0.28
	HI	HI	118.58	34.42	119.93	44.43	97.60	37.13	0.06	0.11
2	LO	LO	17.67	2.17	21.90	1.74	21.86	3.13	<0.001	<0.001
	MED	LO	19.25	2.73	21.96	2.40	22.83	3.15	<0.001	<0.001
	HI	LO	21.77	2.67	27.53	3.26	27.00	3.79	<0.001	<0.001
	LO	MED	43.92	13.53	43.72	12.83	39.78	12.04	0.31	0.29
	MED	MED	54.26	20.77	51.69	13.57	49.58	14.56	0.41	0.29
	HI	MED	73.78	17.73	80.42	13.51	75.79	20.81	0.74	0.72
	LO	HI	85.89	35.21	67.98	21.32	69.98	31.89	0.13	0.06
	MED	HI	94.15	26.69	90.01	28.76	89.85	29.50	0.63	0.64
	HI	HI	142.60	40.03	135.11	40.40	137.65	34.47	0.68	0.68
3	LO	LO	22.78	2.40	27.04	1.87	26.17	3.35	<0.001	<0.001
	MED	LO	24.08	2.56	28.29	2.52	27.02	3.67	<0.001	<0.001
	HI	LO	27.85	3.42	32.04	2.28	31.72	4.97	<0.001	<0.001
	LO	MED	52.50	17.30	56.73	21.08	50.91	16.87	0.77	0.79
	MED	MED	58.95	14.17	63.02	14.85	60.28	15.82	0.78	0.78
	HI	MED	87.83	20.23	89.95	18.26	93.47	24.07	0.42	0.40
	LO	HI	88.62	31.13	77.97	27.29	81.33	35.32	0.49	0.47
	MED	HI	105.09	37.72	106.51	25.31	103.94	28.38	0.91	0.89
	HI	HI	166.71	44.15	181.36	38.83	164.63	46.26	0.88	0.88
4	LO	LO	32.01	2.52	37.72	2.26	36.29	5.07	<0.001	<0.001
	MED	LO	33.60	2.75	39.38	3.13	37.68	5.22	<0.001	<0.001
	HI	LO	39.78	5.96	47.55	3.41	43.43	5.16	0.04	0.01
	LO	MED	64.52	15.35	66.52	12.80	66.79	16.91	0.66	0.63
	MED	MED	80.81	18.87	79.57	15.40	79.83	18.18	0.87	0.85
	HI	MED	119.41	24.62	120.06	18.73	128.06	25.41	0.27	0.22
	LO	HI	119.00	20.42	111.44	21.46	116.62	21.21	0.72	0.72
	MED	HI	149.57	43.88	143.55	33.11	135.24	39.87	0.28	0.22
	HI	HI	212.07	43.65	231.29	41.48	213.64	41.92	0.91	0.91

Table 4.17 Main Effects of Varying MTTF Levels on the Mean Completion Times Using Each Solution Approach

	LO	MED	HIGH	EFFECT
RS	53.35	62.95	90.64	37.29
LRPT	50.82	62.26	94.32	43.50
MAS	50.25	59.81	90.04	39.79

Table 4.18 Main Effects of Varying MTTF Levels on the Standard Deviation of Mean Completion Times Using Each Solution Approach

	LO	MED	HIGH	EFFECT
RS	16.01	18.24	21.09	5.08
LRPT	12.98	15.55	20.42	7.44
MAS	15.36	16.95	21.97	6.61

Table 4.19 Main Effects of Varying MTTR Levels on the Mean Completion Times Using Each Solution Approach

	LO	MED	HIGH	EFFECT
RS	23.59	65.01	118.34	94.75
LRPT	27.18	63.43	115.79	88.61
MAS	26.39	63.40	110.31	83.92

Table 4.20 Main Effects of Varying MTTR Levels on the Standard Deviation of Mean Completion Times Using Each Solution Approach

	LO	MED	HIGH	EFFECT
RS	2.82	17.54	34.98	32.16
LRPT	2.41	15.24	31.31	28.90
MAS	3.80	17.04	33.44	29.64

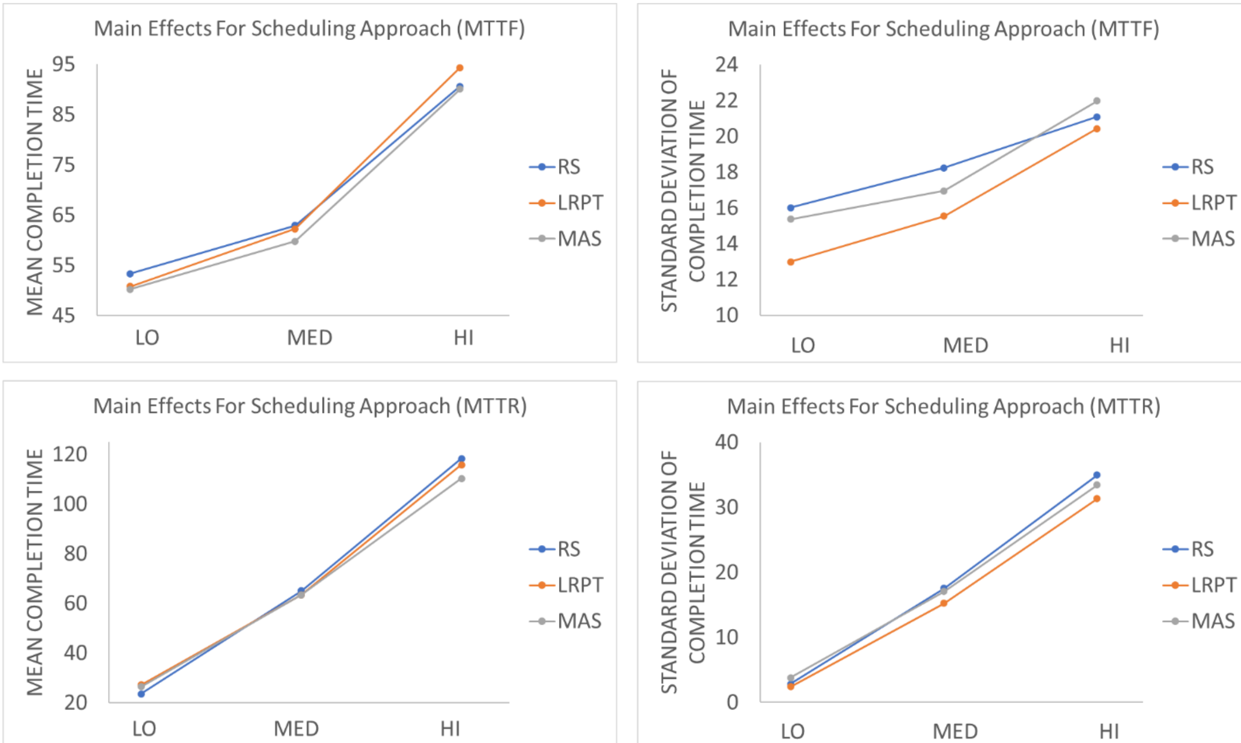


Figure 4.15 Main Effect Plots

4.6.3 Discussion

The results show that the MAS has the best response to uncertainty resulting from machine breakdowns of the three scheduling approaches studied. This solution approach yields the lowest completion times for processing demand orders with comparable standard deviations to the other approaches studied. This can be seen in the main effects for the mean completion times for the uncertainty in MTTF and MTTR (Table 4.17 and Table 4.19) and their associated standard deviations (Table 4.18 and Table 4.20). Table 4.17 shows the main effect on mean completion time when varying MTTF falls between 50.25 and 90.04 time units when using MAS in comparison to 53.35 to 90.64 time units using RS, and between 50.82 to 94.32 time units using LRPT. Looking at the main effects, we can see that the MAS yields the lowest mean completion times of the approaches whilst having comparable standard deviations to the RS and LRPT schedules. Table 4.18 shows that the main effects on the standard deviation in the completion time generally falls between 12.98 to 21.97 times units regardless of the approach used. The results suggest that the MAS approach slightly outperforms the alternative approaches.

The main effects for MTTF indicate that the MAS yields the shortest completion times regardless of the setting. For the low and medium settings, we see a more pronounced difference between the performances MAS and the RS-schedule and LPRPT schedule respectively. Looking at the main effects (Table 4.18) for the MTTF that examines the standard deviation for the completion times, we see that the standard deviations are similar for all three approaches with respect to MTTF. However, we see that the MAS is more sensitive to the uncertainty in the MTTF than the alternative approaches than using RS. With a main effect of 6.61 to 5.08. MAS is less sensitive than LRPT with respect to standard deviation (6.61 to 7.44).

The main effects for the MTTR imply that the MAS approach slightly outperforms the RS and LRPT schedules with respect to mean completion times. Looking at Table 4.19, we see that the main effect of MTTR on completion time falls between 26.39 to 110.31 time units using MAS. This is in comparison to 23.59 to 118.34 time units when using RS, and 27.18 to 115.79 time units when using LRPT. The MAS approach also results in less sensitivity to the level of MTTR. With the MAS approach having a mean main effect of 83.92 time units, RS schedule having a mean main effect of 94.75 time units, and the LRPT schedule having 88.61 time units. For the low MTTR setting the MAS has the second lowest mean completion time with the RS having the lowest. However, for each subsequent level of MTTR (medium, and high) the MAS has the lowest mean completion time of all the approaches examined. Looking at the main effects of the MTTR levels on the standard deviation of the completion times, we see that all three approaches are comparable. At the lowest MTTR level, the MAS approach has the highest standard deviation. However, as the level of MTTR increases, the MAS has the lowest standard deviation of the approaches examined.

Overall, the results of these experiments suggest that in situations with low uncertainty in machine availability, the MAS approach is not as effective as using a predictive-reactive scheduling strategy but yields very comparable results. However, as the level of uncertainty in machine availability increases, the MAS approach outperforms the dynamic scheduling approaches explored. This observation makes sense given that greater levels of uncertainty should mean that the ideal schedule should significantly differ from the nominal schedule (greater schedule instability). Given that the MAS approach is completely reactive, it can adjust to accommodate the disturbances to the system better. As we have seen from previous experiments (see 4.3 and 4.5) the MAS approach performs comparably to the near-optimal schedule derived using RS. It also outperforms the simplicity of the dispatching rule. So, the MAS approach would yield results close to the optimal schedule if all disturbances could be accounted for *a* priori. However, it offers the same flexibility and robustness as the dispatching rule. Given the results of these experiments, we would recommend the MAS approach for manufacturing systems subject to uncertainty from machine availability over predictive-reactive scheduling using right shifting and a nominal schedule or using dispatching rules.

4.7 Scheduling in Environment Subject to Job-Related and Resource Related Sources of Uncertainty

The typical dynamic job shop scheduling problem seen in literature typically focuses on either job-related sources of uncertainty (processing times, new arrivals, demand) or resource-related uncertainty (machine availability, raw material availability, etc.) (Ouelhadj & Petrovic, 2009). These problems typically focus on one or two sources of uncertainty. However, this is not representative of a real-world manufacturing environment which is subject to multiple sources of uncertainty. It is important that any scheduling strategy or approach employed be effective when used in the real manufacturing system.

In this section, we will conduct a series of experiments under both job-related and resource-related sources of uncertainty. This is important as there may be interactions between the different sources of uncertainty that are being ignored. These interactions or compounding effects may result in different efficacy for the different dynamic scheduling approaches being employed for thr given problem. It is for this reason that we will specifically be comparing the performance of our MAS-based approach to that of dynamic scheduling using dispatching rules (LPRT), and right-shifting in a manufacturing environment subject to uncertainty in processing times, demand volume, and machine availability due to breakdown.

In the upcoming subsections, we will present a description of the numerical experiment conducted as well as the results of these experiments, and an analysis of our findings.

4.7.1 Numerical Experiments

For our numerical experiment, the problem we are examining is based on scenario 1. The manufacturing system is a furniture manufacturing consisting of 11 machines (as described in section 4.2). This system can be run using three dynamic scheduling approaches (right-shifting, dispatching rules or our MAS approach). The manufacturing system is tasked with producing six (6) distinct parts (p_1, p_2, p_3, p_4, p_5 and p_6) in an environment subject to four sources of uncertainty. The sources of uncertainty are as follows:

1) Demand volume uncertainty
2) Operation execution duration uncertainty
3) Machine mean time to failure uncertainty
4) Machine mean time to repair uncertainty

We have chosen to use this scenario (demand for uncertain volume of six distinct parts) because the previous experiments conducted in this chapter (sections 4.3, 4.4, 4.5, and 4.6) show that this mix of parts can be produced in times that are indistinguishable from each other using any of the three dynamic scheduling approaches being examined. Given that each solution approach is comparable in experiments where only one source of uncertainty was considered, it serves as good candidate to examine if there are compounding effects with multiple sources of uncertainty. It also allows us to investigate the performance of each scheduling approach under these conditions.

For the experiment we treat each source of uncertainty as a factor with three level settings. This specific information for each factor's levels can be seen in Table 4.21. Each factor has similar values as were used in the experiments conducted in sections 4.4, 4.5 and 4.6. We conducted a full factorial experiment with 20 repetitions of each simulation experiment. This is done using each dynamic scheduling approach (RS, LRPT, and MAS). This results in a total of 81 distinct experiments and 1620 experiments total using each scheduling approach.

Table 4.21 Level Settings for each Source of Uncertainty Our Manufacturing System is Subject To

LEVEL	Demand	Operation Duration Uncertainty	MTTF	MTTR
Low	1	10 % of μ	9	1
Medium	1 to 3	20 % of μ	6	7
High	5 to 10	30 % of μ	3	14

Experiment Conditions

The numerical experiments were run for the right-shifting, LRPT rule-based, and the MAS-based approaches (real-time scheduling). The simulations are run using a script developed in MATLAB R2021b. The conditions for the experiments are as follows;

1) Each experiment is run for four distinct demand mix scenarios (outlined in Table 4.5)
    a. For each scenario all 9 combinations for levels of MTTF and MTTR are run
    b. For each scenario all combination of uncertainty in OED are run
    c. For each scenario all combinations of demand uncertainty presented are run
2) The order for parts arrives to the system at time zero, $T = 0$.
3) Each simulation experiment for each setting for each scenario is repeated 20 times

4.7.2 Results

Tables 4.22 to 4.24 contain the mean completion for each simulation experiment run as well as the associated standard deviation for those times. These tables also present the experiment settings that yielded these results. We performed hypothesis testing on this data to determine if their means are equal, and the P-values calculated are shown in the tables. This section also presents the main effects of each factor on the mean completion times and their standard deviations. These can be seen in Tables 24-31 and are depicted in Figure 4.16.

Table 4.22 Mean Completion Times and Associated Standard Deviations for Simulation Experiments with Fixed Demand

DEMAND	TIME	MTTF	MTTR	RS		LRPT		MAS		P-Value (H_0: $\mu_1 = \mu_3$)	P-Value (H_0: $\mu_2 = \mu_3$)
				μ_1	σ_1	μ_2	σ_2	μ_3	σ_3		
1	10	9	1	13.56	0.94	12.58	1.55	12.01	1.16	<0.001	<0.001
1	10	9	7	32.83	12.33	33.05	16.33	28.80	11.04	0.28	0.36
1	10	9	14	72.75	48.83	70.82	31.86	37.77	13.54	<0.001	<0.001
1	10	6	1	14.47	1.99	14.43	2.42	12.55	1.08	<0.001	<0.001
1	10	6	7	47.91	17.18	42.08	17.27	41.78	16.89	0.26	0.26
1	10	6	14	76.23	25.35	60.36	29.28	56.18	18.05	<0.001	<0.001
1	10	3	1	17.99	1.88	15.35	2.47	14.71	1.94	<0.001	<0.001
1	10	3	7	71.60	20.56	50.52	12.89	47.51	10.83	<0.001	<0.001
1	10	3	14	141.07	38.95	102.33	35.96	93.52	32.19	<0.001	<0.001
1	20	9	1	14.30	1.98	13.82	1.99	11.19	1.14	<0.001	<0.001
1	20	9	7	38.03	18.16	31.37	12.97	25.22	6.84	<0.001	<0.001
1	20	9	14	63.13	26.52	56.00	27.49	48.22	23.60	0.06	0.07
1	20	6	1	14.95	1.84	13.95	1.78	13.06	1.90	0.00	0.00
1	20	6	7	55.05	13.36	40.02	14.59	31.39	8.79	0.00	0.00
1	20	6	14	86.19	30.02	59.33	23.64	70.01	28.53	0.08	0.05
1	20	3	1	17.10	2.73	16.96	2.87	15.72	2.55	0.10	0.11
1	20	3	7	63.42	14.67	58.16	19.96	50.02	11.62	<0.001	<0.001
1	20	3	14	119.07	35.48	99.87	40.46	84.88	26.67	<0.001	<0.001
1	30	9	1	13.92	1.64	13.47	2.10	13.01	1.74	0.09	0.13
1	30	9	7	42.73	9.84	30.21	11.79	26.66	7.46	0.00	0.00
1	30	9	14	58.52	28.78	49.02	26.60	49.13	20.63	0.24	0.21
1	30	6	1	14.81	1.96	14.48	2.55	13.47	1.39	<0.001	<0.001
1	30	6	7	44.74	9.02	38.39	9.12	36.81	12.75	<0.001	<0.001
1	30	6	14	81.89	30.86	56.79	19.92	61.47	23.80	<0.001	<0.001
1	30	3	1	17.57	2.85	17.11	3.03	15.13	2.01	<0.001	<0.001
1	30	3	7	61.87	14.15	53.29	13.10	47.74	16.71	<0.001	<0.001
1	30	3	14	127.48	44.88	94.97	33.29	81.80	27.30	<0.001	<0.001

Table 4.23 Mean Completion Times and Associated Standard Deviations for Simulation Experiments with Uncertain Demand Between 1 and 3 Units

DEMAND	TIME	MTTF	MTTR	RS		LRPT		MAS		P-Value	P-Value
				μ_1	σ_1	μ_2	σ_2	μ_3	σ_3	(H_0: $\mu_1 = \mu_3$)	(H_0: $\mu_2 = \mu_3$)
1 to 3	10	9	1	36.46	2.05	21.10	1.26	18.68	1.41	<0.001	<0.001
1 to 3	10	9	7	70.12	14.62	44.34	12.55	40.29	10.05	<0.001	<0.001
1 to 3	10	9	14	116.70	36.19	65.63	18.93	62.96	25.19	<0.001	<0.001
1 to 3	10	6	1	30.70	3.06	21.68	2.32	22.73	1.47	<0.001	<0.001
1 to 3	10	6	7	83.81	20.81	41.74	12.94	49.36	10.71	<0.001	<0.001
1 to 3	10	6	14	85.86	25.89	87.97	24.20	85.89	24.07	1.00	1.00
1 to 3	10	3	1	44.10	4.94	29.89	3.04	22.05	2.79	<0.001	<0.001
1 to 3	10	3	7	144.78	16.27	89.00	22.43	91.27	15.34	<0.001	<0.001
1 to 3	10	3	14	225.65	57.74	172.89	45.50	124.84	28.46	<0.001	<0.001
1 to 3	20	9	1	36.87	3.12	26.02	2.36	18.16	2.10	<0.001	<0.001
1 to 3	20	9	7	61.88	14.44	48.88	16.10	48.42	15.03	<0.001	<0.001
1 to 3	20	9	14	107.85	35.85	79.09	18.90	69.13	18.19	<0.001	<0.001
1 to 3	20	6	1	34.21	2.71	20.94	2.32	18.61	2.98	<0.001	<0.001
1 to 3	20	6	7	87.82	25.88	57.89	12.80	47.14	10.32	<0.001	<0.001
1 to 3	20	6	14	119.36	45.86	88.64	20.04	79.65	18.92	<0.001	<0.001
1 to 3	20	3	1	32.33	3.34	25.43	2.15	28.09	3.13	<0.001	<0.001
1 to 3	20	3	7	145.63	38.83	75.79	15.92	83.81	18.45	<0.001	<0.001
1 to 3	20	3	14	241.64	48.57	124.39	46.54	125.52	38.69	<0.001	<0.001
1 to 3	30	9	1	37.79	2.32	22.75	2.39	21.96	2.06	<0.001	<0.001
1 to 3	30	9	7	71.82	14.75	36.33	11.68	41.23	10.15	<0.001	<0.001
1 to 3	30	9	14	122.79	43.25	61.19	26.73	66.03	24.15	<0.001	<0.001
1 to 3	30	6	1	39.03	5.35	20.03	2.52	19.19	1.66	<0.001	<0.001
1 to 3	30	6	7	67.57	21.16	62.49	10.02	46.46	13.22	<0.001	<0.001
1 to 3	30	6	14	138.08	38.31	94.91	28.06	78.22	23.76	<0.001	<0.001
1 to 3	30	3	1	36.96	3.37	24.31	2.74	25.74	3.37	<0.001	<0.001
1 to 3	30	3	7	139.12	27.23	82.33	11.62	62.60	12.71	<0.001	<0.001
1 to 3	30	3	14	201.77	58.10	156.96	25.75	114.48	29.87	<0.001	<0.001

Table 4.24 Mean Completion Times and Associated Standard Deviations for Simulation Experiments with Uncertain Demand Between 5 and 10 Units

DEMAND	TIME	MTTF	MTTR	RS		LRPT		MAS		P-Value (H_0: $\mu_1 = \mu_3$)	P-Value (H_0: $\mu_2 = \mu_3$)
				μ_1	σ_1	μ_2	σ_2	μ_3	σ_3		
5 to 10	10	9	1	99.15	3.78	58.56	2.41	49.08	2.28	<0.001	<0.001
5 to 10	10	9	7	229.89	34.98	117.73	13.98	99.89	12.67	<0.001	<0.001
5 to 10	10	9	14	331.05	76.10	181.08	31.73	149.15	20.70	<0.001	<0.001
5 to 10	10	6	1	117.86	5.06	56.86	2.86	49.41	1.86	<0.001	<0.001
5 to 10	10	6	7	208.14	22.24	151.98	15.36	106.04	12.02	<0.001	<0.001
5 to 10	10	6	14	390.51	59.58	214.03	27.60	208.13	33.64	<0.001	<0.001
5 to 10	10	3	1	135.07	7.06	74.68	2.63	70.01	2.62	<0.001	<0.001
5 to 10	10	3	7	319.26	34.37	186.81	23.02	152.81	19.46	<0.001	<0.001
5 to 10	10	3	14	594.41	81.25	344.02	49.48	258.65	27.93	<0.001	<0.001
5 to 10	20	9	1	109.57	3.89	64.78	2.51	56.99	2.32	<0.001	<0.001
5 to 10	20	9	7	181.41	25.74	119.59	17.71	87.69	12.60	<0.001	<0.001
5 to 10	20	9	14	259.54	70.12	180.79	35.95	137.43	24.82	<0.001	<0.001
5 to 10	20	6	1	115.43	5.49	59.93	3.12	59.69	3.60	0.00	0.00
5 to 10	20	6	7	263.64	48.77	139.84	16.32	110.03	8.47	0.00	0.00
5 to 10	20	6	14	397.67	51.65	263.19	41.41	161.79	32.48	0.00	0.00
5 to 10	20	3	1	150.07	7.10	84.82	4.02	62.71	3.94	0.00	0.00
5 to 10	20	3	7	391.88	59.83	196.23	28.08	173.90	23.18	0.00	0.00
5 to 10	20	3	14	613.45	87.00	321.42	37.16	319.95	30.58	0.00	0.00
5 to 10	30	9	1	110.23	6.27	66.70	2.96	54.89	2.52	0.00	0.00
5 to 10	30	9	7	205.05	33.86	102.20	12.64	90.94	16.46	0.00	0.00
5 to 10	30	9	14	298.58	53.68	167.74	25.76	134.71	23.67	0.00	0.00
5 to 10	30	6	1	129.82	7.12	62.31	3.82	55.94	4.41	0.00	0.00
5 to 10	30	6	7	207.66	32.32	132.14	21.04	121.01	18.23	0.00	0.00
5 to 10	30	6	14	369.10	66.46	228.97	34.00	193.89	36.38	0.00	0.00
5 to 10	30	3	1	133.85	8.30	83.77	7.10	65.39	4.10	0.00	0.00
5 to 10	30	3	7	325.13	43.00	218.52	21.68	174.20	22.58	0.00	0.00
5 to 10	30	3	14	643.68	105.22	281.06	49.50	325.78	35.49	0.00	0.00

Table 4.25 Main Effects of Demand Uncertainty on Mean Completion Time

	1	1 TO 3	5 TO 10	**EFFECT**
LRPT	42.92	62.32	154.07	111.15
MAS	38.51	56.02	130.74	92.23
RS	52.71	94.84	271.52	218.81

Table 4.26 Main Effects of Demand Uncertainty on Standard Deviation of the Mean Completion Time

	1	1 TO 3	5 TO 10	**EFFECT**
LRPT	15.46	14.88	19.77	4.31
MAS	12.30	13.64	16.26	3.96
RS	16.92	22.74	38.53	21.61

Table 4.27 Main Effects of Operation Execution Duration Uncertainty on Mean Completion Time

	10%	20%	30%	**EFFECT**
LRPT	87.46	87.67	84.16	86.43
MAS	74.30	75.50	75.48	75.09
RS	138.96	141.54	138.58	139.69

Table 4.28 Main Effects of Operation Execution Duration Uncertainty on Standard Deviation of the Mean Completion Time

	10%	20%	30%	**EFFECT**
LRPT	17.12	17.38	15.61	-1.51
MAS	13.31	14.13	14.76	1.45
RS	24.96	26.78	26.45	1.49

Table 4.29 Main Effects of MTTF Level on Mean Completion Time

	LOW	MED	HIGH	**EFFECT**
LRPT	65.74	79.46	114.11	48.37
MAS	55.54	68.52	101.22	45.68
RS	105.06	123.06	190.96	85.90

Table 4.30 Main Effects of MTTF Level on Standard Deviation of the Mean Completion Time

	LOW	MED	HIGH	**EFFECT**
LRPT	14.42	14.86	20.83	6.41
MAS	11.61	13.75	16.83	5.22
RS	23.11	22.94	32.14	9.03

Table 4.31 Main Effects of MTTR Level on Mean Completion Time

	LOW	MED	HIGH	**EFFECT**
LRPT	35.43	84.48	139.39	103.96
MAS	31.12	72.71	121.45	90.33
RS	58.08	135.66	225.33	167.25

Table 4.32 Main Effects of MTTR Level on Standard Deviation of the Mean Completion Time

	LOW	MED	HIGH	**EFFECT**
LRPT	2.71	15.70	31.69	28.98
MAS	2.35	13.50	26.34	23.99
RS	3.78	24.39	50.02	46.24

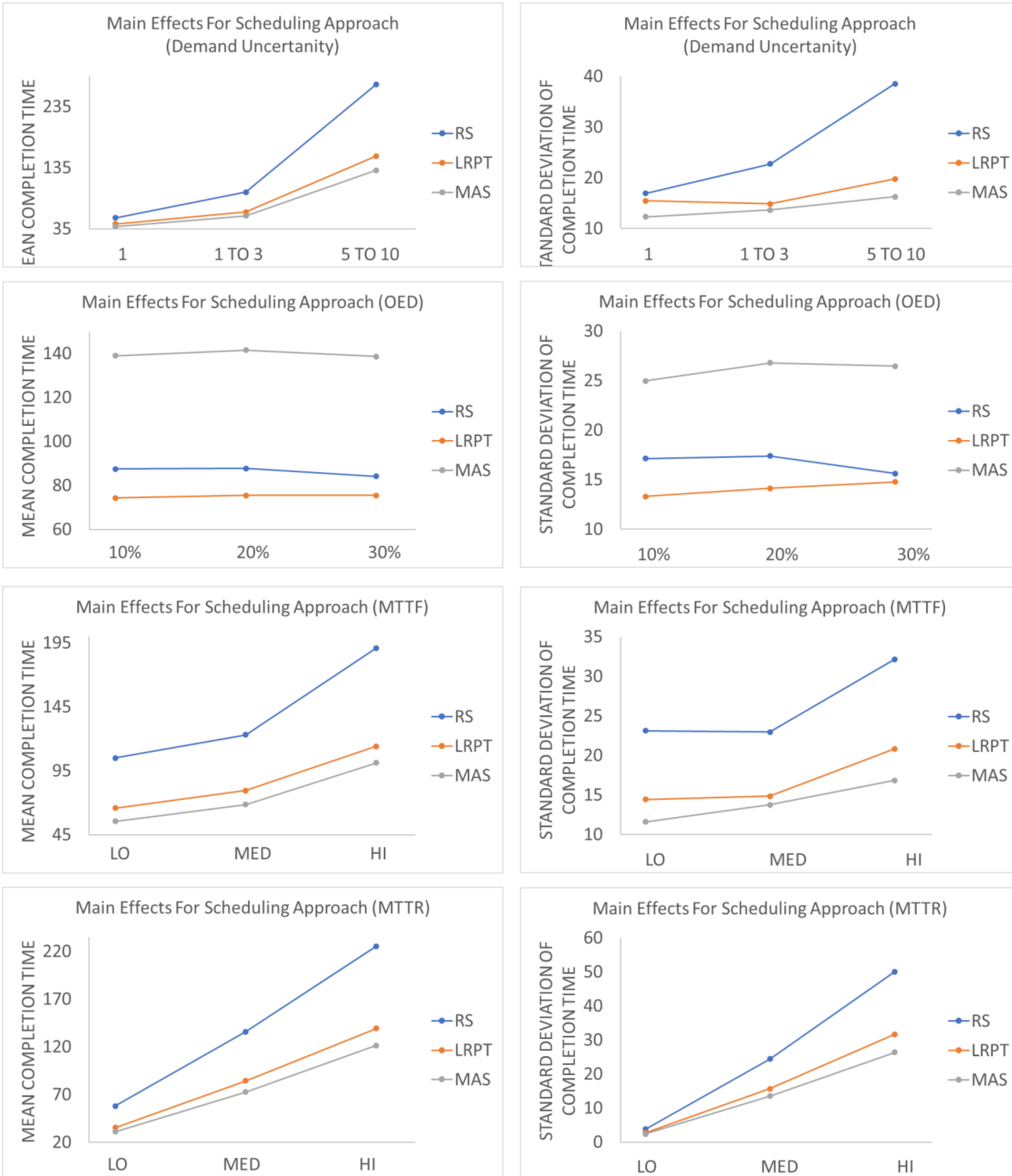


Figure 4.16 Main Effects Plots

4.7.3 Discussion

One of the major observations from these experiments is that the MAS and LRPT approaches (both completely reactive scheduling approaches) outperform the RS approach (a predictive-reactive approach). This is important as the results suggest that looking at the performances of dynamic scheduling methods whilst investigating single sources of uncertainty might not be truly indicative of how these scheduling methods would perform in real-life manufacturing environments. With one source of uncertainty, it stands to reason that a predictive-reactive approach would outperform completely reactive approaches. This is because we begin with the optimal schedule and then adjust to a new optimal for each disruption. However, if there is significant volatility in the system (frequent disruptions), there will be high levels of schedule instability and low schedule efficiency. If these disruptions are frequent enough, there would either need to be constant change to the schedule, or significant delays due to adhering to the original schedule. This is reflected in the results. The results imply that the compounding effects of having multiple uncertainty sources are significant. In Tables 4.22 to 4.24, we observe consistent, poor performance of the RS relative to the LRPT and MAS approaches when in the previous sections, they were equivalent or worse when compared to the RS. This information is also corroborated in the main effects as shown in Tables 4.25 to 4.32. The main effects show that the RS approach was consistently the most sensitive to the level of each factor investigated (with the exception of the OED uncertainty). This was true for both the main effects of the mean completion time and the standard deviation in the completion times.

The experiment results also indicate that there is a benefit to the additional intelligence offered by the MAS over the simple intelligence offered by the LRPT dispatching rules. Looking at the experimental run results shown in Tables 4.22 to 4.24, we see that the MAS outperforms the LRPT in most of the experimental runs executed. This result is to be expected as it is reasonable to assume that using more complex decision-making would yield better results for real-time scheduling.

To discern the effect of each variable for this set of numerical experiments, we compiled the main effects. Looking at the results from Table 4.25, we see that MAS yields the lowest times regardless of the demand uncertainty setting when looking at main effects. The MAS approach also has the lowest standard deviations. This suggests that the MAS is the most reliable approach with respect to its ability to handle uncertain demand. However, it is interesting to note that whilst MAS yields the best times most consistently, LRPT is the least sensitive to increasing uncertainty in demand. As uncertainty in demand increases from 1 to between 1-3 units, we see an increase in mean completion time of 1.45 times when LRPT and MAS are used, and 1.8 when RS is used. When goes from 1 to between 5-10 units, LRPT times increase by 3.6 times, MAS times increase by 4.5 times and RS times increase by 5.1 times. This appears to be a linear relationship.

Looking at Table 4.27, we see that the main effects for operation execution duration is marginal. For LRPT, the difference between the largest and smallest values for the effects is ~4%. For the MAS, the difference between the largest and smallest values for the effects is ~2%. For the RS, the difference between the largest and smallest values for the effects is ~2%. The standard deviations also follow a similar pattern. This is to be expected as the mean completion time required to execute the operations has not changed, only the size of the standard deviation. Looking at the main effect of OED uncertainty on the standard deviation, we see that it rises with the level of uncertainty and then plateaus at 20% when using LRPT and RS, but no plateau when using MAS. This result suggests that MAS approach may be more sensitive to changes in OED with respect to consistency in outcomes than the alternate scheduling approaches studied. Despite having standard deviations being more sensitive to OED uncertainty, the MAS still had the lowest standard deviations, indicating that it had the most reliable performance.

Looking at the main effect of MTTF, we see that MAS yields the lowest times with the lowest standard deviations in the times. The mean main effect of MTTF on completion times is 75.09 time units for MAS, 86.43 time units using LRPT, and 139.69 time units using RS. With changing levels from low to medium, to high, each approach has the same increase in the time yielded; ~1.2 times increase and ~1.8 times increase in completion times respectively. With standard deviation main effects for MTTF, we see that for both LRPT and RS, there is little difference between the low and medium settings. However, there is a significant jump between medium and high settings. This suggests that the standard deviation may exponentially increase with increasing uncertainty in the MTTF when LRPT or RS is the scheduling method employed. With respect to the RS, we see that the standard deviation appears to linearly increase with the level of MTTF uncertainty. Overall, these results suggest that the RS performance is more stable than the alternatives studied and yields the shortest completion times.

Looking at the main effects of MTTR level on the completion times, we see that MAS yielded the lowest completion times with RS yielding the highest times. Each different scheduling approach is equally sensitive to changes in level of MTTR with the transition from low to medium resulting in a ~2.3 times increase in time, and the increase from low to high resulting in a ~3.9 times increase in time. This suggests that they all increase linearly with respect to the MTTR level. With respect to the standard deviations, the same pattern is observed. MAS yields the lowest standard deviations. The relation between the MTTR level and standard deviation appears to be linear.

For this example, we can see that the most significant effects are from the demand uncertainty and MTTR level uncertainty. They cause the most significant changes in the system performance, with the highest settings for both causing an increase of 5 times, and 3.8 times in completion times respectively. The least significant effect appears to be from OED uncertainty, which appears to mostly affect the consistency of the performance of the scheduling approach.

Overall, our results suggest that in an environment with multiple sources of uncertainty, the completely reactive scheduling approaches are best. With MAS being the best approach, at least for this given problem. This is true even at low levels of uncertainty but becomes more pronounced with larger uncertainty.

4.8 Summary of Experiments on Multi-Agent System Approach Performance

To investigate the performance of the MAS approach to dynamic scheduling that we propose, we conducted four separate sets of experiments whilst considering four different demand mixes. Whilst each set of experiments has its own unique results and findings, there are some common observations that appear consistent throughout each. In this subsection, we will highlight the general observations.

Looking at the manufacturing system we present, we see that the processing route options for each part utilize the same operations and follow similar sequences. Also, the machines within the system that perform the same operation are similar in their ability to process the operation (same setup and processing times). This allows for a system where the greedy option of choosing the shortest path to the next operation to yield the best result. It is important to understand this, as it highlights why the MAS approach proposed is capable of consistently solving for the optimal solution despite not considering future states of the system and possible downstream implications of agent decisions. We feel that the manufacturing system we present is representative of real-world manufacturing systems.

For each experiment, we see that our MAS approach outperforms the alternate dynamic scheduling approaches for the first demand mix scenario. The only exception to this finding was for the experiments where there is no uncertainty or disruptive events to the system. For this set of experiments, the results indicate that the MAS performs as well as the alternative dynamic scheduling options (it yields the shortest time possible). The question is why is this scenario unique? With scenario 1, as we examine the processing routes for each part requested in the demand order for the near-optimal schedule (as provided by the GA in Figure 4.4) we see that one multi-purpose CNC machine would be free after the first operation it completes. This adds flexibility to the system in the form of an alternate routing operation for each downstream operation. This implies that the MAS can now use the additional flexibility this machine provides to minimize the impact of the any disruption to the system. With the subsequent scenarios, we see the CNC machines have significantly heavier workload relative to the first. For these scenarios, the MAS performs comparably to the alternate dynamic scheduling options.

In a more realistic set of experiments, where multiple sources of uncertainty are considered simultaneously, we see that the compounding effect of the sources of uncertainty are more significantly deleterious to the performance of the right-shifting approaches than to our MAS approach. This suggests that the MAS approach we propose may be the best approach to employ in a real-world manufacturing system.

In summary, our findings imply that our MAS approach is particularly effective when duplicate machines are available for operations that commonly occur downstream. However, if there are no duplicate machines for these downstream operations, the results of our experiments indicate that our MAS approach will either outperform or perform comparably to right shifting a nominal schedule so long as some job-related or resource related uncertainties in the system are considered.

Chapter 5

Machine Resource Deployment for Multi-Agent System Based Manufacturing System

In the previous chapter, the results of our simulation experiments suggested that the manufacturing system performance is affected by three factors: (1) the machine time to failure (MTTF) for each machine, (2) mean time to repair (MTTR) for each machine, and (3) the level of uncertainty with setup and processing times. The results of the numerical experiments in the previous chapter suggest that these three factors can be influenced by the flexibility of the system. The flexibility, as we defined it here, has to do with the number of machines available to perform a given operation at any given time. This is a variable that can be easily adjusted or controlled by facility planners by adding more machines so long as the budget for designing the facility allows. There are several questions to answer. The facility planner must determine the selection of machines to be used in the system. They must also determine the number of duplicates of these machines to deploy in the system. This requires an understanding of the point at which an additional duplicate machine of a given type will have no improvement to system performance. As such, the facility planner must understand the significance of the trade-off between robustness offered by having duplicates, and the cost (capital and operation costs) of having these duplicates.

This chapter will focus on strategies for deciding how many distinct machines to have within a system as well as how many duplicates of each distinct machine are needed for effective implementation of a MAS-based smart manufacturing system. This problem is a machine selection problem combined with the machine duplication problem. We will refer to this problem as a machine deployment problem (MDP).

With the MDP, we are tasked with determining which types of machines will be used in the system as well as the number of each of these types of machines that will be used. This selection must be made from a given catalogue of machines. This selection of machines must also result in a manufacturing system designed to process all the part types requested of it. As with most design problems, the design must not violate a set of constraints. With the MDP, these constraints are usually budget and/or space related (maximum number of machines that the space can hold).

There are two possible applications of the MDP. The first is in the design of completely new systems. This would be the case when a new facility is to be constructed and the facility planner needs to decide what machines to use in the manufacturing process. The other application is when upgrading an existing system by deciding which machines are bottlenecks in the system and require duplication in order to improve overall system performance.

In the upcoming sections, we will present two approaches to solving the MDP in a MAS-based smart manufacturing system. The first involves the use of conventional robust design methodology with simulation as the basis for a framework for our decision-making. This approach is particularly useful when assessing the value of adding more resources to an existing system. The second is a simulation-based optimization model for solving the machine deployment problem. This approach is useful when designing a system in its entirety.

This chapter is organized into four (4) sections. In the first section, we will provide an overview of the MDP. In the second section, we will present our robust design methodology-based framework for solving MDP. This section will not only include an overview of the procedure, but also numerical experiments to demonstrate its application. The third section will focus on the formulation of a simulation-based optimization model for determining the number of each distinct machine type to deploy when designing a robust smart manufacturing system (SMS). Finally, in the fourth section, we will conduct a comparison study for the MDP solution methods that were presented in this chapter (robust design, meta-heuristic, and exhaustive search approaches). The objective behind this comparison study is to identify the benefits and drawbacks of using each approach when solving the MDP.

5.1 The Machine Deployment Problem

The machine deployment problem (MDP) is not an entirely new problem. It is similar to the machine selection problem (Wang et al., 2000), machine requirements problem (Miller & Davis, 1977), and the equipment requirements problem (Kusiak, 1987). All of which are problems concerned with determining the appropriate selection of machines. There are also aspects of the MDP covered with the machine duplication problem.

With the MDP, a subset of machines must be selected from a catalogue of machines for use in a manufacturing system. Determining the machines that comprise this subset of machines is the main decision with the MDP. This subset of machines must allow the manufacturing facility to be able to fulfill the processing requirements for producing a set of parts. The decision on this selection of machines is typically limited by the available budget, the catalogue of machines to select from, and the space within the facility. The intention behind this MDP is to design a manufacturing system with good performance with respect to the common manufacturing metrics. These metrics are typically related to order completion time, flowtime, makespan, tardiness and lateness. In this section, we will present our model for the MDP.

5.1.1 Proposed Machine Deployment Problem Model

In this research, the MDP is focused on determining the selection of machines that minimizes the mean order completion time required to fulfill the demand as well as the standard deviation in that completion time. We are tasked with determining $MCH = [mch_1, mch_2, \ldots, mch_{nm}]$. Where *MCH* represents the number of machines of each type in the design. For example, $mch_1 = 2$, implies that there are 2 units of machine type m_1 in the system. It is assumed that we are given $O = \{o_1, o_2, \ldots, o_{no}\}$ which represents the operations that a manufacturing system must be able to perform. We are given a catalogue containing *nm* different types of machines to select from, the cost of each machine type (c_i), and a budget (B). Each of these machines has the capacity to perform a set of operations, $OM(m_i)$. Where $OM(m_1) = \{o_1, o_3\}$ would indicate that machine m_1 is capable of operations o_1 and o_9. It is also assumed that we have the forecast part demand volumes for the system. This information is all fed into a simulation which provides the mean completion time and standard deviation data used to evaluate our solutions.

Decision Variables

We consider one decision variable in the MDP, $MCH = [mch_1, mch_2, \ldots, mch_{nm}]$. This is the combination of machines being used in the system. This variable consists of the number of each machine type that will be used in the system. It can be represented by a string of integers with values greater than or equal to zero. The length of this string is the number of machines available to purchase, *nm*. The assignment of values to the elements of this string is limited by the available budget for procuring machines as well as the number of each machine type available to procure, and the operations required of the manufacturing system. The product of all possible elements of this string would represent the solution space of the MDP.

For example, a facility planner can purchase four types of machines; mch_1, mch_2, mch_3, and mch_4. The facility requires operations o_1, o_2 and o_3. Operation o_1 can be performed by machine mch_1 and mch_2. Operation o_2 can be performed by machine mch_3. Operation o_3 can be performed by machine mch_4. The facility planner has a budget of \$1000, and each machine of any of the four available types of machine costs \$300. The supplier has adequate units of each type of machine, such that the availability is not a constraint. For this example, the facility planner decides on one of each of the following machines; mch_1, mch_3, and mch_4. This is because the facility planner can only purchase three machines due to budget constraints. As such, due to the operation requirements, they selected machines mch_1, mch_3, and mch_4 because it did not exceed the budget, but allowed the system to be capable of all operations. A string that could be used to represent this solution is MCH = [1 0 1 1].

Modelling Assumptions

In our development of the MDP we make the following assumptions regarding the problem:

1) The part demand is known
2) There is sufficient space within facility to contain any feasible design solution
3) Machines are organized into equidistant functional layout departments
    a. Distance between machines in a department are negligible
4) The manufacturing system is subject to multiple sources of uncertainty simultaneously
5) All sources of uncertainty or disruption to the system are known and can be represented in the form of a known distribution

Mathematical Model Formulation – Performance Metrics

With our MDP, the most important metrics by which the quality of the solution will be evaluated are the total cost of the machines to be utilized in the system and the mean time required to complete the order for parts received by the system. With respect to the mean completion time, we focused on the opportunity cost related to missing due dates (*DD*). To assign monetary value to the opportunity cost we must specify a conversion factor (β). It is important to try to keep both the cost of the design and the potential opportunity costs as low as possible. However, it is also important that the solution be robust. As such, the quality of the solution will also be dependent on the standard deviation (σ_s) in the performance of the system. These three metrics are represented as follows:

Metric 1: The total design cost (TDC) for implementing the design (smaller being better)

$$TDC(MCH) \tag{5.1}$$

where

$$MCH = [mch_1, mch_2, \dots, mch_{nm}] \tag{5.2}$$

$$TDC(MCH) = \sum_{i=1}^{nm} mch_i c_i \tag{5.3}$$

Metric 2: The costs (or penalties) associated with instances of tardiness of the jobs (smaller being better)

$$\beta(\bar{y}_s(MCH) - DD) \tag{5.4}$$

where,

$$s = \left(p_1, p_2, \dots, p_{np}\right) \tag{5.5}$$

$$\beta = \begin{cases} 0, & if\ \bar{y}_s(MCH) - DD < 0 \\ \beta, & if\ \bar{y}_s(MCH) - DD \geq 0 \end{cases} \tag{5.6}$$

Metric 3: The standard deviation with respect to mean order completion time

$$\sigma_s \leq \sigma_{max} = 10^{-\frac{\delta}{20}}(DD - \alpha) \tag{5.7}$$

Equation 5.7 shows the desired relationship between the desired completion time (*DD*) and the standard deviation (σ_s). Equation 5.7 also allows the incorporation of the amount of slack time (α) desired. The equation allows for a selection preference for combinations that are close to target with decreased noise. If there is no due date for the order, the target can simply be set to the mean completion time.

Mathematical Model Formulation - Model Constraints

Our MDP is subject to two main constraints. These constraints are as follows:

1) The selection of machines must result in a system capable of all operations required to fulfill the demand. Let *sm* represent the indices for selected machines, such that $sm = \{ i \mid mch_i > 0\}$. Given *sm*, the constraint associated with the system capacity is shown in Equation 5.8.

$$O \subseteq \bigcup_{i \in sm} OM(m_i) \tag{5.8}$$

2) The cost associated with implementing the system must not exceed the budget (*B*) allotted. The total cost of implementing the system is dependent on the cost of each

machine (c_i) and the number of each machine type selected (mch_i). Equation 5.9 shows this constraint.

$$\sum_{i=1}^{nm} c_i mch_i \leq B \tag{5.9}$$

5.2 Robust Design-Based Framework for Solving Machine Deployment Problem

The robust design method is a very commonly used approach in manufacturing for improving the function of products, systems, or processes. In our application, we will be using it to solve the MDP. Our objective in solving the MDP using robust design methodology is to design a system that is relatively insensitive to stochastic disturbances. The system disturbances we focus on are variations in operation execution duration, machine breakdown and repair, and demand volume fluctuation. It is important that the machine resources present in the system be sufficient to minimize the effect of these disturbances without the cost of acquiring these machine resources exceeding the allotted budget. In this section, we will present an outline of how to solve the MDP using a robust design approach.

5.2.1 Robust Design Procedure

In this section we will provide a procedure for using robust design methodology to solve the MDP for a smart manufacturing system. The procedure is outlined below:

1) *Define System Input Information*: The system inputs can be separated into two categories; (1) deterministic factors, and (2) stochastic factors. Deterministic input factors consist of the factors that we consider to be known and fixed throughout the production cycle. In this model, this includes the demand volume and mix, the order arrival time, and the budget. Stochastic input factors represent the sources of uncertainty within the system. In

our model, these factors are the mean time to failure (MTTF) and mean time to repair (MTTR), as well as the variability in setup time and processing time for each operation. These factors are represented using distributions. We use exponential distributions for each of these factors based on the conventions shown in literature relating to machine reliability (Kececioglu, 2002).

2) *Design and Execution of Experiments*: Each distinct simulation experiment begins with a solution (a set of machines that make up the manufacturing system). This solution (set of machines) is represented by $MCH = [mch_1, mch_2, \ldots, mch_{\mathrm{nm}}]$. There is usually a constraint on the values that can be assigned to elements of *MCH*. These dictate the size of the solution search space.

   To avoid searching the entire solution search space we limit the experiment to searching combinations of machines that only consist of high, medium, and low levels of each machine type. This is a three-level experiment design. We chose this as it allows us to model possible curvature in the system response between two extreme points. The decision on the values level settings for high, medium and low for each element of MCH are based on the lowest and highest permissible values for each element. Our simulation experiments are now reduced to running all combinations of the elements of *MCH* for these three levels. This is a full factorial, three-level experiment design.

   From our experiments we record three pieces of information, (1) the mean completion time, (2) the standard deviation in the completion time, and (3) the cost of implementing the solution. This information is used to determine the main effects of the level each element of *MCH* on the system cost, mean completion time and standard deviation.

3) *Determining Solution (Analysis of Results)*: The decision on the final solution is made by following these steps:

    a. Rank the machines (using the main effects) according to those that have the most significant effect on the mean order completion time to least. This would be reflected in the slope of the effect for each machine as their level changes. The steeper the slope, the more significant the effect.

    b. Repeat the previous step for the standard deviation.

    c. Examine the results to find the solutions that yielded mean completion times, standard deviation, and implementation costs within the acceptable thresholds.

    d. If there is a single result that has both the lowest mean completion time and standard deviation:

        i. Using the rankings for main effects, determine if an additional machine can be added to improve system performance without violating budgetary constraint. Add a machine to the system with the most significant effect system performance. With machines that have most significant effect on completion time given more priority. Note, that the center point in the effects plot should be used to gauge if there is utility in adding more of a machine type

        ii. If two or machines are similar machines (they serve as alternative options), use rankings of the main effects to determine if one or more of the similar machines can be replaced with the better alternative or removed without significant impact to system performance

iii. Repeat previous step until there are no options to add or subtract machines. The result is the final solution

e. If there is no clear best solution from the feasible solutions. The solution can be built using the main effects for each machine. To design this solution, we execute the following steps:

i. Using the main effects, assign the minimum number of machines required to provide the capacity to perform all needed operations. The machines should be selected over alternatives based on the ones that yield the lowest mean times.

ii. Add an additional unit of a machine with the most significant impact on completion time which can be added given the remaining budget.

1. If there are two machines with similar completion times, make a decision based on the standard deviation

iii. Repeat previous step until there are no options to add machines. The result is the final solution.

5.2.2 Demonstrative Example

In this subsection, we are continuing with the furniture manufacturing system presented in chapter 4. The simulation experiments conducted as part of this demonstration were run using in-house script developed in MATLAB R2021a. We will be examining two design problems and using the robust design procedure outlined in the previous section to solve these problems. These problems will focus on the use of the robust design procedure to determine how to upgrade a facility by the addition of new machines. However, note that the same procedure can be used for designing a completely new facility. In this problem, the system consists of five distinct types of machines:

1. One-sided edging machine (number of machines: 1)
2. Two-sided edging machine (number of machines: 3)
3. Cutting Machine machines (number of machines: 2)
4. Drilling machines (number of machines: 3)
5. Multi-purpose CNC machines (number of machines: 2)

We are exploring two (2) distinct MDP's that stem from two different demand forecasts. Our objective is to determine the appropriate combination of machines to deploy for each different demand forecast if the total available budget is $13,000. The facility planner has decided that additional 2-sided edging and/or CNC machines may be needed to improve system performance as both machines appear to be needed in most of the processing routes for each part the system can produce. We must now determine how many duplicates of either machine we should add to the system in both scenarios.

For the experiments, we assume that operator availability limits the facility to a maximum of five CNC machines and a maximum of six 2-sided edging machines. These values will serve as the high settings. The low setting is based on the original problem design showing a system with two CNC machines and three 2-sided edging machines. The medium level setting is simply the halfway point between these two settings. Table 5.1 shows the level settings for each variable. The experiment design is a full factorial (3^2) with 100 repetitions of each experiment.

Table 5.1 Level Settings for Control Factors

LEVEL	NO. OF CNC MACHINE DUPLICATES	NO. OF 2-SIDED EDGING MACHINE DUPLICATES
LOW	2	3
MEDIUM	3	4
HIGH	5	6

With this problem, the facility planner is concerned with accounting for varying setup and processing times, and machine reliability in the design. The variation in setup and processing times are determined based on the planner's knowledge of the operators' skills as well as from experience with similar systems. For these experiments we use the setup and processing times for operations outlined in chapter 4. The machine reliability information is provided by the manufacturers of each machine. We assume that all machines in the system have the same machine reliability. Table 5.2 shows the settings for the stochastic factors.

Table 5.2 Stochastic Factors for Numerical Experiments

VARIABLE SETUP & PROCESSING TIMES	MTTF	MTTR
Distribution: Exponential Mean: provided in chapter 4	Distribution: Exponential Mean: 6	Distribution: Exponential Mean: 5

The facility planner is tasked with presenting design solutions for two different scenarios. There are three deterministic variables for these scenarios. These are the demand mix of parts, the arrival times of these parts, and the cost of each unit of the two machine types. These are shown in Table 5.3 and Table 5.4 respectively.

Table 5.3 Demand Mix for the Machine Deployment Problems Being Solved

Scenario	Part Time	Arrival Times
1	$\{p_1, p_3, p_5, p_{12}, p_{13}, p_{14}, p_{15}\}$	T = {0, 0, 0, 0, 0, 0, 0}
2	$\{p_1, p_3, p_5, p_7, p_{12}, p_{12}, p_{13}, p_{13}\}$	T = {0, 0, 0, 0, 0, 0, 0, 0}

Table 5.4 Costs Associated with Deploying Duplicates

COST OF EACH CNC MACHINE ($)	COST OF EACH 2-SIDED EDGING MACHINE ($)
2000	1000

5.2.3 Results for Scenario One

The following subsection contains the results of the simulation experiments outlined in section 5.2.2 for scenario 1. In Table 5.5, we present the total cost for each solution option as well as the mean completion times to expect if the solution is implemented and the associated standard deviation. In Figure 5.1, we present the plots for the main effects on the cost, mean completion time and standard deviation when levels of CNC machines vary. In Figure 5.2, we present the plots for the main effects on the cost, mean completion time and standard deviation when levels of 2-sided edging machines vary. This information is also presented in Tables 5.6 and 5.7 respectively.

Table 5.5 Simulation Experiment Results for Scenario 1 Demand Mix

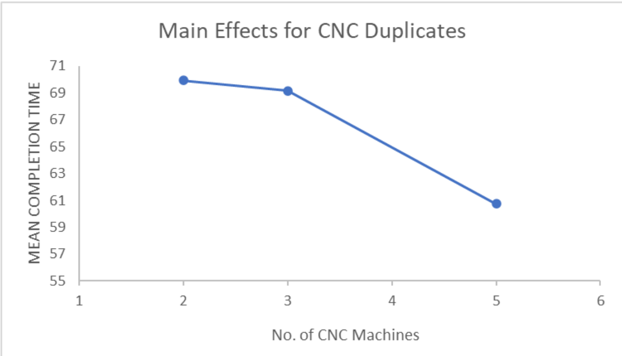
No. of CNC Machines	No. of 2-Sided Edging Machine	Mean Completion Time	Standard Deviation	Total Cost
2	3	75.27	76.80	$7,000.00
3	3	73.07	78.90	$9,000.00
5	3	60.59	64.84	$13,000.00
2	4	65.88	54.34	$8,000.00
3	4	65.12	73.73	$10,000.00
5	4	60.79	54.00	$14,000.00
2	6	68.66	74.53	$10,000.00
3	6	69.29	60.11	$12,000.00
5	6	60.80	54.06	$16,000.00

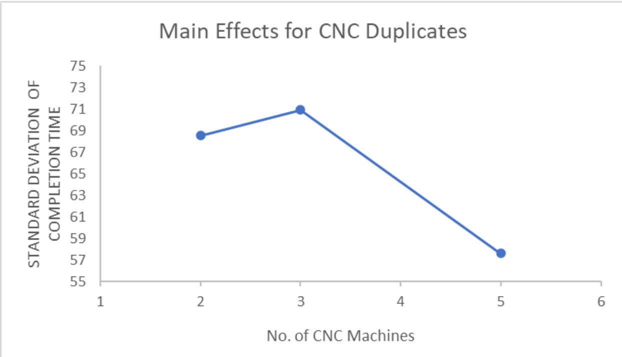
Table 5.5 Main Effect of Number of CNC Machines on Completion Time

CNC	2	3	5	EFFECT
TIME	69.94	69.16	60.73	-9.21
STD	68.56	70.92	57.64	-10.92
COST	$8,333.33	$10,333.33	$14,333.33	6000.00

Table 5.6 Main Effect of Number of 2-sided Edging Machines on Completion Time

2 SIDE EDGING	3	4	6	EFFECT
TIME	69.64	63.93	66.25	-3.39
STD	73.52	60.69	62.90	-10.62
COST	$9,666.67	$10,666.67	$12,666.67	3000.00





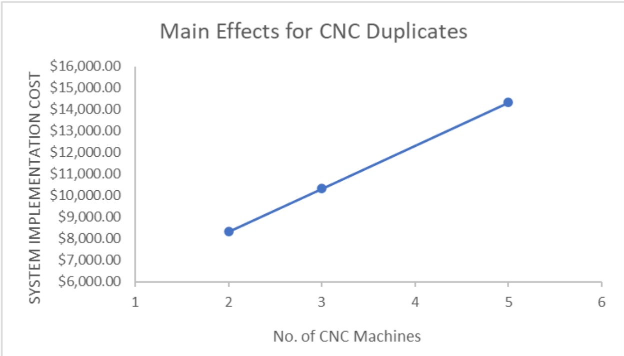


Figure 5.1 Main Effects Plots for CNC Machines for Scenario 1 Demand Mix

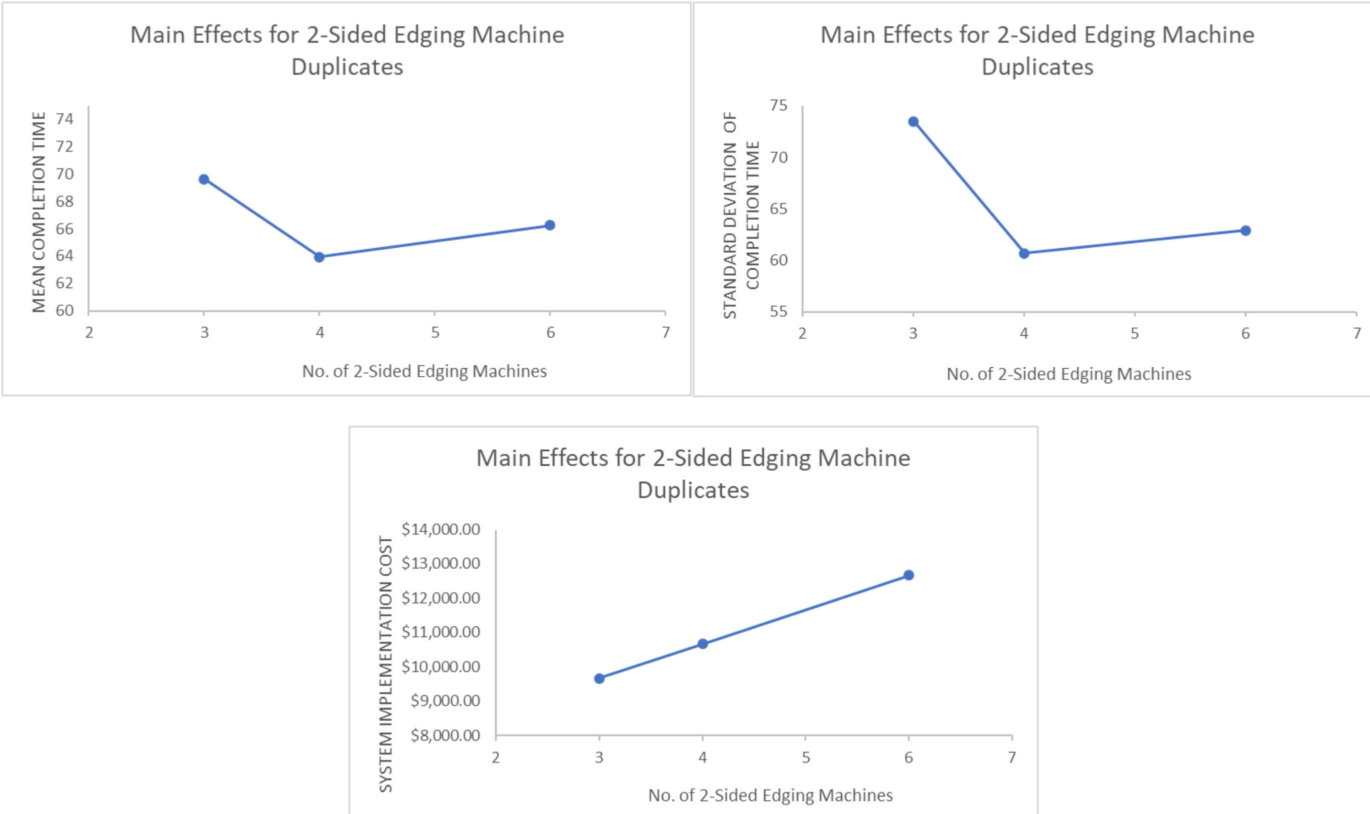


Figure 5.2 Main Effects Plots for 2-Sided Edging Machines for Scenario 1 Demand Mix

Discussion

The results of the experiments for scenario 1 show that there is more value to the CNC machines than 2-sided edging machines. Table 5.5 shows the results of each experimental run for scenario 1. It shows that the lowest completion times (between 60.5 to 61 time units) occur when there are 5 CNC machines. The main effects plots shown in Figure 5.1 for the CNC and Figure 5.2 for the 2-sided edging machines show that the number of CNC machines has the more significant impact on system performance. The specific numerical values for the main effects can be seen in Tables 5.6 and Table 5.7. They show that the effect of the CNC is higher than that for the level of the 2-sided edging machine (-9.21 time units to -3.61 time units). From these results, we can see that the best system performance occurs with five CNC machines and four 2-sided edging machines. This results in both the lowest mean completion time and the standard deviation. However, this solution is not viable as implementing it would exceed the budget of $13,000 (total of $14,000). Given the current information available from the experimental results seen in Table 5.5 and our interpretation of the main effects, we can conclude that the best, feasible solution is to have five CNC machines in the system alongside three 2-sided edging machines (total cost of $13,000). Looking at Table 5.5, we see that the best feasible result occurs with this setting. It has comparable mean completion times (60.59 to 60.79 time units respectively), and but worse standard deviation (64.84 to 54.00 respectively). Looking at the main effects for the number of CNC machines, we see that each additional CNC machine results in significant improvement in the system performance. However, Figure 5.2 shows that the beyond four 2-sided edging machines we see no benefit to additional 2-sided edging machines. In fact, the mean completion times suggest that there might be a detriment to having more of them. However, when the standard deviation at the largest setting for the 2-sided edging machines is considered alongside the mean, we see that there is no statistically significant difference between the two results. From the main effects, we can conclude that the best mean completion time that can result from focusing on maximizing number of 2-sided machines is about ~64 time units. However, the best mean completion time when focusing on maximizing the number of CNC machines is ~61 time units. Both these options have similar standard deviations. As such, we would prioritize the CNC, and select five CNC machines and then use the remaining budget on 2-sided edging machines.

5.2.4 Results for Scenario Two

The following subsection contains the results of the simulation experiments outlined in section 5.2.2 for scenario 2. In Table 5.8, we present the total cost for each solution option as well as the mean completion times to expect if the solution is implemented and the associated standard deviation. In Figure 5.3, we present the main effects plots for the cost, mean completion time and standard deviation when levels of CNC machines vary. In Figure 5.4, we present the main effects plots for the cost, mean completion time and standard deviation when levels of 2-sided edging machines vary. This information is also presented in Tables 5.9 and 5.10 respectively.

Table 5.7 Simulation Experiment Results for Scenario 2 Demand Mix

No. of CNC Machines	No. of 2-Sided Edging Machine	Mean Completion Time	Standard Deviation	Total Cost
2	3	63.03	61.31	$7,000.00
3	3	65.80	58.76	$9,000.00
5	3	56.95	50.18	$13,000.00
2	4	57.83	53.67	$8,000.00
3	4	62.92	54.28	$10,000.00
5	4	67.00	64.86	$14,000.00
2	6	67.66	64.09	$10,000.00
3	6	51.75	58.89	$12,000.00
5	6	53.75	53.99	$16,000.00

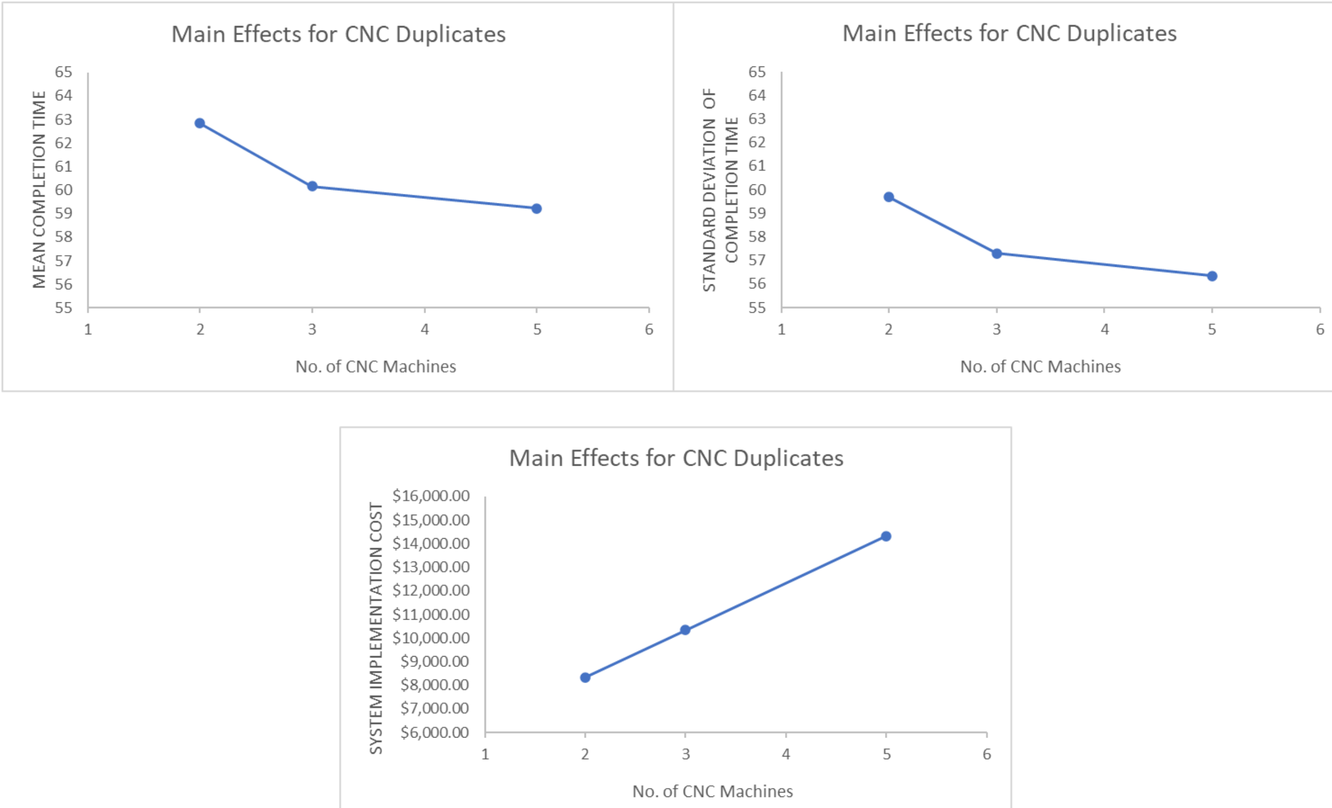


Figure 5.3 Main Effects Plots for CNC Machines for Scenario 2 Demand Mix

Table 5.8 Main Effect of Number of CNC Machines on Completion Time

CNC	**2**	**3**	**5**	**EFFECT**
TIME	62.84	60.16	59.23	-3.61
STD	59.69	57.31	56.35	-3.34
COST	$8,333.33	$10,333.33	$14,333.33	6000

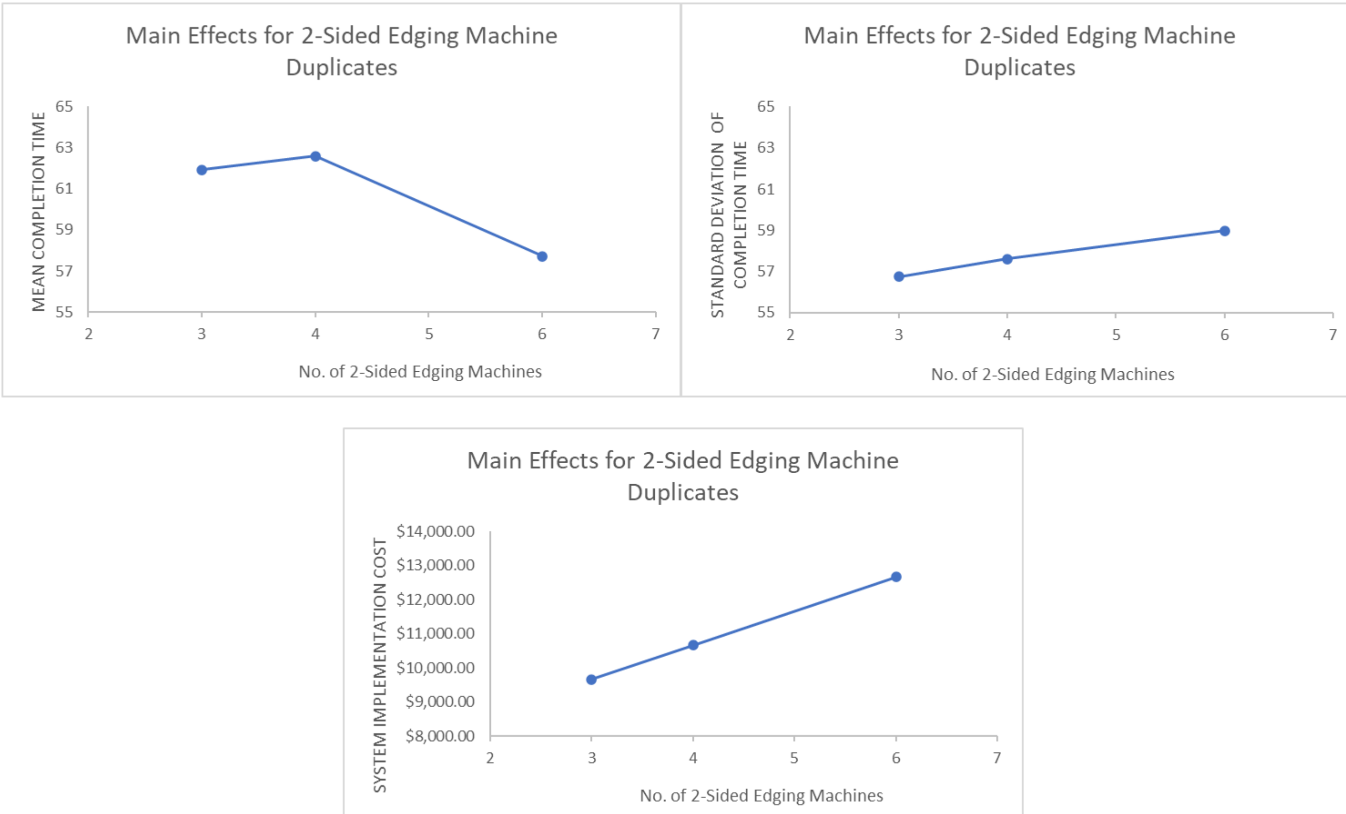


Figure 5.4 Main Effects Plots for 2-Sided Edging Machines for Scenario 2 Demand Mix

Table 5.9 Main Effect of Number of 2-sided Edging Machines on Completion Time

2 SIDE EDGING	**3**	**4**	**6**	**EFFECT**
TIME	61.93	62.58	57.72	-4.21
STD	56.75	57.60	58.99	2.24
COST	$9,666.67	$10,666.67	$12,666.67	3000

Discussion

From the experiments for Scenario 2, we see that there is more value to the 2-sided edging machines than the CNC machines. The results of each experimental run can be seen in Table 5.8. The main effects plots seen in Figure 5.3 seem to indicate that the effect on system performance appears to plateau at 3 CNC machines. However, from Figure 5.4, we see that we continue to see significant system improvement up to 6 2-sided edging machines. The specific numerical values for the main effects can be seen in Table 5.9 and Table 5.10. We can see that the main effect for the additional 2-sided edging machines is greater than that for the CNC. The number of 2-sided edging machines have an effect of -4.21 time units to -3.61 time units for the number of CNC machines. So, we favour having the most 2-sided edging machines as feasibly possible. From the experimental results seen in Table 5.8, the best, feasible option would be to select three CNC machines and six 2-sided edging machines. This results in the best performance of the system without violating the budget constraint of $13,000 (total cost $12,000). As we can see, a change in the demand mix (1 additional unit of parts p_{12} and p_{13} and no parts p_{14} and p_{15}) results in a drastic change in the near-optimal system design from scenario 1. This change is reasonable. By increasing the number of units of p_{12} and p_{13} we increase the need for 2-sided edging machine specifically. By removing parts p_{14} and p_{15} we reduce the need for the CNC specific operation, and as such reduce the need for the CNC machine specifically. In this scenario, the CNC machine would serve more as a duplicate machine to the other machines in the system. It would fill the need when there was one (from any machine in the system). This would result in system performance improving with more CNC's. However, since the CNC can fulfill the need for any operation requested, it can be occupied with other operations than 2-sided edging. Whereas any additional 2-sided edging machines would not. Given that the system has greater need for 2-sided edging specifically, there is more significant effect to having more 2-sided edging machines than the multipurpose CNC machines.

5.3 Genetic Algorithm Implementation for Solving Machine Deployment Problem

In this subsection, we present the mathematical formulation for the MDP objective used when using a genetic algorithm to solve the problem. The MDP model will focus on selecting a combination of machines from a group of possible machines to be used in a system. The selected combination of machines should be the lowest cost option that reliably yields the shortest completion times. This section will contain the optimization model representation for the MDP that the GA will be solving. This section will also contain a description of the GA implementation of the model.

5.3.1 Model Development

In this section, we will describe the decision variables, modelling assumptions, as well as present the formulation of our objective function and our modelling constraints. This section will also provide a description of the solution method we will be employing for solving the optimization problem developed.

Machine Deployment Problem Optimization Model

Our model for the MDP focuses on minimizing the cost of implementing the system as well as the opportunity cost of missing due dates. The metrics for evaluating MDP as presented in section 5.1 have been utilized in the formulation of the optimization problem shown between Equation 5.10 and Equation 5.15. The model is as follows:

$$\min_{MCH} TDC(MCH) + \beta(\bar{y}_s(MCH) - DD) \tag{5.10}$$

s.t.

$$\sigma_s(MCH) \leq \sigma_{max} = 10^{-\frac{\delta}{20}}(DD - \alpha\ ) \tag{5.11}$$

$$TDC(MCH) \leq B \tag{5.12}$$

$$\{mch_i \ \in I \mid mch_i \geq 0\} \qquad \forall\, i = 1, 2, \dots, nm \tag{5.13}$$

$$O \subseteq \bigcup_{i \in sm} OM(m_i) \tag{5.14}$$

where

$$sm = \{\, i \mid mch_i > 0\} \tag{5.15}$$

5.3.2 Genetic Algorithm Implementation

The GA is used to determine the near optimal combination of machines and their respective duplicates for each machine type within the facility. This section will discuss our GA implementation. The GA consists of 5 stages: initial population generation, selection, crossover operation, mutation operation and generation of new population. This sequence of operations is repeated until the solution converges or some alternate stopping criterion is met. Table 5.11 provides an overview of the parametric settings of the GA used in this research. Whilst there are multiple stages to GA implementation, this subsection will only expand upon encoding the chromosome, population initialization, and the fitness function. This is because the other stages of GA are not unique to this problem and have their parameters set based on the suggestions made in Mitchell (1998).

Table 5.10 GA Parametric Settings

Operator	Setting
Crossover Operator	Uniform Crossover
Crossover Probability	0.55
Mutation Operator	Random Setting
Mutation Probability	0.20
Population Size	10 × *n*
Maximum Number of Iterations	100
Number of Simulation Runs	10
Termination Condition	ΔFitness < 0.01
Selection Process	Tournament Selection

Encoding the Chromosome

The problem is encoded such that each gene represents a machine type available to be placed in the facility. A sample chromosome can be seen in Figure 5.5. The value assigned to the allele is the number of duplications for the respective machine. In this example, there are five machine options of which four have been selected for this solution. There are two units of machine mch_1, one unit of mch_2, four units of mch_4, and one unit of mch_5.

M1	M2	M3	M4	M5
2	1	0	4	1

Figure 5.5 MDP Sample GA Chromosome

Population Initialization

At the initial stage of the problem, we need to generate a set of random, solutions to populate the initial generation. In our application, we ensure that the initial population is only filled with feasible solutions. Initially, for each solution we assign each gene a value of zero before proceeding to assign an actual value. We allow for the possibility of not having any of a given machine type. From this point we proceed to build the solution. To ensure that the solutions are feasible and random, we randomly select a locus on the chromosome (a machine type) and randomly assign it to a value that falls between the minimum and maximum values for that machine given its cost and the overall design budget. After this, we deduct this value from the budget and repeat the process for the other loci until all machines have been assigned a value or the budget is completed expended. This is repeated for each solution that makes up the population. A population size of ten times the number of variables is based on a rule of thumb for evolutionary algorithms presented by Storn (1996). The algorithm for generating the initial GA population can be seen in Table 5.12.

Table 5.11 MDP GA Population Initialization Algorithm

Algorithm 1	MDP GA Population Initialization
Input:	budget, cost, nm, no, machine_data
Output:	population

```
Begin
Initialize:  budget, cost, nm, no, machine_data, population = []
for i = 1 to (10×nm) do
        while (1) do
                Initialize: s = (s_1, s_2, …, s_nm), s_j = 0 ∀ j = 1,2,…, nm
                Initialize: M = (1,2,…,nm)
                while (|M|> 0) do
                        r = random integer (between 1 and |M|)
                        m = M_r
                        upper_limit_machine = (Budget/cost_m) rounded to nearest integer
                        set_machine = random integer (between 0 and upper_limit_machine)
                        temp_budget = budget – (cost_m × set_machine)
                        If (temp_budget > 0)
                               budget = temp_budget
                               s_m = set_machine
                               delete (M_r element from M)
                        Elseif (temp_budget = 0)
                               budget = temp_budget
                               s_m = set_machine
                               delete (M_r element from M)
                               Break
                        End
                 End
                 for k = 1 to no do
                         Initialize: operations = [], total_operations = 0
                         for l = 1 to nm do
                               If (s_l > 0)
                                      If (k ϵ machine_data_l)
                                              options = [options, k]
                                      End
                               End
                         End
                         If (|options| > 0)
                               total_operations = total_operations + 1
                         End
```

```
                **End**
                **If** (total_operations = no)
                    **Break**
                **End**
            **End**
            population_i = s
    **End**
    **Return** population
**End**
```

Fitness Function

The fitness of a solution to our MDP is based on the cost of the design and the completion time. The lower the cost and the completion time, the higher the fitness. However, the solution is considered unfit if it violates the constraint that the total cost of the machines selected should not exceed the budget. This should be reflected in the fitness function. The fitness function (*f*) used in our implementation of the GA is shown in Equation 5.16. The value for the mean completion time, $\bar{y}_s$, is determined via simulation. All other terms in the fitness function are deterministic and can be derived based on the proposed solution.

$$fit(MCH) = \begin{cases} \dfrac{1}{TDC(MCH) + \beta(\bar{y}_s(MCH) - DD)}, & TDC(MCH) \leq B \\ 0, & otherwise \end{cases} \tag{5.16}$$

5.3.3 Demonstrative Example

In this section we present a demonstrative example of the implementation of the model presented in section 5.3.1. We solve an MDP using our GA model and compare the results to that of an exhaustive search. In the subsequent subsections we will discuss the details of the problem and present the results of the experiments as well as a discussion of the results.

Problem Description

For our demonstrative example, we will be examining two scenarios for an MDP involving the design of manufacturing system tasked with producing three different types of parts. This manufacturing system is tasked with manufacturing these three distinct parts which arrive at the system at time $T = 0$.

For each scenario, the facility planner is tasked with determining the number of each type of machine to have in the manufacturing system from a group of four different types of machines. There are three distinct types of parts, each with their own operations sequences for their production, and machine specific setup and processing times. The manufacturing system must be capable of performing four distinct operation types each of which can only be performed by one of the four distinct types of machines respectively. Operation o_1 can only be performed by machine mch_1. Similarly, operation o_2 can only be performed by machine mch_2, and so on.

The part operation sequence information for both scenarios is the same. This information is provided in Table 5.13. The machine information for each scenario is given in Table 5.14 for scenario 1 and Table 5.15 for scenario 2. For each machine, the processing and setup times follow an exponential distribution. Tables 5.14 and 5.15 show the means for these distributions. The primary difference between two scenarios is in the processing (*PT*) and setup (*ST*) times for machines when processing part p_3. We make the following assumptions whilst solving the problems:

1. The budget provided is \$4500, and must not be exceeded
2. Machine MTTF and MTTR follow an exponential distribution
3. Part processing and setup times follow an exponential distribution

4. There is sufficient space to fit any number of machines in the facility (space is not a restriction)
5. Transfer times between machines are considered negligible
6. 1 unit of each part is demanded at time zero (T = 0)
7. The due date is considered to be 18 time units (T = 18)
8. The penalty for being late is $100 per time unit ($\beta$ = 100)
9. α is set to 0
10. δ is set to 10

Solutions for each MDP scenario were determined using an exhaustive search. These solutions are provided in Table 5.16. These solutions will serve as the basis of the comparison to our model for model verification.

Table 5.12 Part Operation Sequence

Part	Operation Sequence
P1	O1➔O2
P2	O3➔O4
P3	O1➔O4

Table 5.13 Scenario 1 Machine Information

	MCH1	MCH2	MCH3	MCH4
Operation	O1	O2	O3	O4
Processing (PT) & Setup (ST) Times				
P1	PT: 5 ST: 0.5	PT: 10 ST: 1.0		
P2			PT: 4 ST: 0.4	PT: 3 ST: 0.3
P3	PT: 6 ST: 0.6			PT: 2 ST: 0.2
Reliability Information (μ)				
MTTF	3	5	5	4
MTTR	2	1	1.5	2
Unit Price				
Cost	$1,000.00	$400.00	$600.00	$500.00

Table 5.14 Scenario 2 Machine Information

	MCH1	MCH2	MCH3	MCH4
Operation	O1	O2	O3	O4
Processing (PT) & Setup (ST) Times				
P1	PT: 5 ST: 0.5	PT: 10 ST: 1.0		
P2			PT: 4 ST: 0.4	PT: 3 ST: 0.3
P3	PT: 2 ST: 0.2			PT: 6 ST: 0.6
Reliability Information (μ)				
MTTF	3	5	5	4
MTTR	2	1	1.5	2
Unit Price				
Cost	$1,000.00	$400.00	$600.00	$500.00

5.3.4 Results

Table 5.16 presents the solution obtained for the MDP using both an exhaustive search and our proposed optimization model via a GA.

Table 5.15 Solutions for MDP for Scenarios 1 and 2

Scenario	Solution Approach	MCH1	MCH2	MCH3	MCH4
1	Exhaustive	2	1	1	2
	GA	2	1	1	2
2	Exhaustive	1	1	1	1
	GA	1	1	1	1

5.3.5 Discussion

For this demonstration, we compare our proposed optimization model against the solutions determined using the exhaustive search approach. We examine two similar scenarios involving four distinct machine types to design a system with, and four operations which the system must perform. Both scenarios are similar except for their different processing and setup times for certain operations. This allows for slight changes to the optimal system design, and if our proposed model yields matching solutions, it would imply that the model we proposed tends towards the optimal solution for small, simple problems.

The results of the exhaustive search show that for scenario 1, the best system design is to have two of machine mch_1, one of machine mch_2, one of machine mch_3, and two of machine mch_4. For scenario 2, the exhaustive search shows that the ideal system should have one of each machine type. Using our proposed optimization model alongside GA, we were able to solve both scenarios yielding the same solution as the from the exhaustive search. From these results, the optimization model proposed has adequately demonstrated that it can be used to find the near-optimal solution to the MDP for small problems with good quality solutions.

5.4 Comparison Study

In this section, we will be comparing the results of using the optimization model we proposed in section 5.3 against the results when using robust design methodology, and the results from using an exhaustive search approach. The purpose of this study is to identify the benefits and drawbacks to using either the robust design methodology or the metaheuristic approaches presented earlier in this chapter. The exhaustive search approach will serve as a benchmark for comparing the quality of the solutions yielded using these approaches. The performance of these approaches will be evaluated based on the quality of the solution yielded and the total number of experiments required to arrive at the solution. By solution quality, we mean the option that yields lowest completion times and standard deviations in these completion times without exceeding the budgetary constraints.

In this subsection, we will present a definition of the exhaustive search approach which will be used as the benchmark. After which, we will present an overview of the example MDP to be solved. This section ends with a presentation of the solutions determined using the three approaches being compared, and a discussion of those solutions.

5.4.1 Exhaustive Search-Based Approach to Solving Machine Deployment Problem

The exhaustive search we apply in this section approach simply involves running simulations for each possible solution in the solution search space of the given MDP. From the results of these experiments, we determine the solution that yields the lowest mean completion time with the lowest variation in the results.

5.4.2 Example Problem

The problem to be solved is the design of a manufacturing system that produces four (4) distinct types of parts using three (3) distinct operations. These operations can be performed using a variety of different machines which are provided in Table 5.17. The facility planner is limited to only being able to purchase a maximum of four (4) of each machine type but must remain within the budget of $2400.

Table 5.16 Machine Selection Options for Numerical Experiments

	O1	**O2**		**O3**		
Options	MCH1	MCH2	MCH3	MCH4	MCH5	Budget
Unit Cost ($)	250	405	425	405	450	2400
Max Units	4	4	4	4	4	

Each machine option available to be used has its own unique specific capability and machine reliability information. The part specific processing and setup times for each of these machine options can be seen in Table 5.18. Table 5.18 also shows the demand for each part type the system produces. The operation sequences required to produce the part are seen in Table 5.19. Similarly, the machine reliability information for each machine can be seen in Table 5.20.

Table 5.17 Part-Specific Processing (P) and Setup (S) Times for each Machine Option

	O1	**O2**		**O3**		
Options	MCH1	MCH2	MCH3	MCH4	MCH5	
Part	**Mean Processing (PT) and Setup (ST) Times**					**Demand**
P1	PT: 5.0 ST: 0.5			PT: 4.0 ST: 0.4	PT: 4.0 ST: 0.4	10
P2	PT: 3.0 ST: 0.5	PT: 3.5 ST: 0.3	PT: 2.5 ST: 0.3	PT: 5.0 ST: 0.5	PT: 5.0 ST: 0.5	7
P3		PT: 3.0 ST: 0.5	PT: 3.0 ST: 0.5	PT: 2.0 ST: 0.5	PT: 2.0 ST: 0.5	5
P4	PT: 2.0 ST: 0.2	PT: 2.5 ST: 0.2	PT: 2.0 ST: 0.2	PT: 4.0 ST: 0.4	PT: 4.0 ST: 0.4	15

Table 5.18 Operations Sequences for each Part

Part	**Operation Sequences**
P1	O1➔O3
P2	O1➔O3➔O2
P3	O3➔O2
P4	O1➔O2➔O3

Table 5.19 Machine Option Reliability Information

	O1	**O2**		**O3**	
Options	MCH1	MCH2	MCH3	MCH4	MCH5
MTTF	15	15	20	12	20
MTTR	3	2	2	3	5

The following additional assumptions are made for the problem in order to account for the additional required information:

1. The desired completion time (*DD*) is in 70 time units
2. The penalty for being late is \$30 per time unit ($\beta = 30$)
3. α is set to $0.05DD$
4. δ is set to 10
5. Order is placed to the system at time, $T = 0$

In our evaluation, we will compare the performance of the solutions from the robust design approach and the optimization model presented in this section in order to determine which approach to solving the MDP yields the most robust solution. We define robustness as a system's insensitivity to stochastic disturbances. To that effect, we will be evaluating each solutions' performance based on their mean completion times and standard deviations to determine if the two solutions are equivalent or if one is superior to the other.

With our robust design methodology, we have assigned the number of each machine three levels; a low, mid and high level. The values assigned to these levels can be seen in Table 5.21. Based on these level assignments, we ran a full factorial experiment to determine the effect number of each machine type in the system. After which, we determined the solution based on the constraints imposed upon the system.

Table 5.20 Factor Level Settings Used in Robust Methodology

	O1	O2		O3	
Options	**MCH1**	**MCH2**	**MCH3**	**MCH4**	**MCH5**
LO	1	1	1	1	1
MED	2	2	2	2	2
HIGH	4	4	4	4	4

We will also compare the solutions determined using our MDP model and robust design methodology to the optimal solution to the presented problem as determined using an exhaustive approach. With this exhaustive approach, we will test all possible solutions to the problem and determine the minimum point.

Experimental Conditions

The numerical experiments are run using our own in-house simulation code written on MATLAB R2021b. The optimization problem is solved using a GA that is also implemented using our own in-house code on MATLAB R2021b. Each individual simulation experiment is repeated 20 times.

5.4.3 Results

This section contains the results for the experiments described in section 5.4.2. In this section, we present the solutions obtained using an exhaustive approach, the robust design methodology, and metaheuristic-based approaches for solving the MDP in Table 5.24. These results are first presented and discussed individually. After which they are discussed collectively. Table 5.25 presents the number of experiments required to solve the problem outlined using the three approaches (exhaustive search, robust design, and GA). Lastly, we will present the main effect plots used in the robust design methodology-based approach.

Robust Design Results

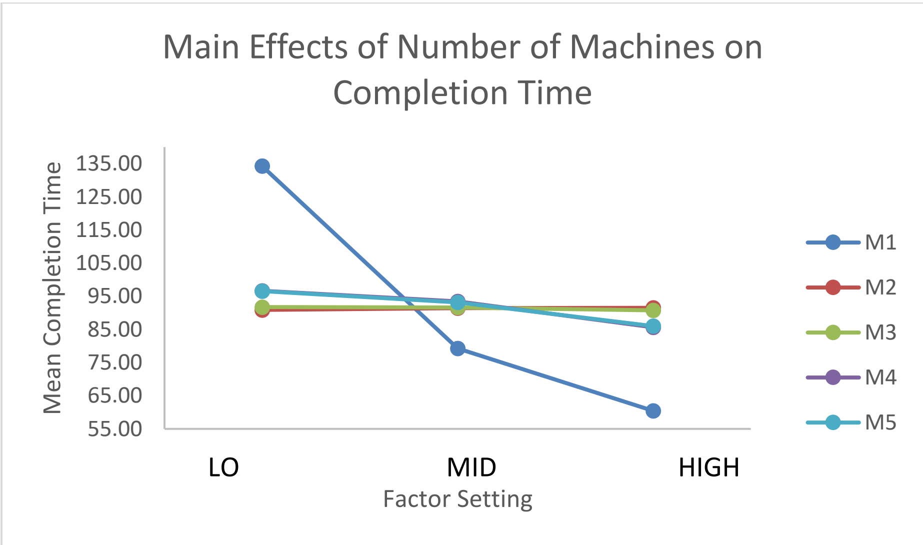


Figure 5.6 Main Effects of the Number of Each Machine Type on the Mean Order Completion Time

Table 5.21 Main Effects of the Number of Each Machine Type on the Mean Order Completion Time

	LO	MID	HIGH	EFFECT
MCH1	134.25	79.02	60.45	-73.80
MCH2	90.61	91.39	91.51	0.90
MCH3	91.74	91.39	90.75	-0.99
MCH4	96.65	93.22	85.66	-10.99
MCH5	96.28	93.12	86.01	-10.27

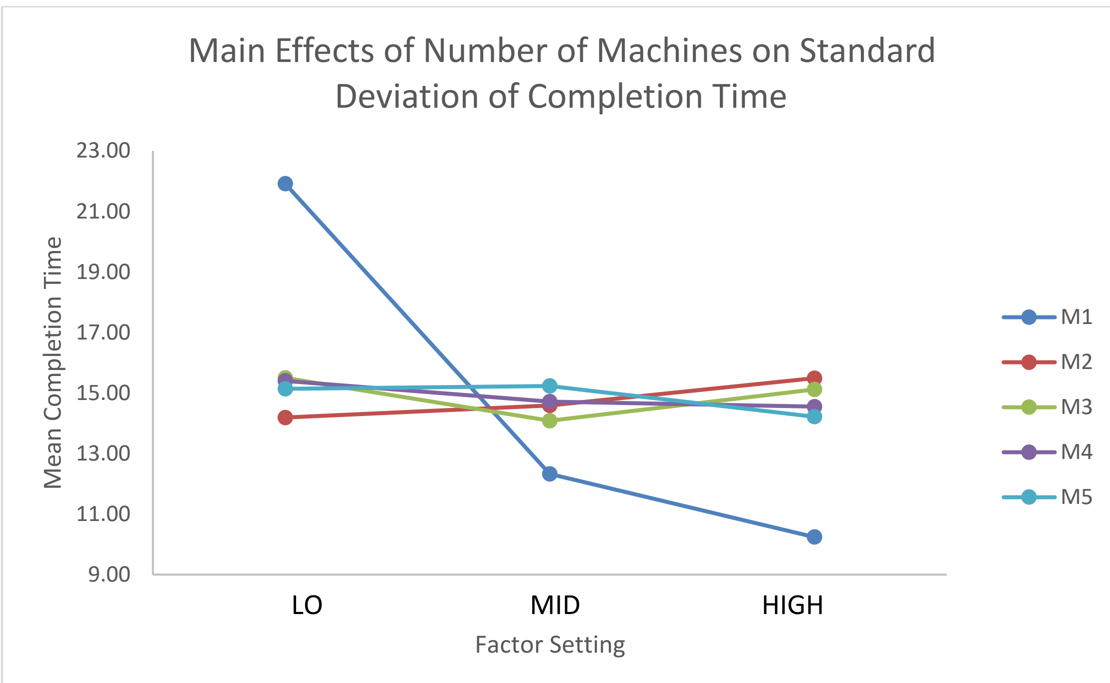


Figure 5.7 Main Effects of the Number of Each Machine Type on the Standard Deviation of the Completion Time

Table 5.22 Main Effects of the Number of Each Machine Type on the Standard Deviation of the Completion Time

	LO	MID	HIGH	EFFECT
MCH1	21.91	12.25	10.24	-11.67
MCH2	14.09	14.59	15.49	1.40
MCH3	15.50	14.02	15.12	-0.38
MCH4	15.40	14.65	14.55	-0.85
MCH5	15.03	15.23	14.22	-0.81

The results using the robust methodology suggest that the “optimal” design for our given MDP is to have 4 units of mch_1, 1 unit of mch_2, and 2 units of mch_4. Using this method, we were able to arrive at this solution using only by examining 243 possible solutions out of 2500 possible solutions. The total cost to implement the system is $2235. As such, the solution does not violate the budgetary constraint and is a viable solution. This proposed solution should be able to complete the order within ~86 time units with a standard deviation of 11.16 time units. This should result in a completion time that is less than 70 time units 7% of the time (z-score ≈ -1.46). As such, there are more instances of incurring the penalty for missed due dates than not. This is not ideal. However, based on the experiments conducted using this method, it is the best solution that does not violate the budgetary constraint.

One of the insights we gain from the robust methodology is that machine mch_1 numbers have the most significant effect on system performance for the given demand. This can be seen in Figure 5.6 and Table 5.22. At the low setting, the average system completion time is 134.25 time units. At the high setting for mch_1, the average system order completion time is 60.45 time units. This has the most significant main effect on the system compared to all other machines whose different settings result in completion times between 96.65 to 85.66 time units. With machine mch_1, we see significant improvement in system performance with increase in numbers of this machine type from low to medium numbers, but there are diminishing returns between medium and high numbers of this machine type. This suggests that machine mch_1 is the major bottleneck in system performance. This is to be expected as operations o_1 and o_3 are the most requested from the system given the high demand for parts p_1 and p_3 relative to p_2. The significance of the main effects of number of machine type mch_1 is the reason for selecting 4 units of this machine type using the robust methodology.

Interestingly, neither machine mch_4 nor mch_5 have as significant an effect on system performance as mch_1. The settings do have an impact on system performance as it improves from ~96 time units to ~86 time units as we move from low to high numbers for either of these machine types. The results show that machines mch_4 and mch_5 are essentially interchangeable. As a result, the smart option is to select the cheaper option which is machine mch_4.

The main effects results show that the performance of the system is relatively unaffected by the number of machine type mch_2 or mch_3. However, we can see that there is slightly more benefit to having mch_3 than mch_2. The results here a interesting given that system performance is shown to get worse as the number of machine mch_2 increases. It is important to understand that mch_2 and mch_3 are similar machines. So, they can be used interchangeably during the production cycle. As such, from the results, we can infer that as there are more machine mch_2, there is more of a chance that mch_2 is used to fulfill operation o_2. Therefore, the results showing the poorer system performance would suggest that mch_2 is a worse option than mch_3 as increasing the likelihood that mch_3 is selected when a part needs operation o_2 results in improved system performance where the alternative leads to worse. That being said, given the low effect on the system performance, the best option when using the robust design approach is to have 1 unit to serve parts requiring o_2. This results in the selection of one unit of machine type mch_3. However, this observation presents a possible area for improvement in the MAS design for part agent (PA) decision-making. The decision process can be improved by incorporating the machine reliability information into PA intelligence.

Genetic Algorithm Results

The results using the GA show that the "optimal" design for our given MDP is to have 3 units of mch_1, 1 unit of mch_2, and three units of mch_4. The total cost to implement the system is $2370. As such, the solution does not violate the budgetary constraint and is a viable solution. This proposed solution should be able to complete the order within ~67 time units with a standard deviation of 10.29 time units. This should result in a completion time that is less than 70 time units 60% of the time (z-score $\approx$ 0.26). As such, there are less instances of incurring the penalty for missed due dates.

The GA was able to arrive at this solution within 250 unique experiments (20 repetitions of each experiment). This drastically reduced the solution search space from 2500 possible solutions.

Exhaustive Search Results

The total number of experiments required to search the entire solution search space is 2500 unique experiments. As a result, it required 50,000 simulation experiments to arrive at the solutions presented in Table 5.25. Whilst this might be more computationally taxing than the other approaches explored, it does allow for us to have higher certainty in the optimality of the solution that is arrived at.

The results using an exhaustive search yield two possible solutions for the "optimal" design for our given MDP. The decision on which is best depends on the manufacturer's sensitivity to the trade-off between implementation cost, and the order completion time (and possible penalties for missing due dates). One is to have 3 units of mch_1, 1 unit of mch_2, and three units of mch_4. The total cost to implement the system is $2370. This proposed solution should be able to complete the order within ~67 time units with a standard deviation of 10.20 time units. This should result in a completion time that is less than 70 time units 60% of the time (z-score ≈ 0.26). This means that the manufacturer would miss the due date ~40% of the time.

The other possible solution is to have 3 units of mch_1, 1 unit of mch_3, and three units of mch_4. The total cost to implement this system is $2390. This proposed solution should be able to complete the order within ~62 time units with a standard deviation of 10.12 time units. This should result in a completion time that is less than 70 time units 77% of the time (z-score ≈ 0.74). This means that the manufacturer would miss the due date ~23% of the time.

Neither of the solutions violates the budgetary constraint and as such, both are viable solutions. The manufacturer is faced with deciding if the additional cost $20 of solution 2 is superseded by the benefit of being 43% less likely to miss the due date. Using Equation 5.17, we can see that in both instances the standard deviation does not violate the threshold constraint ($\delta(\sigma_s) > 10$). This would lead the manufacturer to lean more towards solution 1 given that it is lower in cost, and the threshold of its performance does not exceed their tolerance for risk.

Comparative Summary of Results

Table 5.23 MDP Solutions Using All Three Investigated Approaches

Strategy	MCH1	MCH2	MCH3	MCH4	MCH5	Total Cost	Mean Completion Time	STD
Exhaustive Approach	3	1	0	3	0	2370	67.32	10.20
Exhaustive Approach (Alt)	3	0	1	3	0	2390	62.43	10.12
Robust Design Methodology	4	0	1	2	0	2235	86.31	11.16
Genetic Algorithm	3	1	0	3	0	2370	67.35	10.29

Table 5.24 Number of Simulation Experiments Required to Solve MDP

	# of Unique Experiments	# of Repetitions	Total Experiments
Exhaustive Approach	2500	20	50000
Robust Methodology	243	20	4860
GA	250	20	5000

5.4.4 Discussion

The results of the experiment show that the results yielded from the GA can be the same as the optimal solution as determined through exhaustive search. This can be seen in Table 5.24 which shows that both the GA and exhaustive search to have the same solution. This finding is also consistent with the results for the problem done as a demonstration of the GA model and for the numerical experiment conducted in section 5.3.

The total cost of the “optimal” system is $2370 which is within the budget of $2400. The system should complete the order within approximately 67 time units and does not violate the threshold for $\delta(\sigma_a)$ that is greater than 10. This time is within the 70 time units being targeted for the due date (there is no advantage to being much earlier than the due date). Also, it does not exceed the tolerance for missed deadlines. However, it is important to note that this solution does not yield the shortest completion time. As can be seen in Table 5.24, there is an alternative solution that yields a shorter completion time (~62 time units). This solution is when you have 3 units of machine mch_1, 1 unit of machine mch_3, and 3 units of machine mch_4. However, this option is more expensive in terms of cost. The problem presents us with a trade-off situation, and as such cost must be considered as well as order completion time. With the MDP, each additional machine can improve system performance, but it also increases the cost of implementing the system. It is up to the facility stakeholders to decide the weight assigned to each factor. In this case, their tolerance for variation in the performance means that the lower cost option is the best option.

From our experiments, we see that the exhaustive approach would yield the best results, whilst also offering more options for possible solutions. This would allow the planner to better assess the trade-off between cost and completion time. However, this approach requires a lot computational power and will scale poorly as the number of possible machines to select from increases. Both the GA-based and robust design approaches yield good quality results and are more scalable to problem size. They both required fewer experiments to be conducted in order to solve the problem. For this problem, the GA yielded the same solution as the exhaustive search approach suggesting that metaheuristics may be the best tool to employ for these problems. However, the metaheuristic-based approach does not give much insight into the system or provide alternate options for the planner to consider. As such, the metaheuristic-based approach may be a good option to use to solve large MDP problems effectively, but it is a black box. It would require re-solving the MDP and running experiments again with any changes to problem constraints. For example, if the budget increased, the problem would need to be solved again. However, this would not be the case with the robust design methodology or exhaustive search. With those options, the facility planner would only need to reanalyze existing data.

From the comparison of the results, we also see that the GA can potentially outperform using the robust design methodology approach when solving MDP's. For the problem presented in section 5.4.2, the robust design methodology suggests that the system performs best when you have 4 units of machine mch_1, 1 unit of machine mch_3, and 2 units of machine mch_4. The total cost of this system is $2235 which is within the budget of $2400. The system should complete the order within approximately 86 time units. This solution, whilst cheaper, is not optimal with respect to mean completion time and standard deviation. It would result in missing due dates more frequently than with the results from the GA or exhaustive search. This disparity in the result stems from the design of the experiments used in the robust design. Specifically, the current settings for the levels for each machine type did not allow us to search near the best solution in the solution search space. This is partially due to the nature of the solution search space. Solutions surrounding the near-optimal solution would violate the budgetary constraint (≤ $2400). The decision on the level settings has a direct impact on the observed effects. All in all, this leads to the quality of the solution being poorer when using the robust methodology than the alternative options.

Whilst the robust design methodology may not necessarily always yield the best solution, it has a few advantages over the GA and exhaustive approaches that make it worthwhile to consider as well. This method may require fewer experiments to be conducted in order to arrive at a good quality solution. In this problem, robust design required the fewest experiments to be run. Robust design also allows us to know the main effects of having different numbers of each individual machine type and how the effect that can have on the system performance. This gives the floor manager a clearer idea of which machine types are bottlenecks, and which machine types have sufficient numbers such that additional units would have no value. With this information, alternative options can more easily be explored. This is not the case with the GA approach. Also, the information can be used to upgrade the system in the future without needing to run more experiments. This assumes that the demand placed on the system does not change. However, with the GA and exhaustive approaches, there is insight on how the variables affect the system performance. The problem would have to be rerun to make any changes for the existing system. It is important to note, however, that if the demand changes, the robust design approach would have to be redone as well.

Overall, despite the insights provided, the robust design approach yielded a lower quality solution than the GA. This suggests that the main effects may not always fully represent the system's response to factor settings. It is possible that interaction effects are significant. It is also possible given that the robust methodology we employ solely explores the extremes and midpoints, that system response is not linear between these points. All of which would make it inaccurate to predict system performance using the main effects alone. One of these must be the case with this problem, where the main effects suggest that it is better to have the maximum amount of machine type mch_1 to the detriment of the number of machine type $m\text{ch}_4$ that can be purchased. The GA and exhaustive approaches bypass this problem and can yield better results than the robust approach. However, they do not provide any insight to the system, but each comes with it's own benefit. The exhaustive approach yields the best solution. However, this approach scales poorly with the increasing number of possible machine types, making it less feasible. The robust design and GA approaches scale better than the exhaustive approach. That being said, the GA yields solutions closer to the optimal results as per the exhaustive approach. However, it is a black-box and offers no insights. Nevertheless, the GA provides an excellent for solving large MDP problems with good solution quality.

Chapter 6

Machine Location Assignment for Multi-Agent System Based Manufacturing Systems

In chapter 3, we presented a model for an MAS-based smart manufacturing system (SMS). In chapter 5, we presented a model for determining the machines that would make up an SMS. Both of these chapters address how the SMS would operate and how to determine the resources that would be needed respectively. However, the process of designing an SMS requires an additional step. The last step in designing an SMS is determining the layout configuration for the facility. This chapter will focus on determining how the machines within the facility should be organized relative to each other within the available space so as to maximize the manufacturing system's performance. Determining this layout configuration usually requires knowledge of department planning as well as the application of this knowledge to solving the facility layout problem (FLP) and subsequently, the machine location problem (MLP).

Departmental planning is concerned with determining the appropriate layout type for the manufacturing facility based on demand volume and product variety by taking into consideration the flow, space, and activity relationships. (Tompkins et al., 2001). There are four conventional layout types; product, fixed location, functional and group technology layouts. Hybrid layouts are also possible layout type solutions (Ariafar et al., 2011). These layouts are typically some combination of the four conventional layouts. Each layout type has its benefits and drawbacks. However, the layout type determines the machines that constitute each department. Once the layout type is selected, the facility layout problem needs to be solved. This involves determining the relative locations of departments by assigning to each department to spaces within the facility. Lastly, once the department locations have been decided, it becomes necessary to determine the relative locations of machines within the departments. This is the machine location problem.

In literature, these three problems are typically treated as distinct problems. To the best of our ability, we were unable to find literature that addresses them together. However, they use the same input information (product demand volume and mix, available space and available machines) and they are directly related to each other. They all also influence the material handling costs and the efficiency of the manufacturing system (with respect to cycle time, flow time and transfer times). Solving each problem independently may result in the need to iteratively solve each problem based on the results of the others. This can be problematic especially as the complexity of the problem grows. However, we propose treating the entire problem as one large MLP. It reduces the assumptions made in the layout design. For instance, we do not assume that one predefined layout type is optimal based on given the volume and frequency of part orders. It also allows for unique, and hybrid, layouts which can better allow the system to serve the demand put on it. This is particularly important for mass customization and personalization manufacturing environments where there is no clear best layout to use given the high product variety and high demand as well as the high level of uncertainty from the inputs.

In this chapter, our primary focus is on the development of an optimization model for determining the near-optimal facility layout given a set of machines (including similar and duplicate machines), a set of possible locations to assign them, and known manufacturing environment conditions (part demand, part processing routes, machine information, etc.). Our definition of the optimal layout is one that minimizes the completion time for a given demand order of parts with consistency. In our model, we will allow flexible routing options for each part (sequence, machine, and process flexibility) which can be used interchangeably during the processing run.

This chapter is organized into three (3) upcoming sections. In the first section, we provide a more detailed description of the MLP. This section will present the formulation of our optimization model for solving the layout design problem. The second section will provide an outline of how a genetic algorithm can be used, in combination with simulation, to solve the MLP. Finally, the chapter will conclude with numerical experiments where the performance of the layout derived using our optimization model is compared to conventional layout types under series of different manufacturing environments.

6.1 Layout Design – The Machine Location Problem

The machine location problem (MLP) is intrinsically related to the facility layout problem (FLP). The FLP involves determining the department assignments for machines as well as location assignment for each department within the available space in the facility. The typical objective of the FLP is to minimize the material handling cost (MHC). This is typically done by minimizing the flow, f_{ij}, (load volume per unit of time) and the distance between departments, d_{ij} (Tompkins et al., 2001) given a cost for moving each load from one department to another, c_{ij}, and a predetermined number of departments, *nd*. The typical formulation of the objective is shown in Equation 6.1.

$$min \sum_{i=1}^{nd} \sum_{j=1}^{nd} f_{ij} c_{ij} d_{ij} \tag{6.1}$$

The MLP shares a similar formulation to the FLP. As such, like the FLP, the MLP is an NP-hard problem (Garey & Johnson, 1979) and has a problem solution space in the order of *n*!.

With this problem, we must both determine the spaces to which machines will be assigned in the facility, and the arrangement in which those machines will occupy those spaces. It can be seen as a follow-up to department planning and FLP. Once a department type (or layout type) is chosen and departments are decided, it is important to determine where each machine should be located within the department such that flow through the department is minimized (Chaeib et al, 2001). The machine location problem (MLP) specifically focuses on relative distances between the machines that constitute each department. It can be looked at as an extension of the FLP, in that it focuses on intra-department design whereas the FLP focuses on inter-department design. As such the formulations of the objectives for FLP and MLP are quite similar. The typical objective function is shown in Equation 6.2. Note that in this formulation the flow and distances are now determined based on the two machines in question as opposed to the departments.

$$min \sum_{i=1}^{nm} \sum_{j=1}^{nm} f_{ij} d_{ij} \tag{6.2}$$

The conventional MLP is focused on minimizing the total distance that parts flow through within the system, and in so doing, minimize the time spent being transported for parts in the system. In minimizing transfer times, the total completion time should also be minimized. The conventional MLP uses the distance between departments as a proxy measure for time. However, our MLP model focuses on minimizing the completion time for the order. Using analytical methods, this would not be feasible due to the stochastic nature of the system inputs and its impact on the system's operation. However, we can do this as we use simulation in solving the problem. The key benefit of using this measure is that by focusing on minimizing the time required to fulfill the order, we can find the assignments that are the most robust as opposed to the assignment that only results in the shortest transfer distances. In the rest of this section, we will present our decision variables, modelling assumptions, performance metrics, and MLP model.

Decision Variables

The main decision being made in the MLP is the location assignment for each machine in the system. We assume that the facility floor has been divided up into an r by k grid of possible machine locations. A sample facility grid can be seen in Figure 6.1. This figure shows a facility with nine available locations for machines to be assigned to. Each space on the grid is assigned a number to represent the location. The decision variable with our MLP is the arrangement of the machines on the facility floor. We choose to represent this variable in the form of vector $L = [l_1, l_2, l_3, \ldots, l_{nm}]$. Where L is a string of integers representing the location assignments for each machine in the system. The length of the string is determined by the number of machines, nm, that exist in the manufacturing system. For example, if $l_2 = 3$, it would indicate that machine m_2 has been assigned to location 3 on the facility grid. There is one major constraint on location assignments, and that is, adjacent spaces to a machine location assignment must be empty.

	1	2	3	
	4	5	6	
	7	8	9	

Figure 6.6.1 Sample Facility Grid

Modelling Assumptions

In our development of our MLP we make the following assumptions regarding the problem:

1) All machines that comprise the system are given
2) The demand to the system is known

a. If demand is uncertain, a representative distribution is provided

3) Shipping and receiving spaces are outside of the available space for machine location assignment
4) There is sufficient space within the facility to assign all machines a location whilst allowing access to all machines
5) All sources of uncertainty of disruption to the system are known and can be represented in the form of a known distribution
6) There are sufficient transporters in the system such that there is no wait for a transporter to become free
7) Transportation is immediate. Once a part is done being processed on a machine it immediately begins being transported to the next location
8) There is sufficient buffer space for parts waiting for service

Performance Metrics

The metric we consider to be most important when evaluating the solution of our MLP is the mean order completion time to fulfill a given order mix (represented by s) when using a specific layout configuration (as represented by vector L). The metric is represented by $\bar{y}_s(L)$. This metric needs to be as low as possible without violating any of the problem's constraints.

Alongside the mean order completion time, we are also concerned with the system robustness. As such, we will also be evaluating the quality of the solution based on one additional performance metric, the standard deviation in the order completion time for each solution, $\sigma_s(L)$.

Machine Location Problem Optimization Model

Our model for the MLP optimization problem focuses on minimizing the completion time for processing parts demanded of the system. This completion time must fall within an acceptable threshold, y_{max}. As robustness is also important, the standard deviation in the completion time must also fall within an acceptable threshold, σ_{max}. The layout configuration must not violate the spatial requirements. As such, the distance between any two machines, D_{ij}, must be greater than zero. The equations that constitute the model are shown from Equation 6.3 to 6.12.

$$\min_{L} \ \bar{y}_s(L) \tag{6.3}$$

s.t.

$$\bar{y}_s(L) \leq y_{max} \tag{6.4}$$

$$\sigma_s(L) \leq \sigma_{max} \tag{6.5}$$

$$\{l_i \ \in I \mid l_i > 0\} \qquad \forall\ i = 1, 2, \dots, nm \tag{6.6}$$

$$\{l_i \ \in I \mid l_i \leq (r \times k)\} \qquad \forall\ i = 1, 2, \dots, nm \tag{6.7}$$

$$dm_{ij} > 0 \qquad \begin{array}{l} \forall\ i = 1, 2, \dots, nm \\ \forall\ j = 1, 2, \dots, nm \\ i \neq j \end{array} \tag{6.8}$$

where

$$L = [l_1, l_2, \dots, l_{nm}] \tag{6.9}$$

$$s = \left[p_1, p_2, \dots, p_{np}\right] \tag{6.10}$$

$$dm_{ij} = \omega\left[\left(r_i - r_j\right) + \left(k_i - k_j\right)\right] \tag{6.11}$$

$$\omega = \begin{cases} 0, & \left(r_i - r_j\right) \times \left(k_i - k_j\right) \leq 1 \\ 1, & \left(r_i - r_j\right) \times \left(k_i - k_j\right) > 1 \end{cases} \tag{6.12}$$

6.2 Solving the Machine Location Problem Using Genetic Algorithm

To solve the MLP, we will use an algorithm that combines hierarchical clustering and GA to solve the problem. Hierarchical clustering (HC) will be incorporated into the population initialization phase of the GA and will also affect the mutation operation. The primary purpose of using HC is to group together similar machines and separate dissimilar machines. It will be used to divide the facility grid into smaller sections of similar machines (if possible). We will then proceed to restrict the possible location assignments for these machines to their respective section of the grid. This would effectively reduce the problem's solution space and improve the convergence of the GA. This is the key benefit in using the hybrid of HC and GA.

The GA part of the algorithm is used to determine the near optimal layout of the machines within the departments determined using HC. This section will discuss our GA implementation. The GA consists of 5 stages: initial population generation, selection, crossover operation, mutation operation and generation of new population. This sequence of operations is repeated until the solution converges or the alternate stopping criterion is met. Table 6.1 provides an overview of the parametric settings of the GA used in this research. Whilst there are multiple stages to GA implementation, this section will only expand upon encoding the chromosome, population initialization, fitness evaluation and the mutation operation. This is because the other stages of GA are not unique to this problem and have their parameters set based on the suggestions made in Mitchell (1998).

Table 6.1: GA Parametric Settings

Operator	Setting
Crossover Operator	Uniform Crossover
Crossover Probability	0.55
Mutation Operator	Random Setting
Mutation Probability	0.20
Population Size	$10 \times n$
Maximum Number of Iterations	100
Number of Simulation Runs	10
Termination Condition	ΔFitness < 0.01
Selection Process	Tournament Selection

Encoding the Chromosome

The problem is encoded such that each gene represents a machine that needs to be assigned to a location in the facility. The encoded chromosome looks like a vector containing a set of integers. The value assigned to the allele (locus in the chromosome) is the location to which the respective machine has been assigned as per the facility grid. In this example, there are six machines that must be assigned to the locations on the 5×3 facility grid shown in Figure 6.2. The machines have been assigned to the locations shown in Figure 6.3. Figure 6.4 depicts how the assignments shown in Figure 6.3 would be encoded in the chromosome.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

Figure 6.2 Sample Facility Grid

M1	2	M3	4	M6
6	7	8	9	10
M5	12	M2	14	M4

Figure 6.3 Example Machine Location Assignments

M1	M2	M3	M4	M5	M6
1	13	3	15	11	5

Figure 6.4 Chromosome Encoded Based on Example Machine Location Assignment

Population Initialization

At the initial stage of the problem, we need to generate a set of random, feasible solutions to populate the initial generation. Each solution is in the form of a chromosome with a length that corresponds to the number of machines that need to be assigned a location.

In our application, we ensure that the initial population is only filled with feasible solutions. Initially, we begin by identifying which machines should be grouped together. We do this using hierarchical clustering (HC). The similarity coefficient used to determine the machine grouping is based on reachability for machines based on their machine-part incidence matrix (*MP*). This requires determining a reachability matrix (*RM*) based on the machine-part incidence matrix. If a machine can be reached from another machine, according to the machine-part incidence matrix, then it is considered to be similar by the HC. The similarity coefficient (SC) used is as follows:

$$SC_{ij} = \begin{cases} 1, & if\ RM_{ij} > 0 \\ 0, & if\ RM_{ij} = 0 \end{cases} \tag{6.18}$$

Once the machine groupings have been determined using HC, we now need to split the facility grid up into sections. After this, we assign these sections of the grid to each machine group. To do this, we divide the available spaces into smaller sections based on the number of machines in each group, and the number of machine groups. For example, if there are two machine groups; one group with 4 machines, and the other with 8 machines. The available spaces on the grid would be divided into two sections with one section being twice as large as the other.

Once the space has been divided up, machines from each group can only be assigned to spaces from their respective sections of the grid. This provides us with the values that can be assigned to each locus on the chromosome.

When generating a potential solution, we begin by randomly selecting a locus on the chromosome to assign to a location. This locus represents a machine that needs to be assigned a location. The machine is then assigned to a location randomly based on the spaces available to the machine group to which it belongs. Once this assignment is decided, the space, and spaces adjacent to that space are removed from the available spaces for machines to be assigned. This process then continues until all machines have assigned a location. After which, this entire process is repeated for each solution that makes up the population. A population size of ten times the number of variables is based on a rule of thumb for evolutionary algorithms presented by Storn (1996). The algorithm for generating the initial GA population can be seen in Table 6.2.

Table 6.2 MLP GA Population Initialization Algorithm

Algorithm 1	MLP GA Population Initialization
Input:	rows, cols, nm
Output:	population

Begin
Initialize: rows, cols, nm, population = [], no_of_spaces = rows × cols
Use clustering to determine machine groups
Divide facility grid into spaces to match number of machine groups
Initialize: available_spaces$_j$ ⊆ no_of_spaces; ∀ j = 1,2, …, nm
for i = 1 to (10×nm) **do**
  Initialize: sp = (sp$_1$, sp$_2$, …, sp$_{nm}$), s$_j$ = 0 ∀ j = 1,2, …, nm
  Initialize: free_space = (1,2,…,no_of_spaces), count = 0,
    while count ≤ nm **do**
      Randomly select element of s that has not been assigned location, sp$_k$
      r = random integer (between 1 and |free_space|)
      sp$_k$ = free_space$_r$
      Delete space assigned to s$_k$ and all adjacent spaces from free_space
      count = count +1
    End
  population$_i$ = sp
End
Return population
End

Fitness Function

The fitness of a solution to our MLP is based on the mean completion time, $\bar{y}_s(L)$. The lower the completion time the higher the fitness. However, the solution is considered unfit if it violates the constraints shown in Equations 6.4 and 6.5. This should be reflected in the fitness function. The fitness function (*fit*) used in our implementation of the GA is shown in Equation 6.13.

$$fit(L) = \begin{cases} \dfrac{1}{\bar{y}_s(L)}, & \bar{y}_s(L) \leq y_{\max} \; and \; \sigma_s(L) \leq \sigma_{max} \\ 0, & otherwise \end{cases} \tag{6.13}$$

Mutation Operator

The mutation operator we are using in our GA is random setting. Normally, this would involve randomly setting the value of the selected locus on the chromosome to an acceptable value. In our application, it is similar, however, each machine has been restricted to a set number of spaces that it can be assigned to during the population initialization phase. The randomly set value must fall within this set of spaces excluding spaces that are already occupied by other machines. Also, the new assignment must not violate the spacing requirements (one space between machines). This further reduces the possible random values that can be used. All these restrictions in the set of values used in the random setting mutation ensures that the chromosome that results is a feasible solution.

6.3 Numerical Experiments – Comparison Study

In this section, we will present experiments that compare the performance between a layout derived using our MLP model and two different conventional layout types; functional and cellular. In the upcoming subsections, we will discuss the benchmarks which we will be comparing the designs derived using our model against. We will then proceed to present the details of the different problems that will be the focus of our experiments as well as present the results of the experiments and discussions of the results.

6.3.1 Layout Design Approach Benchmarks

In this research, we will primarily be using two benchmarks; functional layouts and cellular layouts. These are the most commonly studied in literature. The layouts that are generated using our model will be compared against these two approaches to evaluate its solution quality. The metrics for evaluating solution quality are the mean completion times and associated standard deviation for completing the order.

Functional Layout

Functional layouts are best suited for high product variety manufacturing environments with low demand volumes (Tompkins et al, 2001). With high product variety manufacturing environments, there are many different types of parts being requested, each with different operations requirements. As such, it is impractical to use group technology-based layouts. This type of layout design simply requires grouping machines by function. Machines that perform similar operations are grouped together into functional departments. For example, all drills are placed together in a manufacturing system. As such, the number of departments in a functional layout will be dependent on the number of different operations that the facility needs to be able to perform.

With a functional layout, the design primarily involves determining the relative distance of these departments to each other. This can be done using a metaheuristic to solve for the design that minimizes the objective function presented in Equation 6.1. The intention behind the location assignment for each department is the minimization of the transfer time of parts in the system. Each part in the system will flow from department to department based on the operation sequence required to produce it. As such, the location of the departments will directly impact the time required to complete the part. Minimizing the transfer time between departments for parts in the system reduces the completion time for the part.

Cellular Layout – Cell Formation Problem

A group technology-based layout (or cellular layout) is based pairing machines into groups based on part families. Part families are a grouping of parts that share a number of similar operations that require the same set of machines. With group technology, these machines are grouped into departments that are dedicated to that part family. There are many ways to determine machine-part pairings, the most commonly used is direct cluster analysis (DCA) on the machine-part incidence matrix (Tompkins et al., 2001).

DCA requires the use of a similarity coefficient to determine cell formations. The use of similarity coefficients allows for the presence of exceptional parts in the machine grouping process. They also allow for the incorporation of production volumes, operation sequences and operation execution times (Seifoddini & Djassemi, 1995). One of the most commonly used similarity coefficients is the Jaccard similarity which can be seen in equation 6.14. This similarity coefficient assigns similarities to the machines based on how many parts are processed by two machines relative to the parts processed between the two parts. This coefficient is used as the basis for grouping machines using the clustering algorithm with similar machines being more likely to be grouped together and vice versa.

$$SC_{ij} = \frac{a}{a + b + c} \tag{6.14}$$

Where

SC_{ij} similarity of machine m_i to machine m_j

a number of parts that both m_i and m_j service

b number of parts that only m_i services but not m_j

c number of parts that only m_j services but not m_i

6.3.2 Model Performance Evaluation – Example Problem

We will be determining the near-optimal layout for the furniture problem presented in chapter 4. The system consists of 11 machines that can produce 16 distinct part types. For this investigation, we will be exploring a facility grid that has been separated into a 7 by 7 grid of possible location assignments. This facility grid can be seen in Figure 6.5. All machines must be placed into a location on the grid and must be spaced from each other. Transporting WIP between each adjacent point on the grid takes 1 time unit.

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

Figure 6.5 Facility Grid for Furniture Problem Example

All solutions (our solution, the functional layout, and the cellular layout) will use the same part demand, machine capability, transporting times, and facility grid. The cellular layout and functional layouts used in this comparison are modified versions of the layouts presented by Eshragh (2015). With our modifications, we introduce our own facility grid and transportation speeds. The functional layout configuration can be seen in Figure 6.6.

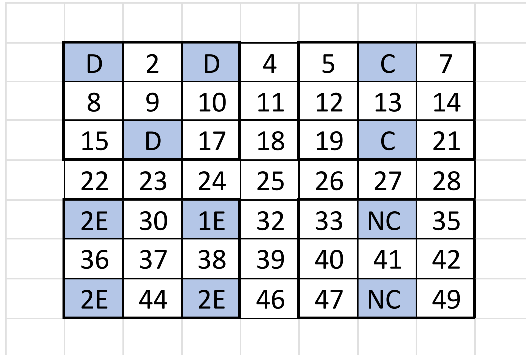


Figure 6.6 Functional Layout for Facility Grid Option

The cellular layout can be seen in Figure 6.7. With the cellular layout, we have chosen to explore two different operating conditions for our numerical experiments. For one set of experiments, we will assume that parts can receive service at any cell given that the machine they need is within the cell. In the second set of experiments, we do not allow for intercellular transfer. For this set of experiments, parts can only receive service at the cell assigned to the part families to which they belong. The specific details for the part families and their cell assignments can be seen in Table 6.3.

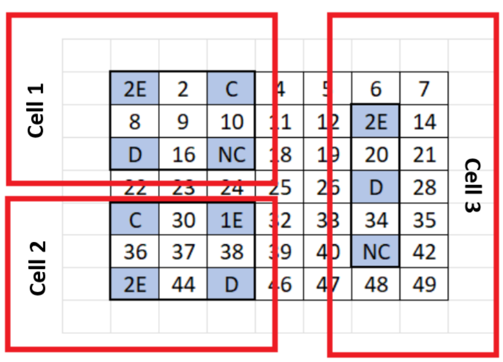


Figure 6.7 Cellular Layout for Facility Grid Option

Table 6.3 Part Part-Family and Cell Assignments

Part Family	Parts Assigned	Cell Assignment
1	$\{p_2, p_5, p_6, p_9, p_{12}\}$	1
2	$\{p_1, p_3, p_4, p_7, p_8, p_{11}\}$	2
3	$\{p_{10}, p_{13}, p_{14}, p_{15}, p_{16}\}$	3

We will examine three scenarios. The difference between each of these scenarios is the part demand mix. The demand mixes can be seen in Table 6.4.

Table 6.4 Demand Information for each Scenario Investigated

Scenario	Parts Demand Mix	No. of Units of Each Part
1	$\{p_1, p_3, p_5, p_{12}, p_{14}, p_{16}\}$	5
2	$\{p_1, p_3, p_5, p_{12}, p_{14}, p_{16}\}$	13
3	$\{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}\}$	5
4	$\{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}\}$	2

For each demand scenario, we calculated the Jaccard similarity for the machines in the system. The Jaccard similarity coefficient is a measure of how similar two machines in a system are given the parts that each machine services. In calculating the Jaccard similarity, we have chosen to ignore the similarity coefficients for duplicate machines. The similarity coefficients for the system's machines can be seen in Figure 6.8 and Figure 6.9.

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11
m1	0	0	0.3	0.3	0.3	0.3	0.2	0.2	0.2	0.5	0.5
m2	0	0	0.3	0.3	0.3	0.3	0.2	0.2	0.2	0.5	0.5
m3	0.3	0.3	0	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.3
m4	0.3	0.3	0.2	0	0	0	0.6	0.6	0.6	0.8	0.8
m5	0.3	0.3	0.2	0	0	0	0.6	0.6	0.6	0.8	0.8
m6	0.3	0.3	0.2	0	0	0	0.6	0.6	0.6	0.8	0.8
m7	0.2	0.2	0.3	0.6	0.6	0.6	0	0	0	0.5	0.5
m8	0.2	0.2	0.3	0.6	0.6	0.6	0	0	0	0.5	0.5
m9	0.2	0.2	0.3	0.6	0.6	0.6	0	0	0	0.5	0.5
m10	0.5	0.5	0.3	0.8	0.8	0.8	0.5	0.5	0.5	0	0
m11	0.5	0.5	0.3	0.8	0.8	0.8	0.5	0.5	0.5	0	0

Figure 6.8 Jaccard Similarity Matrix for Demand Scenarios 1&2

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11
m1	0	0	0.2	0.5	0.5	0.5	0.4	0.4	0.4	0.6	0.6
m2	0	0	0.2	0.5	0.5	0.5	0.4	0.4	0.4	0.6	0.6
m3	0.2	0.2	0	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.3
m4	0.5	0.5	0.2	0	0	0	0.7	0.7	0.7	0.9	0.9
m5	0.5	0.5	0.2	0	0	0	0.7	0.7	0.7	0.9	0.9
m6	0.5	0.5	0.2	0	0	0	0.7	0.7	0.7	0.9	0.9
m7	0.4	0.4	0.3	0.7	0.7	0.7	0	0	0	0.6	0.6
m8	0.4	0.4	0.3	0.7	0.7	0.7	0	0	0	0.6	0.6
m9	0.4	0.4	0.3	0.7	0.7	0.7	0	0	0	0.6	0.6
m10	0.6	0.6	0.3	0.9	0.9	0.9	0.6	0.6	0.6	0	0
m11	0.6	0.6	0.3	0.9	0.9	0.9	0.6	0.6	0.6	0	0

Figure 6.9 Jaccard Similarity Matrix for Demand Scenarios 3&4

We will use our optimization model to determine the near-optimal layout. We will then compare the quality of the solution against the cellular and functional layouts previously presented. This will be done by running simulations for each scenario using the three different layouts. As a result of this set of experiments, we will be able to determine the efficacy of the proposed model as well as determine the potential impact of demand on optimal layout.

Experimental Conditions

The numerical experiments are run using our own in-house simulation code written on MATLAB R2021b. The optimization problem is solved using a GA that is also implemented using our own in-house code on MATLAB R2021b.

6.3.3 Results

This section contains the results of the simulation experiments for the different layouts (cellular, functional and hybrid) for each demand scenario mentioned in section 6.3.2. This can be seen in Table 6.5. In this table we present the mean completion times and standard deviation (both in time units) for each layout under the four different experiment scenarios.

We will also present three sets of figures for each solution determined using the GA and our optimization model presented in section 6.2. Firstly, we present the solution layout. With these figures, duplicate machines have been highlighted with the same color. We also present the relative distances between each distinct machine in the system. Lastly, we present the average shortest relative distance between machine types.

Table 6.5 Order Completion Times and Standard Deviations for Different Layout Options

	Cellular Layout (No Intercellular Transfer)		Cellular Layout (Intercellular Transfer)		Functional Layout		GA Layout	
	Mean	**STD**	**Mean**	**STD**	**Mean**	**STD**	**Mean**	**STD**
Scenario 1	187.99	8.64	129.77	9.49	137.25	25.29	111.87	10.13
Scenario 2	599.48	21.74	353.80	50.46	341.06	28.42	274.62	15.67
Scenario 3	447.47	57.31	302.86	51.03	334.41	21.37	260.71	9.35
Scenario 4	184.60	31.13	123.46	13.72	143.20	38.08	113.23	13.07

Scenario 1 GA Results

8	0	4	0	10	0	0
0	0	0	0	0	0	0
6	0	1	0	5	0	0
0	0	0	0	0	0	0
11	0	9	0	3	0	0
0	0	0	0	0	0	0
0	0	2	0	7	0	0

Figure 6.10 GA Obtained Layout for Demand Scenario 1

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11
m1	0	0	2	3	2	5	3	5	3	5	4
m2	0	0	6	5	3	4	8	11	2	8	7
m3	2	6	0	3	6	7	3	3	5	3	6
m4	3	5	3	0	0	0	4	7	2	2	8
m5	2	3	6	0	0	0	5	7	3	7	4
m6	5	4	7	0	0	0	4	8	6	3	10
m7	3	8	3	4	5	4	0	0	0	7	3
m8	5	11	3	7	7	8	0	0	0	4	7
m9	3	2	5	2	3	6	0	0	0	6	5
m10	5	8	3	2	7	3	7	4	6	0	0
m11	4	7	6	8	4	10	3	7	5	0	0

Figure 6.11 Distance Matrix for GA Layout for Demand Scenario 1

	C	1E	2E	D	CNC
C	0	4	2.5	2.5	5.5
1E	4	0	3	3	3
2E	3.7	5.3	0	3	3
D	3.3	3.7	4.3	0	4
CNC	4.5	4.5	3	4	0

Figure 6.12 Matrix of Mean Shortest Relative Distances Between Machine Types Scenario 1

Using the model presented in section 6.2, we obtain a layout seen in in Figure 6.10. There is no apparent pattern in this layout which could distinguish it as functional or cellular. As such, we consider this layout to be a hybrid layout. Looking at Table 6.5, we see that the hybrid layout results in the lowest mean completion time with 111.87 time-units. The worst time with the cellular layout when no intercellular transfers are permitted (187.99 time-units). The performance of the cellular layout is significantly improved by allowing for intercellular transfer (from 187.99 to 129.77 time-units). The functional layout appears to underperform both in terms of optimality and consistency in performance. It has a completion time of 137.25 time-units, which is the third worst of the layouts examined. It also had the worst standard deviation in completion times, with 25.29 time-units. The most consistency in performance is achieved with cellular layouts. They have standard deviations of 8.64 time-units when no intercellular transfer is allowed, and 9.49 when they are.

Scenario 2 GA Results

0	0	8	0	5	0	9
0	0	0	0	0	0	0
0	6	0	0	0	11	0
0	0	0	10	0	0	0
0	1	0	0	0	4	0
0	0	0	0	0	0	0
0	7	0	3	0	2	0

Figure 6.13 GA Obtained Layout for Demand Scenario 2

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11
m1	0	0	4	6	7	2	2	5	9	3	6
m2	0	0	2	2	8	4	6	9	7	5	6
m3	4	2	0	4	7	6	2	7	9	3	6
m4	6	2	4	0	0	0	6	7	5	3	2
m5	7	8	7	0	0	0	9	2	2	4	3
m6	2	4	6	0	0	0	6	3	7	3	4
m7	2	6	2	6	9	6	0	0	0	5	8
m8	5	9	7	7	2	3	0	0	0	4	5
m9	9	7	9	5	2	7	0	0	0	3	6
m10	3	5	3	3	4	3	5	4	3	0	0
m11	6	6	6	2	3	4	8	5	6	0	0

Figure 6.14 Distance Matrix for GA Layout for Demand Scenario 2

	C	1E	2E	D	CNC
C	0	3	3	4	4
1E	2	0	4	2	3
2E	3.7	5.7	0	3.3	2.7
D	4.7	6	3.3	0	4
CNC	4.5	4.5	2.5	4	0

Figure 6.15 Matrix of Average Shortest Relative Distances Between Machine Types Scenario 2

Table 6.5 shows that for scenario 2 the shortest mean completion times and standard deviations are obtained when using the layout derived using our model (274.62 and 15.67 time-units respectively). As expected, the mean completion times and standard deviations are significantly higher for this scenario than with scenario 1. This is because it is the demand mix but with higher demand volumes. With this scenario, we see that the functional layout now yields the second-best performance. It yields a mean completion time of 341.06 time-units with a standard deviation of 28.42 time-units. Interestingly, the worst performances with respect to the mean and standard deviations are seen when using cellular layouts. The longest completion times are observed when a cellular layout is used with no intercellular transfer allowed (599.48 time-units). The largest standard deviations are observed with cellular layouts when intercellular transfer is allowed (50.46 time-units).

Scenario 3 GA Results

0	6	0	0	0	0	0
0	0	0	1	0	10	0
4	0	0	0	0	0	0
0	0	0	11	0	5	0
0	2	0	0	0	0	0
0	0	0	0	9	0	7
3	0	8	0	0	0	0

Figure 6.16 GA Obtained Layout for Demand Scenario 3

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11
m1	0	0	8	4	4	3	8	6	5	2	2
m2	0	0	3	2	5	4	6	3	4	7	3
m3	8	3	0	4	8	7	9	2	5	10	6
m4	4	2	4	0	0	0	6	6	7	4	6
m5	4	5	8	0	0	0	3	6	2	2	2
m6	3	4	7	0	0	0	10	7	8	5	5
m7	8	6	9	6	3	10	0	0	0	5	5
m8	6	3	2	6	6	7	0	0	0	8	4
m9	5	4	5	7	2	8	0	0	0	3	5
m10	2	7	10	4	2	5	5	8	3	0	0
m11	2	3	6	6	2	5	5	4	5	0	0

Figure 6.17 Distance Matrix for GA Layout for Demand Scenario 3

	C	1E	2E	D	CNC
C	0	5.5	2.5	4	2.5
1E	5.5	0	6.3	5.3	8
2E	3	6.3	0	5.3	4.3
D	4.3	5.3	3.7	0	4
CNC	2	8	2	3.5	0

Figure 6.18 Matrix of Average Shortest Relative Distances Between Machine Types Scenario 3

For scenario 3, we have a similar total number of parts being ordered as with scenario 2. As such, we expect similar completion times to those from scenario 2. The results in Table 6.5

substantiate this expectation. Once again, the mean completion times and standard deviation are lowest when using the hybrid layout derived using our model. This layout yielded a mean completion time of 260.71 time-units with a standard deviation of 9.35 time-units. The cellular layout with intercellular transfer allowed yields the second-best performance with respect mean completion times (302.86 time-units). However, it has the second highest standard deviation in its performance. The functional layout is comparable to the cellular layout with intercellular transfer allowed. It has a longer completion time (334.41 time-units) but significantly lower standard deviation (21.37 time-units). The worst performance observed was with the cellular layout when no intercellular transfers are permitted.

Scenario 4 GA Results

0	0	0	0	0	0	0
0	0	0	0	11	0	0
8	0	7	0	0	0	6
0	0	0	0	0	0	0
0	3	0	1	0	5	0
0	0	0	0	0	0	0
10	0	4	0	9	0	2

Figure 6.19 GA Obtained Layout for Demand Scenario 4

	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11
m1	0	0	4	2	2	2	6	4	2	4	4
m2	0	0	4	8	6	6	3	8	2	8	4
m3	4	4	0	6	2	6	2	8	2	6	6
m4	2	8	6	0	0	0	8	2	6	2	6
m5	2	6	2	0	0	0	6	6	4	2	6
m6	2	6	6	0	0	0	8	2	4	6	2
m7	6	3	2	8	6	8	0	0	0	7	6
m8	4	8	8	2	6	2	0	0	0	6	6
m9	2	2	2	6	4	4	0	0	0	2	6
m10	4	8	6	2	2	6	7	6	2	0	0
m11	4	4	6	6	6	2	6	6	6	0	0

Figure 6.20 Distance Matrix for GA Layout for Demand Scenario 4

	C	1E	2E	D	CNC
C	0	4	4	2	4
1E	4	0	2	2	6
2E	2	4.7	0	2.7	2
D	3	4	4	0	4.7
CNC	4	6	2	4	0

Figure 6.21 Matrix of Average Shortest Relative Distances Between Machine Types Scenario 4

Scenario 4 has a similar number of parts ordered as scenario 1. As such, it was expected that the completion times would be similar. We observe the same pattern in mean completion times as with scenario 1. The hybrid layout yields the lowest mean completion time (113.23 time-units). This is followed by the cellular layout when intercellular transfer is allowed (123.46 time-units). The third-best option is the functional layout with a mean completion time of 143.20 time-units. Similar to the other scenarios, the longest completion times are observed with using cellular layouts with no intercellular transfers permitted. With respect to standard deviations in the completion time, we see that the hybrid layout yields the lowest one. Similar to scenario 1, the highest standard deviation occurs when using a functional layout (38.08 time-units). However, unlike scenario 1, we see that standard deviations in completion time are higher when using cellular layouts if intercellular transfers are not permitted than when they are.

6.3.4 Discussion

Our numerical experiments show that in a MAS-based manufacturing environment with alternate routing options and overlap in part processing routes, a hybrid style layout would be the best layout type to employ. This is based on the results of the experiments showing that the layout determined using the GA resulted in better system performance than the cellular and functional layouts for the manufacturing environment studied. Looking at Table 6.5, we can see that the mean completion times are consistently lower when the GA layout is used in comparison to the alternative layout types. The completion times when using the GA layouts are ~8% to 19.5% lower than the next best conventional layout type employed. With respect to the reliability of the performance, the results indicate that the GA layout tends to result in more consistent system performance than with the alternative layout options. Excluding the first scenario, the GA layout consistently had the lowest standard deviations.

The results of the experiments suggest that the performance of the system is significantly impacted by the layout selection decision. We can also see that having a more hybrid layout may, in some circumstances, be preferable to using the conventional layout types. However, there is a lot of information to consider when making this decision, and this makes the MLP difficult to solve. Determining the ideal layout, especially with all the uncertainties in the system inputs, is made easier by using our proposed optimization model.

Upon initial inspection of the layout designs obtained using the GA, the results may appear random. However, once we look at the machine similarities and compare that to the relative distance of the machines in solution layouts, we begin to see a potential pattern in the solution. For each scenario for which numerical experiments were run, we see that when the average shortest distances between machine types are calculated, the distances inversely correlate with the degree of similarity of the machine types. For example, looking at Figure 6.12, we can see that when we look at the 2-sided edging machine row that the 2-sided edging machines are closest to the CNC machines and the furthest from the 1-sided edging machine. Looking at Figure 6.8, we see that the similarity between CNC and 2-sided machines are the highest and conversely, the lowest similarity for the 2-sided edging machines occurs with the 1-sided edging machines. This pattern repeats itself for all machines and this is true for all demand scenarios tested.

Generally, what we see from the solutions is that each machine in the system is close to at least one machine of a type to which it is highly similar. This results in the average distance from each machine type to another similar machine of a different type being lower. This pattern appears to be consistent regardless of the volume of parts demanded, however appears to change with the demand mix. As we can see from Figure 6.12 and Figure 6.15, the relative distances may be different, but they are in the same order when ranked from closest to farthest for each machine type. However, this is not the case when they are compared with Figure 6.18 and Figure 6.21 where the demand mix is different. This is particularly evident when looking at the similarity between the 2-sided edging machines and the drills. In scenarios 1 and 2 the drills are the highest similarity for the 2-sided machine outside of the CNC machines. However, in scenarios 3 and 4, they are the penultimate similarity. This results in the average 2-sided edging machine being further from drills in the latter scenarios but closer in the prior. This is to be expected as the demand mix has a direct impact on the Jaccard similarity coefficient whereas the demand volume does not.

The demand volume does appear to have an impact on “optimal” location of the machines within the facility grid. We see that as the demand volume for each part type increases without any change in the demand mix (between Figure 6.10 and Figure 6.13, and between Figure 6.16 and Figure 6.19 respectively), there appears to be a distinct change in the pattern of where machines are optimally located. In the scenarios where demand volume is lower, Figure 6.10 and Figure 6.16, we see that cutting and 2-sided edging machines are at the center of the layout and CNC are relegated to the outskirts. However, as the demand volume increases, Figure 6.13 and Figure 6.19, the CNC are moved to a more central location in the layout. This observation can be seen in the corresponding matrix of average shortest relative distances between machine types for each figure. The distance between the average CNC machine and all other machine types decreases with increasing demand. This is to be expected as the CNC is a duplicate for all machines in our experimental setup. So, as the demand increases the CNC machine being used as a duplicate machine is more useful with demand volume increasing for this specific problem, whereas, whilst overall demand is lower, the CNC only performs its own unique operation.

The results of the experiment also suggest that art routing may potentially play a role in the location assignment for the machines in the system. For all demand scenarios studied, we see that the cutting operation is the most common first operation for most of the distinct part types (9 of the 16 distinct part types begin with cutting). As such, it follows that cutting machines should be central in the layout and all other machines that perform downstream operations surround it.

As the total part demand volume increases, the CNC machine becomes more central as more parts require cutting than there are cutting machines to cut as it provides an alternate cutting machine. This is a potential area for further study.

Overall, our experiments suggest that the proposed optimization model and the use of a GA to solve the MLP yields better results than using conventional layout types. The results also suggest that hybrid layouts can in fact outperform conventional layouts. The ideal layout design appears to be strongly correlated to the machine similarities and the potentially the part routing. However, more study is required before making more definitive statements on these observations.

Chapter 7

Conclusion

7.1 Summary

In this research we present a framework for designing a smart manufacturing system. We define a smart manufacturing system to be an extension of the traditional manufacturing system that incorporates more autonomy into the system. The system's components are given decision-making capabilities in scheduling decisions. The system is essentially a cyber-physical production system and there are various ways of designing such a system. To that effect, this dissertation focuses on answering four questions: (1) "What constitutes our smart manufacturing system?" (2) "How does the smart manufacturing system function?" (3) "Under what condition should this system be employed?", and (4) "How should such a system be designed?".

In chapter 2, we examined the literature the encompasses the four questions that are key to this work. The primary difference between the traditional and smart manufacturing system is in the added autonomy of the system's components in making scheduling decisions. As such, chapter 2 begins by examining the scheduling strategies that could be employed. These strategies typically fall into three categories; predictive-reactive, completely reactive, and proactive scheduling strategies. Each strategy has its own benefits and drawbacks, and, as such, has been applied to different manufacturing environments conditions. The subsequent section proceeds to explore which strategies and approaches have been applied when scheduling in various manufacturing environments. The chapter concludes with literature on the design of manufacturing systems from machine selection to layout design.

The focus of chapter 3 was addressing our definition of the smart manufacturing system as well as how our proposed system would function. In this chapter, we present the model we developed for the operation of a smart manufacturing system (SMS). The model outlines what an SMS would consist of, as well as how the system functions. This chapter heavily focuses on how jobs are scheduled within the smart manufacturing system (SMS). The model developed is a multi-agent system model for real-time scheduling in manufacturing environments subject to multiple sources of uncertainty. The model consists of two domains; a physical domain and an agent domain. The physical domain consists of the same components as a traditional manufacturing system. These are parts, machines, transporters, etc. The agent domain consists of the decision-making elements within the system. These are agents for the machines and parts as well as an agent for supervising the interactions between the agents. In our model, we assume that there are alternative routing and operation sequence options, and all these options can be utilized by the system based on the real-time conditions. The model also allows for duplicate and similar machines as well as multi-functional machines. All in all, our model for a smart manufacturing system is designed to be robust so it minimizes the effect of stochastic disturbances to the system on the time required to fulfill orders placed.

The third question this work intended to address was regarding the utility and application of the smart manufacturing system. This was addressed in chapter 4, where we investigated the conditions under which the use of a multi-agent system for scheduling in a manufacturing environment would be beneficial. More precisely, we investigated the performance of different dynamic scheduling strategies when subjected to a single source of uncertainty (at multiple levels) as well as when subjected to multiple sources of uncertainty simultaneously. The examination of the performance of dynamic scheduling strategies subject to multiple different sources of uncertainty simultaneously is relatively novel. Despite there being a large body of research that investigates dynamic scheduling with uncertainty, they mostly focus on singular sources of uncertainty. This study highlights the significance of looking at all sources of uncertainty when deciding the dynamic scheduling approach to employ. The key finding in this chapter is that our proposed model significantly outperforms the alternate scheduling approaches explored in manufacturing environments subject to high levels of uncertainty. This suggests that this type of system would perform well in a mass customization manufacturing environment.

The fourth question this thesis sought to answer is "how should the smart manufacturing system be designed?". This question was addressed in chapters 5 and 6 of the dissertation. In these chapters, we present the models we developed for designing an MAS-based manufacturing system in a manufacturing environment subject to uncertainty. The first model (presented in chapter 5) provided a framework for machine resource deployment decision-making where the types of machines in the system would be decided as well as the number of each distinct type of machine. The second model (presented in chapter 6) provided a framework for deciding machine location assignments within the facility. With the combination of the two models, we are able to design a smart manufacturing system in order to maximize the system performance and robustness. With both models we employ a simulation-based optimization strategy which allows for estimated order completion time to be used in the decision-making process as opposed to the approximations. With our model for solving machine deployment, we found that it performed comparably to an exhaustive approach without the need to search the entire solution search space. With our model for solving the machine location problem, we found the approach to yield better results than with functional or cellular layouts for the problem examined.

Overall, this work successfully outlines a framework for the design of a smart manufacturing system. It also presents the conditions under which such a system would be employed and the benefits of using this system.

7.2 Research Contributions

There are three main areas where this thesis has made contribution to research. The first is in the findings from our investigation into the performance of three (3) dynamic scheduling strategies (right-shifting, dispatching rules, multi-agent systems) when they are employed in manufacturing systems subject to varying levels of uncertainty from multiple sources of uncertainty. The sources of uncertainty examined were variable setup and processing times, uncertain demand, and machine availability (MTTF and MTTR). In our investigation of these factors both separately and together, we provide further insights into the conditions under which each approach would be the best option to employ.

1. It is widely accepted in the literature that predictive-reactive scheduling approaches generally outperform completely reactive approaches. However, our results show that this is dependent on the type of uncertainty and the number of sources of uncertainty the is subject to. When faced with demand uncertainty or multiple sources of uncertainty simultaneously, it is generally better to employ completely reactive scheduling.
2. There are conditions under which these strategies are statistically similar to each other when considering only singular sources of uncertainty.
    a. When the level of the uncertainty is high, there is little difference in the system performance regardless of approach employed
    b. When there is excess capacity in the system (i.e., no bottlenecks and underutilized duplicate or similar machines), there is no difference in system performance

Overall, this collection of data, and the analysis that follows provides information that provides more context to the literature consensus whilst providing insights that are useful in the decision-making for manufacturing systems.

The second contribution is the development of a model for solving the machine deployment problem using simulation-based optimization. By using simulation-based optimization, we eliminate the need for using proxy measures and approximations for the system inputs in our analytical model. Also, the use of simulation allows for accounting for two factors that are too complex to easily capture with analytical models:

1) The interrelationship between the number of machines and the system inputs
2) The interdependence and competition between machines

The problem of determining which selection of machines, and the number of each of the selected machines should be used in a manufacturing system is quite complex. The model we present in this work is a tool for making the design decisions that result in a robust manufacturing system.

The third major contribution is the development of a model for machine location assignment within a smart manufacturing factory. The contribution here is twofold (1) the simulation-based optimization model for designing the facility, and (2) the insights obtained from the investigation into hybrid layouts performance against cellular and functional layouts. By using a simulation-based optimization model, we avoid the use of proxy measures for system inputs in our analytical model. This allows for the final design to yield results that should be more relevant to real world applications. Also, in our numerical experiments comparing the layouts that resulted from our model against functional and cellular layouts we gained insight into the potential for unconventional or hybrid layouts. These layouts may be beneficial to employ for system productivity in certain scenarios. Hybrid layouts are not extensively studied in literature, this work provides data to suggest the potential in further investigation of hybrid layouts.

7.3 Implementation Considerations for Multi-Agent System Based Manufacturing

While implementing a multi-agent system (MAS) in manufacturing can offer numerous benefits, including increased flexibility and efficiency, there are a number of limitations and challenges that must be considered when implementing them in the real world. These can vary based on the unique characteristics of their manufacturing processes and systems. Thorough planning, resource allocation, and expert guidance are essential to navigate these challenges effectively. These considerations are as follows:

1. **Complexity of Implementation:** Developing a MAS for manufacturing entails creating custom software for agents, designing communication protocols, and integrating the system into existing manufacturing processes. This complexity can result in longer development timelines and higher upfront costs than alternative manufacturing systems.
2. **High Development and Maintenance Costs:** Building a MAS requires specialized expertise in a variety of disciplines, which can be costly prohibitive. Ongoing maintenance, updates, and debugging can also strain budgets over time.
3. **Integration Challenges:** Manufacturing facilities often rely on legacy systems that are not necessarily compatible with MAS-based manufacturing. Achieving seamless integration might involve retrofitting or overhauling existing infrastructure, leading to additional implementation expenses.

4. **Data Security and Sharing:** Agents in a MAS need to share data for effective coordination. Ensuring the security of, and consistency in data transmission and storage, especially when dealing with sensitive manufacturing data, demands robust security measures.
5. **Training and Expertise Requirements:** Personnel responsible for operating and maintaining a MAS-based manufacturing system need specialized training in agent-based systems, which might not be readily available.
6. **Robustness and Fault Tolerance:** It is important to ensure that the MAS can continue functioning in the presence of agent failures or disruptions. This requires complex fault tolerance mechanisms and thorough testing, adding to development and maintenance complexity.
7. **Regulatory and Compliance Challenges:** Manufacturing often operates in highly regulated environments. Ensuring that the MAS complies with industry standards and regulations involves careful documentation, validation, and adherence to legal requirements, potentially adding administrative overhead.

7.4 Recommendations for Future Work

The following is a list of recommendations for further study:

1. **Advance the agent intelligences used in the MAS model used for decision-making**. Currently, the part agents and machine agents use and react to the current system status. However, it might be worthwhile to allow these agents to extrapolate the current system data and make decisions based on possible future states. For example, part agents could decide machines for two consecutive operations at a time. Similarly, machine agents

could be designed to assess the current operations available in order to assess which operation would better meet its objective. It could also predict the operations that the part agents would soon be requesting. The machine agent could then assess the current and future state of the system. It could then use that information to determine two things: (1) whether or not to being on an existing operation or wait, and (2) which operation to bid on.

2. **Incorporate transporters into the MAS model**. Currently, it is assumed that there are infinite transporters that move at constant speed without ever interfering with or obstructing each other. As such, once a scheduling decision is made, the part is immediately enroute to its destination machine. However, this is not representative of a real-world system. The current MAS model can be modified to incorporate a transporter agent. This agent would provide the part agents with estimated transfer times to each machine for the agents to use in their decision of which machine to assign work. It would also determine the routes for all transporters once scheduling decisions have been made.
3. **The machine deployment problem and the machine location problem can be combined into one comprehensive problem**. The machine deployment problem is influenced by spatial constraints. The ideal combination of machines for the "optimal" manufacturing system that satisfies the budget constraint may not be practical for the space available to house the system. Also, as demonstrated in chapter 6, the location assignment for machines can have a significant effect on the system performance. As such, there may be benefit in considering the interdependence between number of machines and the layout design in the system design process.

References

Abumaizar, R. J., & Svestka, J. A. (1997). Rescheduling job shops under random disruptions. *International Journal of Production Research*, *35*(7), 2065–2082.

Adenso-Diaz, B. (1996). An SA/TS mixture algorithm for the scheduling tardiness problem. *European Journal of Operational Research*, *88*(3), 516-524.

Adibi, M. A., Zandieh, M., & Amiri, M. (2010). Multi-objective scheduling of dynamic job shop using variable neighborhood search. *Expert Systems with Applications*, *37*(1), 282-287.

Ahmadi, Ehsan, Zandieh, Mostafa, Farrokh, Mojtaba, & Emami, Seyed Mohammad. (2016). A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms. Computers & Operations Research, 73, 56–66. https://doi.org/10.1016/j.cor.2016.03.009

Aloulou, M. A., & Portmann, M. C. (2005). An efficient proactive-reactive scheduling approach to hedge against shop floor disturbances. In Multidisciplinary scheduling: theory and applications (pp. 223-246). Springer, Boston, MA.

Arslan M., Catay B., and Budak E.,(2004), A decision Support System for Machine Tool Selection, Journal of Manufacturing Technology Management, volume 15, Number 1, 101-109.

Aytug, H., Koehler, G.J., Snowdon, J.L., 1994b. Genetic learning of dynamic scheduling within a simulation environment. Computers and Operations Research 21, 909–925.

Barbosa, J., Leitão, P., Adam, E., & Trentesaux, D. (2015). Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. Computers in industry, 66, 99-111.

Barenji, A. V., Barenji, R. V., Roudi, D., & Hashemipour, M. (2017). A dynamic multi-agent-based scheduling approach for SMEs. *The International Journal of Advanced Manufacturing Technology*, *89*(9-12), 3123-3137.

Bean J, Birge J (1986) Match-up real-time scheduling. In: Proceedings of the symposium on real-time optimization in automated manufacturing facilities, national bureau of standards, special publication 724, pp 197–212

Bean, J. C., Birge, J. R., Mittenthal, J., & Noon, C. E. (1991). Matchup scheduling with multiple resources, release dates and disruptions. *Operations research*, *39*(3), 470-483.

Bhaskaran, K., Pinedo, M., 1991. Dispatching. In: Salvendy, G. (Ed.), Handbook of Industrial Engineering. John Wiley, New York. Chapter 83.

Birge J, Dempster M (1995) Optimal match-up strategies in stochastic scheduling. Discrete Appl Math 57(2–3):105–120

Blackstone, J. H., Phillips, D. T., & Hogg, G. L. (1982). A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *The International Journal of Production Research*, *20*(1), 27-45.

Bongaerts, L., Monostori, L., McFarlane, D., & Kadar, B. (2000). Hierarchy in distributed shop floor control. *Computers in Industry*, *43*(2), 123–137

Brennan, R.W., & Norrie, D. H. (2001). Evaluating the performance of reactive control architectures for manufacturing production control. *Computers in Industry*, *46*(3), 235–245.

Buddala, R., & Mahapatra, S. S. (2019). Two-stage teaching-learning-based optimization method for flexible job-shop scheduling under machine breakdown. The International Journal of Advanced Manufacturing Technology, 100(5-8), 1419-1432.

Bukkur, K. M. M. A., Shukri, M. I., & Elmardi, O. M. (2018). A review for dynamic scheduling in manufacturing. *Global Journals of Research in Engineering*, *18*(J5), 25-37.

Bussmann, S., & Schild, K. (2001, October). An agent-based approach to the control of flexible production systems. In *ETFA 2001. 8th International Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No. 01TH8597)* (Vol. 2, pp. 481-488). IEEE.

Cavalieri, S., Garetti,M., Macchi, M., & Taisch,M. (2000). An experimental benchmarking of twomulti-agent architectures for production scheduling and control. *Computers in Industry*, *43*(2), 139–152.

Chang, Ping-Teng, and Yu-Ting Lo. “Modelling of Job-Shop Scheduling with Multiple Quantitative and Qualitative Objectives and a GA/TS Mixture Approach.” International Journal of Computer Integrated Manufacturing, vol. 14, no. 4, 2001, pp. 367–384.

Chao, Y. P., Cheng, S. R., Shih, C. H., Yu, K., Hsu, P. H., Chen, H., & Wu, W. H. (2021). A hybrid heuristic for agent-based scheduling problem with an ordinal scale objective criterion. *Journal of Information and Optimization Sciences*, *42*(1), 109-134.

Chen, C., Yih, Y., 1996. Identifying attributes for knowledge base development in dynamic scheduling environments. International Journal of Production Research 34 (6), 1739–1755.

Church, L. K., & Uzsoy, R. (1992). Analysis of periodic and eventdriven rescheduling policies in dynamic shops. *International Journal of Computer Integrated Manufacturing*, *5*(3), 153–163.

Cowling, P. I., & Johansson, M. (2002). Using real-time information for effective dynamic scheduling. *European Journal of Operational Research*, *139*(2), 230–244.

Daniels, R. L., & Kouvelis, P. (1995). Robust scheduling to hedge against processing time uncertainty in single-stage production. *Management science*, *41*(2), 363-376.

Demirkol, E., Mehta, S., & Uzsoy, R. (1997). A computational study of shifting bottleneck procedures for shop scheduling problems. *Journal of Heuristics*, *3*(2), 111-137.

Dominic, P. D., Kaliyamoorthy, S., & Kumar, M. S. (2004). Efficient dispatching rules for dynamic job shop scheduling. *The International Journal of Advanced Manufacturing Technology*, *24*(1), 70-75.

Dong, Yao-Hsiang, & Jang, Jaejin. (2012). Production rescheduling for machine breakdown at a job shop. International Journal of Production Research, 50(10), 2681–2691. https://doi.org/10.1080/00207543.2011.579637

Dorn, J., Kerr, R. M., & Thalhammer, G. (1995a). Reactive scheduling: improving the robustness of schedules and restricting the effects of shop floor disturbances by fuzzy reasoning. *International Journal of Human Computer Studies*, *42*, 687–704.

Dorn, J., Kerr, R. M., & Thalhammer, G. (1995b). Reactive scheduling: improving the robustness of schedules and restricting the effects of shop floor disturbances by fuzzy reasoning. *International Journal of Human Computer Studies*, *42*, 687–7

Duffie, N. A., & Piper, R. S. (1987). Non-hierarchical control of a flexible manufacturing cell. *Robotics and computer-integrated manufacturing*, *3*(2), 175-179.

Ebufegha, A., & Li, S. (2021, December). Multi-agent system model for dynamic scheduling in flexibile job shops. In *2021 Winter Simulation Conference (WSC)* (pp. 1-12). IEEE.

Eshragh, B. (2020). *Investigation of the impact of uncertainty factors on performance of cellular and functional manufacturing systems: A simulation study* (Master's thesis, Schulich School of Engineering).

Fang, J., & Xi, Y. (1997). A rolling horizon job shop rescheduling strategy in the dynamic environment. *The International Journal of Advanced Manufacturing Technology*, *13*, 227-232.

Fattahi, P., & Fallahi, A. (2010). Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability. *CIRP Journal of Manufacturing Science and Technology*, *2*(2), 114-123.

Fazayeli, M., Aleagha, M. R., Bashirzadeh, R., & Shafaei, R. (2016). A hybrid meta-heuristic algorithm for flowshop robust scheduling under machine breakdown uncertainty. *International Journal of Computer Integrated Manufacturing*, *29*(7), 709-719.

Fortemps, P. (1997). Jobshop scheduling with imprecise durations: a fuzzy approach. *IEEE Transactions on Fuzzy Systems*, *5*(4), 557-569.

Gao, K. Z., Suganthan, P. N., Tasgetiren, M. F., Pan, Q. K., & Sun, Q. Q. (2015). Effective ensembles of heuristics for scheduling flexible job shop problem with new job insertion. *Computers & Industrial Engineering*, *90*, 107-117.

Garey, M.R. and D.S. Johnson, (1979), Computers and intractability: a guide to the theory of NP- completeness. W.H. Freeman, New York.

Golovin, J., 1989. Real time dispatching for optimum scheduling. In: Proceedings of the A&D Symposium, APICS.

Goren, S., & Sabuncuoglu, I. (2009). Optimization of schedule robustness and stability under random machine breakdowns and processing time variability. IIE Transactions, 42(3), 203-220.

Gronau, N., & Theuer, H. (2016). Determination of the optimal degree of autonomy in a cyber-physical production system. *Procedia CIRP*, *57*, 110-115.

Hasan, S. K., Sarker, R., & Essam, D. (2011). Genetic algorithm for job-shop scheduling with machine unavailability and breakdowns. International Journal of Production Research, 49(16), 4999-5015.

Haupt, R. (1989). A survey of priority rule-based scheduling. *Operations-Research-Spektrum*, *11*(1), 3-16.

Haupt, R., 1989. A survey of priority rule-based scheduling. OR Spektrum 11, 3–16.

Hsu, C. Y., Kao, B. R., & Lai, K. R. (2016). Agent-based fuzzy constraint-directed negotiation mechanism for distributed job shop scheduling. Engineering applications of artificial intelligence, 53, 140-154.

Ishibuchi, H., Yamamoto, N., Murata, T., & Tanaka, H. (1994). Genetic algorithms and neighborhood search algorithms for fuzzy flowshop scheduling problems. *Fuzzy Sets and systems*, *67*(1), 81-100.

J. Adams, E. Balas, D. Zawack, The shifting bottleneck procedure for job shop scheduling, Management Science 34 (1988) 391– 401

Jeong, K.C., Kim, Y.D., 1998. A real-time scheduling mechanism for a flexible manufacturing system using simulation and dispatching rules. International Journal of Production Research 36, 2609–2626.

Kamble, S. S., Gunasekaran, A., Ghadge, A., & Raut, R. (2020). A performance measurement system for industry 4.0 enabled smart manufacturing system in SMMEs-A review and empirical investigation. *International Journal of Production Economics*, *229*, 107853.

Kang, K., Zhang, R. F., & Yang, Y. Q. (2007). MAS equipped with ant colony applied into dynamic job shop scheduling. In Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence: Third International Conference on Intelligent Computing, ICIC 2007, Qingdao, China, August 21-24, 2007. Proceedings 3 (pp. 823-835). Springer Berlin Heidelberg.

Kececioglu, D. (2002). *Reliability engineering handbook* (Vol. 1). DEStech Publications, Inc.

Kim, M.H., Kim, Y.D., 1994. Simulation-based real time scheduling mechanism in a flexible manufacturing system. Journal of Manufacturing Systems 13, 85–93.

Kutanoglu, E., & Sabuncuoglu, I. (2001). Routing-based reactive scheduling policies for machine failures in dynamic job shops. International journal of production research, 39(14), 3141-3158.

L.J. Muris , G.F. Moacir , Production planning and control for remanufacturing: literature review and analysis, Prod. Plann. Control 23 (2012) 419–435 .

Lawler, E. L., Lenstra, J. K., Kan, A. H. R., & Shmoys, D. B. (1993). Sequencing and scheduling: Algorithms and complexity. *Handbooks in operations research and management science*, *4*, 445-522.

Lawrence, S. R., & Sewell, E. C. (1997). Heuristic, optimal, static, and dynamic schedules when processing times are uncertain. *Journal of Operations Management*, *15*(1), 71-82.

Li Y, Li SJ, Liu Y, Liu ZG (2005) Dynamic scheduling method based on combination of contract net with mediator. Proceedings of IEEE International Conference on Machine Learning and Cybernetics, 18–21 Aug, Guangzhou, China, pp 339–344. doi:10.1109/ICMLC.2005.1526969"

Li, K., Leung, J. T., & Cheng, B. Y. (2014). An agent-based intelligent algorithm for uniform machine scheduling to minimize total completion time. Applied soft computing, 25, 277-284.

Li, Y., Wang, J., & Liu, Z. (2022). A simple two-agent system for multi-objective flexible job-shop scheduling. Journal of Combinatorial Optimization, 43(1), 42-64.

Liao, C. J., & Chen, W. J. (2003). Single-machine scheduling with periodic maintenance and nonresumable jobs. *Computers & Operations Research*, *30*(9), 1335-1347.

MacCarthy, B. L., & Liu, J. (1993). Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. *International Journal of Production Research*, *31*(1), 59–79.

Madureira, A., Gomes, N., Santos, J., & Ramos, C. (2007, July). Cooperation mechanism for team-work based multi-agent system in dynamic scheduling through meta-heuristics. In 2007 IEEE International Symposium on Assembly and Manufacturing (pp. 233-238). IEEE.

Mehta, S. V., & Uzsoy, R. (1999). Predictable scheduling of a single machine subject to breakdowns. *International Journal of Computer Integrated Manufacturing*, *12*(1), 15–38.

Mehta, S. V., & Uzsoy, R. (1999). Predictable scheduling of a single machine subject to breakdowns. *International Journal of Computer Integrated Manufacturing*, *12*(1), 15–38.

Mehta, S.V., Uzsoy, R.M., 1998. Predictable scheduling of a job shop subject to breakdowns. IEEE Transactions on Robotics and Automation 14, 365–378.

Merdan, M., Moser, T., Vrba, P., & Biffl, S. (2011). Investigating the robustness of re-scheduling policies with multi-agent system simulation. The International Journal of Advanced Manufacturing Technology, 55(1-4), 355-367.

Mohan, S., Clancy, D., 1990. SIS—Rule based software for automating jobdispatch on the factory floor. In: Proceedings of the Third International Conference on Expert Systems and the Leading Edge in Production Planning and Control, Hilton Head, NC.

Mönch, L., Schabacker, R., Pabst, D., & Fowler, J. W. (2007). Genetic algorithm-based subproblem solution procedures for a modified shifting bottleneck heuristic for complex job shops. *European Journal of Operational Research*, *177*(3), 2100-2118.

Monostori, L. (2014). Cyber-physical production systems: Roots, expectations and R&D challenges. *Procedia Cirp*, *17*, 9-13.

Moratori, P., Petrovic, S., & Vázquez-Rodríguez, J. A. (2010). Integrating rush orders into existent schedules for a complex job shop problem. *Applied Intelligence*, *32*(2), 205-215.

Muhlemann, A. P., Lockett, A. G., & Farn, C. K. (1982). Job shop scheduling heuristics and frequency of scheduling. *The International Journal of Production Research*, *20*(2), 227-241.

Nie, L., Gao, L., Li, P., & Shao, X. (2013). Reactive scheduling in a job shop where jobs arrive over time. *Computers & Industrial Engineering*, *66*(2), 389-405.

Nie, Q., Tang, D., Zhu, H., & Sun, H. (2022). A multi-agent and internet of things framework of digital twin for optimized manufacturing control. International Journal of Computer Integrated Manufacturing, 35(10-11), 1205-1226.

Nouiri, M., Bekrar, A., Jemai, A., Trentesaux, D., Ammari, A. C., & Niar, S. (2017). Two stage particle swarm optimization to solve the flexible job shop predictive scheduling problem considering possible machine breakdowns. Computers & Industrial Engineering, 112, 595-606.

O’Donovan, R., Uzsoy, R., & McKay, K. N. (1999). Predictable scheduling of a single machine with breakdowns and sensitive jobs. *International Journal of Production Research*, *37*(18), 4217–4233.

Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of scheduling*, *12*(4), 417-431.

Ovacik, I.M., Uzsoy, R., 1997. Decomposition Methods For Complex Factory Scheduling Problems. Kluwer Academic Publishers.

Pal, M., Mittal, M. L., Soni, G., Chouhan, S. S., & Kumar, M. (2023). A multi-agent system for FJSP with setup and transportation times. Expert Systems with Applications, 216, 119474.

Parunak, V.D., Manufacturing experience with the contract net. In Distributed Artificial Intelligence, edited by M.N. Huhns, pp. 285–310, 1987 (Pitman: London).

Perry, C. N., & Uzsoy, R. (1993, October). Reactive scheduling of a semiconductor testing facility. In *Proceedings of 15th IEEE/CHMT International Electronic Manufacturing Technology Symposium* (pp. 191-194). IEEE.

Petrovic, S., & Fayad, C. (2004, September). A fuzzy shifting bottleneck hybridised with genetic algorithm for real-world job shop scheduling. In *Proceedings of Mini-Euro Conference, managing uncertainty in decision support models, Coimbra, Portugal* (pp. 1-6).

Piramuthu, S., Park, S.-C., Raman, N., Shaw, M.J., 1991. Integration of simulation modelling and inductive learning in an adaptive decision support system. In: Boczelc, A., Whinston, A. (Eds.), Model Management Systems. IEEE Society Press

Rajabinasab, A., & Mansour, S. (2011). Dynamic flexible job shop scheduling with alternative process plans: an agent-based approach. The International Journal of Advanced Manufacturing Technology, 54, 1091-1107.

Ramasesh, R., 1990. Dynamic Jobshop Scheduling: A survey of Simulation Research. OMEGA 18, 43–57.

Sabuncuoglu, I., & Bayiz, M. (2000). Analysis of reactive scheduling problems in a job shop environment. *European Journal of Operational Research*, *126*(3), 567–586.

Sabuncuoglu, I., & Karabuk, S. (1998). Analysis of the frequency of rescheduling in an FMS with stochastic processing times and machine breakdowns. *Department of Industrial Engineering. Bilkent University, Turkey*.

Safari, E., Jafar Sadjadi, S., & Shahanaghi, K. (2010). Scheduling flowshops with condition-based maintenance constraint to minimize expected makespan. *The International Journal of Advanced Manufacturing Technology*, *46*, 757-767.

Seifoddini, H., & Djassemi, M. (1995). Merits of the production volume based similarity coefficient in machine cell formation. *Journal of Manufacturing Systems*, *14*(1), 35-44.

Shaked, M., & Shanthikumar, J. G. (1994). *Stochastic orders and their applications*. Academic press.

Shen, W., & Norrie, D. H. (1999). Agent based systems for intelligent manufacturing: a state of the art survey. *International Journal of Knowledge and Information Systems*, *1*(2), 129–156

Shen, W., Norrie, D. H., & Barthes, J. P. A. (2001). *Multi-agent systems for concurrent intelligent design and manufacturing*. London: Taylor & Francis.

Shi, L., Guo, G., & Song, X. (2021). Multi-agent based dynamic scheduling optimisation of the sustainable hybrid flow shop in a ubiquitous environment. *International Journal of Production Research*, *59*(2), 576-597.

Smith, R.G., The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Trans. on Comp., 1980, 29, 1104–1113

Sun, J., & Xue, D. (2001). A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources. *Computers in Industry*, *46*(2), 189–207

Tharumarajah, A. (2001). Survey of resource allocation methods for distributed manufacturing systems. *Production Planning & Control*, *12*(1), 58–68.

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., & Peeters, P. (1998). Reference architecture for holonic manufacturing systems: PROSA. *Computers in industry*, *37*(3), 255-274.

Vieira, G. E., Hermann, J. W., & Lin, E. (2000). Predicting the performance of rescheduling strategies for parallel machine systems. *Journal of Manufacturing Systems, 19*(4), 256–266

Vieira, G. E., Hermann, J. W., & Lin, E. (2003). Rescheduling manufacturing systems: a framework of strategies, policies and methods. *Journal of Scheduling*, *6*(1), 36–92.

Wang, D. J., Liu, F., Wang, Y. Z., & Jin, Y. (2015). A knowledge-based evolutionary proactive scheduling approach in the presence of machine breakdown and deterioration effect. *Knowledge-Based Systems*, *90*, 70-80.

Wang, Z., Zhang, J., & Si, J. (2019, September). Dynamic job shop scheduling problem with new job arrivals: a survey. In *Chinese Intelligent Automation Conference* (pp. 664-671). Springer, Singapore.

Wang, Z., Zhang, J., & Si, J. (2020). Dynamic job shop scheduling problem with new job arrivals: A survey. In *Proceedings of 2019 Chinese Intelligent Automation Conference* (pp. 664-671). Springer Singapore.

Wei, Y., Gu, K., Liu, H., & Li, D. (2007, March). Contract net based scheduling approach using interactive bidding for dynamic job shop scheduling. In 2007 IEEE International Conference on Integration Technology (pp. 281-286). IEEE.

Wu, S. D., Byeon, E. S., & Storer, R. H. (1999). A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness. *Operations research*, *47*(1), 113-124.

Wu, S. D., Storer, R. H., & Chang, P. C. (1991). A rescheduling procedure for manufacturing systems under random disruptions. In *Proceedings joint USA/German conference on new directions for Operations Research in Manufacturing* (pp. 292–306).

Wu, S.D., Wysk, R.A., 1989. An application of discrete-event simulation to on-line control and scheduling of flexible manufacturing. International Journal of Production Research 27 (9).

Xiang, W., & Lee, H. P. (2008). Ant colony intelligence in multi-agent dynamic manufacturing scheduling. Engineering Applications of Artificial Intelligence, 21(1), 73-85.

Xiong, J., Xing, L. N., & Chen, Y. W. (2013). Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns. International Journal of Production Economics, 141(1), 112-126.

Xiong, W., & Fu, D. (2018). A new immune multi-agent system for the flexible job shop scheduling problem. Journal of Intelligent Manufacturing, 29, 857-873.

Yamamoto, M., & Nof, S. Y. (1985). Scheduling/rescheduling in the manufacturing operating system environment. *International Journal of Production Research*, *23*(4), 705-722.

Yu, X., & Ram, B. (2006). Bio-inspired scheduling for dynamic job shops with flexible routing and sequence-dependent setups. International journal of production research, 44(22), 4793-4813.

Yuan, Y., & Xu, H. (2013). Multiobjective flexible job shop scheduling using memetic algorithms. *IEEE Transactions on Automation Science and Engineering*, *12*(1), 336-353.

Yun, Y. S. (2002). Genetic algorithm with fuzzy logic controller for preemptive and non-preemptive job-shop scheduling problems. *Computers & industrial engineering*, *43*(3), 623-644.

Žapčević, S., & Butala, P. (2013). Adaptive process control based on a self-learning mechanism in autonomous manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, *66*(9-12), 1725-1743

Zhang, S., & Wong, T. N. (2017). Flexible job-shop scheduling/rescheduling in dynamic environment: a hybrid MAS/ACO approach. International Journal of Production Research, 55(11), 3173-3196.