Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo and M. Loper, eds.

MULTI-AGENT SYSTEM MODEL FOR DYNAMIC SCHEDULING IN FLEXIBILE JOB SHOPS

Akposeiyifa Ebufegha

Department of Mechanical and Manufacturing Engineering
University of Calgary
2500 University Dr NW
Calgary, AB T2N 1N4, CANADA

Simon Li

Department of Mechanical and Manufacturing Engineering
University of Calgary
2500 University Dr NW
Calgary, AB T2N 1N4, CANADA

ABSTRACT

One of the hallmarks of industry 4.0 is the development of a smart manufacturing system (SMS). These are highly modular systems, with every physical resource being autonomous and capable of exchanging information with each other over an industrial network. The resources can self-organize to schedule job shop operations in real-time. The ability to schedule in real-time allows for better use of the flexibility in part processing operation sequences than with conventional manufacturing systems. This could potentially result in reduced order completion times and increased average machine utilization. However, it is difficult to investigate the benefits of such a system as they are expensive to build as such a simulation is necessary. This paper presents model for a dynamic scheduling in an SMS well as a multi-method model for simulating its operation. The paper also presents a preliminary investigation into the benefits of the proposed scheduling strategy.

1 INTRODUCTION

Manufacturing systems consist of machines that are capable of performing operations required to produce of a set of parts if done in the correct sequence. These parts can then be assembled together to form a set of products. Determining how best a set of machines in a given facility can be deployed to fulfill an order for a set products so as to reduce the material handling cost is the very core of the job shop scheduling problem. Typically, this problem is addressed using proactive and reactive scheduling approaches. With proactive approaches, the project manager creates plans for the work to mitigate the effect of potential disruptions (Xiong et al. 2013; Fazayeli et al. 2016; Wang et al. 2015). Reactive scheduling approaches typically focus on scheduling policies. Reactive scheduling requires modifying the created schedule to adapt to real-time changes to the production environment through focus on scheduling policy (Sun and Xue 2001; Kutanoglu and Sabuncuoglu 2001; Fahmy et al. 2008). There are even instances where proactive-reactive scheduling approaches are employed (Rahmani 2017; Aloulou and Portmann 2005).

However, the problem is that these approaches require restricting the flexibility of the system by pre-assigning the processing routes for each part (the sequence of machines, and the associated operations, by which parts flow through the system). The approach presented in this paper allows the system to utilize more of its flexibility. This is done by turning the machines and products within the system to smart machines and smart parts. The manufacturing now becomes a smart manufacturing system, consisting of autonomous parts and machines that negotiate between themselves to meet the product demand. These negotiations are governed by a set of rules designed to maximize machine utilization and also reduce part flowtime. The greatest benefit of this system is in its robustness as the autonomy of the agents within the system it to make real-time adjustments to unforeseen events. Investigating the use of agents to make real-

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

time scheduling decisions during production using a more complex decision making process than common dispatching rules without having to define a schedule prior to production is the core of this research.

Before scheduling, during the planning phase, the floor manager develops a process plan network to outline all possible combinations of machines and operations that can be used to produce each part. This represents the system's flexibility. There are three types of system flexibility that are relevant to a given set of machines and demand for parts (Gupta and Goyal 1989); (1) routing flexibility, (2) sequential flexibility, and (3) processing flexibility. Routing flexibility refers to the possible options for completing operations on alternate machines. Sequential flexibility is possibility to reordering the sequence of operations required to produce a part. Processing flexibility refers to the options available to process a part using alternate sets of operations. The combination of these types of flexibility provides the solution space for how a part can be made. During scheduling, one of these options from the solution space is selected for each part to meet the scheduling objective (e.g., minimize makespan, minimize material handling cost, etc.). This approach is limiting in that the best option for the production of a part at a specific point in time is not necessarily the processing option that will be chosen. However, selecting the best path for a part to take following each of its processing steps requires constant evaluation of the full system state and complete knowledge of the processing plan. This would prove difficult for a scheduler to do. The problem is further compounded by the fact that this must be done for all parts within the system simultaneously. Normally, this would be impractical. However, with the advancements brought about by industry 4.0 (cloud computing and smart devices, to be specific) it is possible to offload the responsibility of determining the best next step for all parts to the parts themselves. The manufacturing system becomes a multi-agent system (MAS) that acts as a distributed problem solver for scheduling work in real-time. In this paper, we present a model for developing such a system.

Our objectives are twofold. To design a system of autonomous agents that minimizes total order completion time and maximizes machine utilization. We also aim to demonstrate the efficacy of reactive scheduling using a MAS as opposed to using simple dispatching rules along with a predefined schedule. In section 2, we present a description of our proposed manufacturing system. In section 3, we discuss how our proposed system was implemented via simulation. In section 4, we present a sample problem to which we applied our smart manufacturing system model and a conventional approach via simulation. In section 5, we present and discuss the results of our simulation and compare their performances. In section 6, we present our conclusion and recommendations for future directions for the work.

2 SYSTEM DESCRIPTION

The smart manufacturing system is a cyber-physical production system (CPPS). A CPPS consists of a physical layer (smart machines, transporters and parts), a network layer (industrial WAN or LAN), cloud layer (software as a service, cloud storage, etc.) and, supervisory and control layer (human interface with the system) (Wang et al. 2016). This system is defined by the autonomy of the components that comprise the physical layer. In this model description, we will primarily focus on the physical layer, specifically parts and machines. The cloud layer is not explicitly modeled as it is a data storage and data processing system. The network layer is also not explicitly modelled but is implied by the ability of the system's agents to communicate. Similarly, the supervisory and control layer is implicitly modelled in the planning and scheduling process.

2.1 The Physical Layer Description

The agent environment is a shopfloor consisting of machines, transporters, and raw materials in an arrangement that minimizes material handling costs. These resources are put to the production of parts which can be assembled into products. In our model, it is assumed that there are sufficient materials and transporters such that transfer of parts between machines is never delayed and neither is the initiation of work on a part. As such, our model focuses primarily on the parts and machines.

When a part order comes into the system, it requires the system to complete a sequence of operations to produce that part. Let $O = \{o_1, o_2, \ldots, o_{no}\}$ represent all operations that can be performed given a set of nm available machines. Similarly, let J_{ir} represent the sequence of operations required as a production step for part p_i using route nr from the process plan network. In our model, it is possible for parts to be processed using a different set of operations and operations sequences, which we have termed routes. For each part a route consists of a subset of operations that the manufacturing system can produce ($J_{ir} \subseteq \{o_1, o_2, \ldots, o_{no}\}$). The operation sequence choice for each part is decided in real-time by each part's intelligence based on the machine availability and the shortest processing path currently available. This autonomy in sequence choice is part of how our approach allows for more utilization of the system's flexibility. Note that the same operation sequence may be executed with different combinations of machines (routing flexibility). Also, note that some operation sequences may be permutations of each other (sequential flexibility) or they can be entirely distinct from each other (processing flexibility).

2.1.1 Parts

Allow $P = \{p_1, p_2, \ldots, p_{np}\}$ to represent the np parts that can be produced with the proposed manufacturing system. The possible routes that can be used to produce a part has been represented using adjacency matrix, R. Where $R_{ij} = 1$ indicates that from operation o_i it is possible to transition to o_j as the next processing step. An example of this adjacency matrix is shown in Figure 1. The example in Figure 1 shows that, for the given part, there are two routing options {start, o_1, o_5, end} or {start, o_3, o_4, end}. For each part there is an adjacency matrix. Also, when a part order comes into the system, a part agent that represents it is also generated. This agent represents the given part's intelligence and will use the adjacency matrix to determine which operation(s) to request from the machines of the system at the current time as well as to track if the part has been completed. For example, assume the part whose adjacency matrix is depicted in Figure 1 enters the system. The agent representing this part will request operations o_1 and o_3 as work it needs from the system. The system will respond with available options and one of the two operations will be performed. After which, the agent will reference the appropriate column (either o_1 or o_3) to determine which operations to request from the system. This process will continue until it arrives at the end.

	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 1: Operation Sequence Adjacency Matrix

2.1.2 Machines

The manufacturing system is comprised of a set of machines all of which can perform at least one operation. Let all machines be represented $M = \{m_1, m_2, \ldots, m_{nm}\}$. Each element of M can perform some combination of operations from O. Let $o(m_i) \subseteq \{o_1, o_2, \ldots, o_{no}\}$ such that $o(m_i)$ represents the operations that can be performed by machine m_i. The processing and set up times for each of these operations on each machine is part dependent. If $o(m_A) \equiv o(m_B)$ and the processing and set up times for each operation are the same, then m_A and m_B are duplicates. If the processing and set up times are different, then these machines are similar but not duplicates. If $o(m_A) \neq o(m_B)$ they are dissimilar machines regardless of if they share some operations capabilities in common or not. This information can be determined by comparing the machine-part-operation relationship matrix, S, for each machine. A sample machine-part-operation relationship matrix is shown in Figure 2. In our model we use a triangular distribution for the setup times and processing times associated with each operation. With the triangular distribution, we have the best-case duration (B), worst-case duration (W) and most likely duration (M). A triangular distribution is employed because it only needs three data points to construct it. If a machine cannot be used to perform an operation for a part, it is assigned 'inf' which means that it would take an infinite amount of time to perform the operation.

Part/Op		O1		O2		O3	
		S	P	S	P	S	P
1	W	2	6	Inf	Inf	Inf	Inf
	M	1.5	5	Inf	Inf	Inf	Inf
	B	1	4	Inf	Inf	Inf	Inf
2	W	Inf	Inf	2	6	Inf	Inf
	M	Inf	Inf	1.5	5	Inf	Inf
	B	Inf	Inf	1	4	Inf	Inf

	M1	M2	M3	M4	M5
M1	0	1	1	1.2	1.2
M2	1	0	1.2	1	1.2
M3	1	1.2	0	1.2	1
M4	1.2	1	1.2	0	1
M5	1.2	1.2	1	1	0

Figure 2: Sample Machine-Part-Operation Relationship Matrix (left) & Distance Matrix (right)

We assume that the relative distances between all machines in the system is known (contained in a distance matrix, D). This information, in combination with the transporter speed, is used to determine the transfer time between machines. For each machine in the system, there is an agent representative (machine agent). The machine agent uses the information from Figure 2 to provide an estimate of how long it would take to complete a given operation for a part using this machine. For example, let us assume that part p_1 is currently at machine m_1. If part p_1 requests operation o_1 from the system and this can only be performed by machine m_2. The machine agent associated with m_2 (whose part-operation relationship is depicted in Figure 2) would return an estimate stating that the work will take 1-2 time units to set up and between 4 and 6 time units of processing time. It would also return the transfer time of 1 time unit, based on the relative distance between the machine and the part's current location and the transfer speed (1 m/time unit).

2.2 Physical Layer Agent Behavior and Interactions

The approach being presented makes use of 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). The model presented here consists of three (3) agents; a parts agent (PA), machine agent (MA) and supervisory agent (SA). An overview of each agent's functions, inputs and outputs can be seen in Figure 3. Figure 3 also shows the information flow between each agent.

Each PA represents a part in the system and is responsible for the decisions on which operation sequence for their associated part is followed. Before each processing step, a PA announces to all machines that it requires work to be done on the part it represents. It requests bids from machines based on an estimate of time required to get the part to the next processing step. Upon receiving bids, the PA selects a machine to assign the job to and also creates a ranked list of alternate machines. The part agent's objective is to minimize the flow time for the specific part. The end goal is to produce the part as quickly as possible. This is done by selecting the available machine and operation combination at each processing step that advances the part to the next processing step the fastest. This is done by comparing the sum of transfer times (TT), processing times (*PT*) and set up times (*ST*) and selecting the option that yields the shortest durations. The objective function is as follows:

$$\min\left\{ (TT + ST + PT)_{mx}, (TT + ST + PT)_{my}, \dots \right\} \tag{1}$$

An MA represents a specific machine within the system. The objective of the machine agent is to maximize the utilization of the machine it represents. This can be done by increasing the amount of work assigned to the machine whilst reducing the idle time. In the model, the machine agent is free to bid when it has no work assigned to it but cannot bid once assigned work.

The MA's review all requests for work from the PA's in the system and if it 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, set up time and processing time for the specific part and operation combination. The PAs review their bids and select winners and ranks the remaining bids.

If there is a conflict (i.e. two PA's awarding work to the same machine), the SA intervenes. It requests and reviews a ranked list of alternate machines provided by the PAs and then assigns work based on minimizing the maximum flowtime (*FT*) for all parts currently in the system. The objective function the SA utilizes when it interferes is:

$$\min\left(\max\left\{FT_1, FT_2, \dots FT_{np}\right\}\right) \tag{2}$$

The SA's primary function is to resolve conflicts within the system and only acts when a conflict is observed. If no MA bids on a PA's 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. Note, it is assumed that there will always be sufficient storage capacity for work-in-process (WIP) in the system. A more comprehensive illustration of the behavior of and the interaction between the agents in the system is represented in Figure 4.

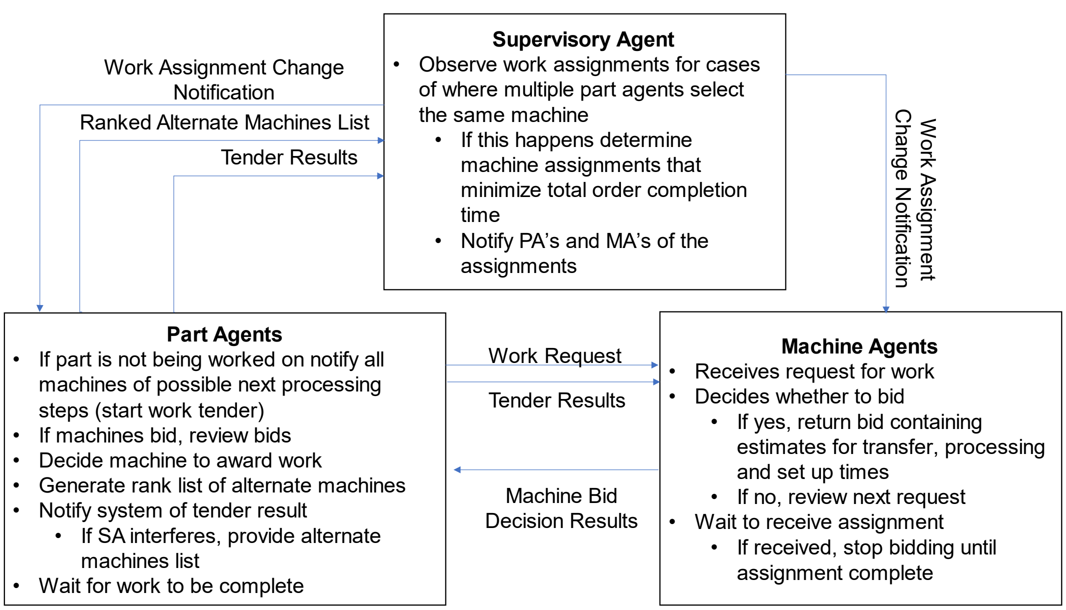


Figure 3: Agent Behavior

3 SYSTEM IMPLEMENTATION

To develop a simulation model for this system, we employed multi-method modelling approach that utilizes the three main modelling techniques used in operations research for simulation;

1. *Agent-based modelling* (AB) – This is used to simulate the autonomy and intelligence of the cyber-physical systems. Its primary purpose is to handle the negotiations that occur between the smart products entering the system and the smart devices within the system. Simulating the multi-agent system is the most important portion of the model as it determines which events will occur. As such, further elaboration of the agent interaction is provided in Figure 4.
2. *System Dynamics* (SD) – System dynamics is used to simulate the flow of the smart product through the manufacturing system. It primarily serves to track changes in information on the status of the physical layer of the factory.
3. *Discrete Event Simulation* (DES) – This approach is used to simulate the execution of jobs in the system.

In our model, when an unprocessed order enters the system, the SD model updates the stock of orders in the system. The AB and DES models simulate the parts decisions and the actual events (operations being executed). This results in the generation of WIP and transition of the WIP to finished goods. The AB and DES cause changes to the system state and this information is used to update the SD model which tracks the flow of WIP to finished products. The interaction and information flow between these three simulation modeling techniques is what results in a multi-method simulation model. The multi-method simulation model was implemented using an in-house developed script written in MATLAB R2021a.

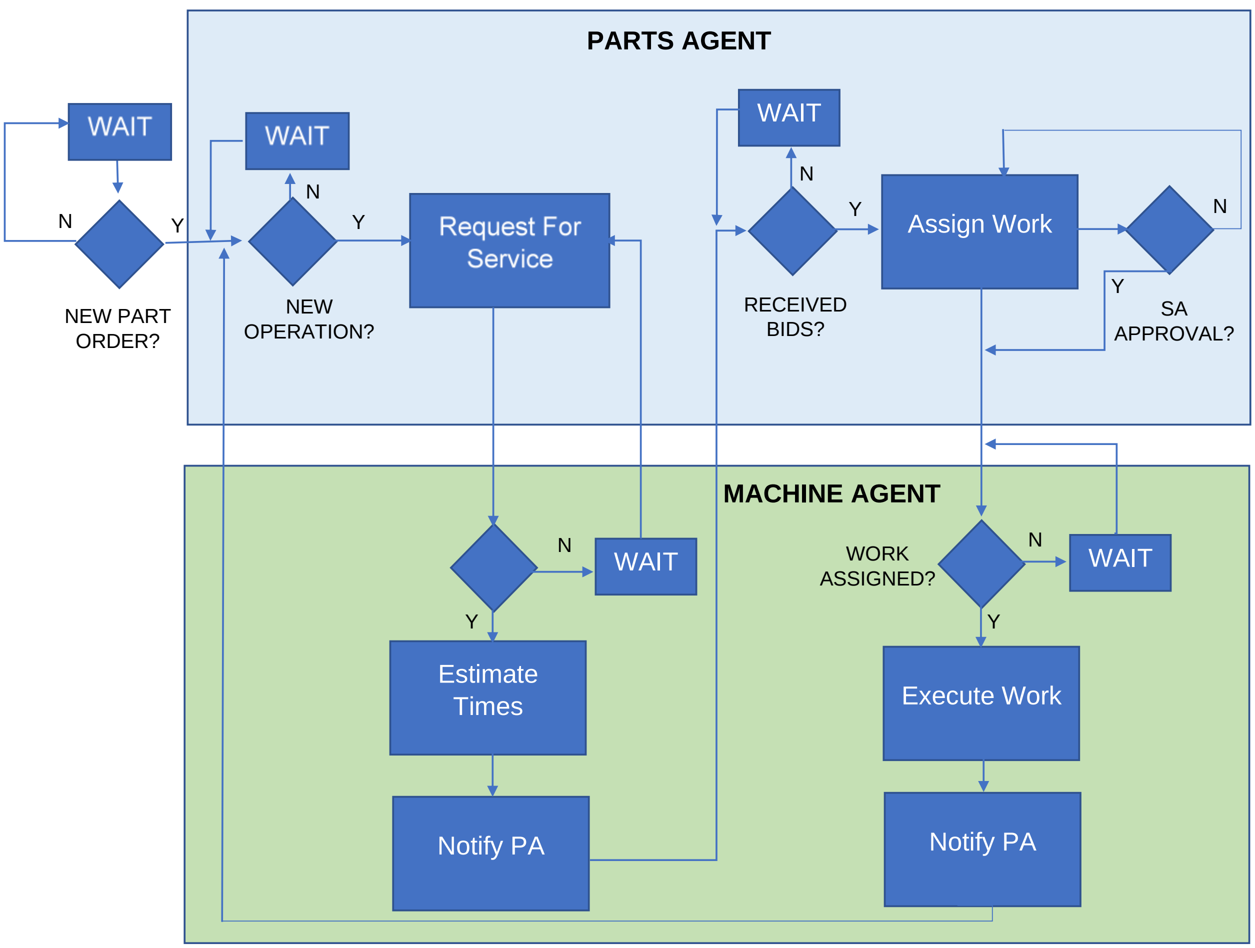


Figure 4: Multi-Agent System Model

4 EXAMPLE PROBLEM

In our study, we simulate the operation of a smart manufacturing system that is capable of producing three (3) parts using four (4) machines that are capable of performing fifteen (15) operations. For the given sample problem, it is assumed that there are sufficient transporters to serve the system's needs and each transporter has a consistent speed of 60 m/hr. This problem is a self-generated problem created for a preliminary investigation into the performance of smart manufacturing system. Table 1 outlines the processing options for each part. Table 2 outlines the capabilities of the system's machines.

The problem consists of four scenarios each with an increase in demand for each part. Table 3 outlines the demand for these scenarios. This is so that the systems performance when there are more parts in the system than machines to service them can be observed. The setup times and processing times are presented in the form of a triangular distribution.

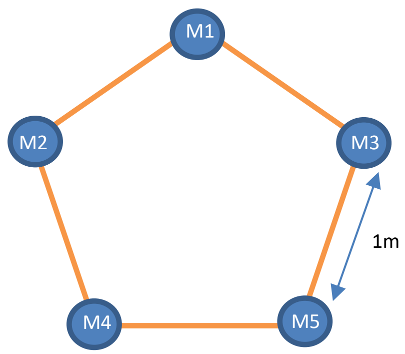
During simulation Latin Hypercube sampling is used to ensure that times used in the simulation are representative of their respective distributions. For each scenario, the problem is simulated five (5) times and the mean and standard deviation for the time and machine utilization are recorded. The following is a summary of the problem conditions:

Table 1: Part Operation Sequence Options

Part No.	Operation Sequence Options
1	Option 1: $\{o_1, o_2, o_3\}$ Option 2: $\{o_7, o_8, o_3\}$
2	Option 1: $\{o_4, o_5, o_6\}$ Option 2: $\{o_{10}, o_5, o_6\}$ Option 3: $\{o_4, o_5, o_{12}\}$ Option 4: $\{o_4, o_5, o_{12}\}$
3	Option 1: $\{o_2, o_{14}, o_{15}\}$ Option 2: $\{o_{13}, o_{14}, o_{15}\}$

Table 2: Machine Capabilities

Machine No.	Operation Capabilities	Processing Time in Hours (Best, Most Likely, Worst)	Set up Times Time in Hours (Best, Most Likely, Worst)
1	o_1	[4, 5, 6]	[1, 1.5, 2]
	o_4	[9, 10, 11]	[1, 1.5, 2]
	o_{14}	[9, 10, 11]	[1, 1.5, 2]
2	o_2	[9, 10, 11]	[1, 1.5, 2]
	o_7	[9, 10, 11]	[1, 1.5, 2]
	o_{10}	[4, 5, 6]	[1, 1.5, 2]
3	o_6	[9, 10, 11]	[1, 1.5, 2]
	o_{12}	[4, 5, 6]	[1, 1.5, 2]
	o_{13}	[4, 5, 6]	[1, 1.5, 2]
4	o_3	[4, 5, 6]	[1, 1.5, 2]
	o_5	[4, 5, 6]	[1, 1.5, 2]
	o_{14}	[4, 5, 6]	[1, 1.5, 2]
5	o_8	[9, 10, 11]	[1, 1.5, 2]
	o_{10}	[1, 2, 3]	[1, 1.5, 2]
	o_{15}	[4, 5, 6]	[1, 1.5, 2]



	M1	M2	M3	M4	M5
M1	0	1	1	1.2	1.2
M2	1	0	1.2	1	1.2
M3	1	1.2	0	1.2	1
M4	1.2	1	1.2	0	1
M5	1.2	1.2	1	1	0

Figure 5: Facility Layout and Accompanying Distance Matrix

Table 3: Part Order Arrival Sequence

Simulation Scenario	Order Arrival Sequence
1	[p_1, p_2, p_3]
2	[p_1, p_2, p_3, p_1, p_2, p_3]
3	[p_1, p_2, p_3, p_1, p_2, p_3, p_1, p_2, p_3, p_1, p_2, p_3]
4	[p_1, p_1, p_2, p_3, p_3, p_1, p_2, p_2, p_3, p_1, p_2, p_3, p_3, p_3, p_1, p_2, p_3]

Figure 5 depicts the facility layout as well as the relative distances between the machines. In Table 4, we present a scheduling solution to the problem presented in this section based on proactive scheduling approach. The performance of this solution is to be compared against that of our agent-based approach. Each part has been assigned a route based on the objective of minimizing total order completion time. The schedule assumes first-come-first-serve (FCFS) dispatching. In this approach, each part has a preassigned route it must follow and the parts are processed in a FCFS basis.

Table 4: Scheduling Solution Using Proactive Scheduling Approach

Part No.	Route and Operation Selection	
1	Route	$m_1 \rightarrow m_2 \rightarrow m_4$
	Operation	$o_1 \rightarrow o_2 \rightarrow o_3$
2	Route	$m_2 \rightarrow m_4 \rightarrow m_3$
	Operation	$o_{10} \rightarrow o_5 \rightarrow o_{12}$
3	Route	$m_3 \rightarrow m_1 \rightarrow m_5$
	Operation	$o_{13} \rightarrow o_{14} \rightarrow o_{15}$

5 SIMULATION RESULTS AND DISCUSSION

Tables 5 – 8 contain the output results from our simulation of the key performance measures for the smart manufacturing system presented in section 4. The measures being examined are machine uptime, machine utilization, mean part wait time and mean part flow time. Table 5 contains the mean completion times for the complete order. Table 6 contains the mean flow times and wait times for each part. Table 7 contains the recorded machine uptimes. Table 8 contains the machine utilization for each machine in the system. Each of the mean values that are recorded are for five simulation runs for each scenario.

The measures that have been selected will guide our evaluation of how well the smart manufacturing system performed in relation to our primary objective of minimizing the order completion time and maximizing machine utilization. Please note that the values provided are means for the five simulation runs and their standard deviations are provided in brackets.

Table 5: Mean Total Order Completion Time (After 5 Simulation Runs)

	Scenario	Total Order Completion Time
Our Proposed Approach	1	25.8 (0.24)
	2	38.0 (0.31)
	3	76.9 (0.40)
	4	93.5 (0.72)
Proactive Approach	1	25.1 (0.19)
	2	43.7 (0.22)
	3	79.8 (0.56)
	4	121.0 (1.13)

Table 6: Part Flowtimes and Wait Times (in Hours)

	Scenario	Flowtime			**Mean**	Wait Time			**Mean**
		p_1	p_2	p_3		p_1	p_2	p_3	
Our Proposed Approach	1	25.3 (0.23)	19.1 (0.32)	25.8 (0.24)	**23.4**	2.1 (0.10)	5.4 (0.17)	1.8 (0.14)	**3.1**
	2	32.6 (0.27)	32.7 (0.32)	30.8 (0.37)	**32.0**	8.7 (0.14)	18.2 (0.30)	8.7 (0.12)	**11.9**
	3	54.9 (1.04)	51.6 (0.29)	37.7 (0.24)	**48.1**	18.9 (0.68)	25.3 (0.10)	10.7 (0.12)	**18.3**
	4	73.2 (0.15)	70.1 (0.63)	54.8 (0.19)	**66.0**	33.2 (0.11)	30.1 (0.19)	33.8 (0.10)	**32.4**
Proactive Approach	1	25.1 (0.19)	20.8 (0.22)	24.5 (0.21)	**23.5**	2.0 (0.09)	5.6 (0.14)	1.6 (0.13)	**3.1**
	2	37.5 (0.23)	23.2 (0.23)	34.0 (0.14)	**31.6**	13.0 (0.20)	3.7 (0.21)	9.5 (0.07)	**8.8**
	3	57.6 (0.14)	41.6 (0.15)	64.0 (0.25)	**54.4**	33.1 (0.14)	22.1 (0.13)	39.5 (0.14)	**31.6**
	4	76.2 (0.40)	61.3 (0.30)	85.9 (0.26)	**74.5**	51.7 (0.12)	41.8 (0.15)	61.4 (0.14)	**51.7**

Table 7: Machine Uptime in Hours

	Scenario	m_1	m_2	m_3	m_4	m_5	**Mean**
Our Proposed Approach	1	16.6 (0.06)	10.6 (0.07)	14.8 (0.04)	8.1 (0.03)	6.9 (0.05)	**11.4**
	2	15.0 (0.04)	20.0 (0.04)	31.8 (0.04)	23.4 (0.06)	21.9 (0.05)	**22.4**
	3	23.3 (0.02)	52.5 (0.03)	53.2 (0.02)	50.7 (0.04)	35.7 (0.04)	**43.1**
	4	58.7 (0.05)	73.9 (0.05)	77.4 (0.08)	69.1 (0.02)	52.6 (0.03)	**66.4**
Proactive Approach	1	13.5 (0.04)	15.1 (0.02)	11.1 (0.02)	8.6 (0.03)	4.5 (0.02)	**10.6**
	2	28.4 (0.02)	29.6 (0.02)	18.1 (0.02)	21.5 (0.02)	8.2 (0.02)	**21.2**
	3	61.5 (0.03)	56.9 (0.02)	41.4 (0.01)	39.0 (0.01)	18.7 (0.03)	**43.5**
	4	96.6 (0.02)	77.6 (0.03)_	59.0 (0.02)	54.1 (0.03)	30.7 (0.03)	**63.6**

Table 8: Machine Utilization

	Scenario	m_1	m_2	m_3	m_4	m_5	**Mean**
Our Proposed Approach	1	0.64 (0.00)	0.42 (0.00)	0.57 (0.00)	0.31 (0.00)	0.27 (0.00)	**0.44**
	2	0.39 (0.00)	0.53 (0.00)	0.84 (0.00)	0.62 (0.00)	0.58 (0.00)	**0.59**
	3	0.30 (0.00)	0.68 (0.00)	0.69 (0.00)	0.66 (0.00)	0.46 (0.00)	**0.56**
	4	0.63 (0.00)	0.79 (0.00)	0.83 (0.00)	0.74 (0.00)	0.56 (0.00)	**0.71**
Proactive Approach	1	0.54 (0.00)	0.60 (0.00)	0.44 (0.00)	0.34 (0.00)	0.18 (0.00)	**0.42**
	2	0.65 (0.00)	0.68 (0.00)	0.41 (0.00)	0.49 (0.00)	0.18 (0.00)	**0.48**
	3	0.77 (0.00)	0.71 (0.00)	0.52 (0.00)	0.48 (0.00)	0.23 (0.00)	**0.54**
	4	0.80 (0.00)	0.64 (0.00)	0.49 (0.00)	0.45 (0.00)	0.25 (0.00)	**0.53**

5.1 Discussion

When comparing the performance of our proposed system to that of one using a proactive scheduling approach, we find that in the initial scenario they perform similarly. There is some discrepancy in performance with respect to machine uptimes and utilization, but this can be explained as being due to the difference in routing that results when a fixed path of machines is set. However, as the number of parts of each type demanded increases, we see a deterioration in the performance of the proactive scheduling approach in comparison to our proposed approach. The proactive scheduling approach results greater mean part flowtimes and total order completion times. This difference in total order completion time is particularly pronounced in scenario 4 (93.5 hours to 121.0 hours).

Using a proactive scheduling approach, we see a consistent pattern in which machines have the highest utilization. We observe that utilization for machines that have been assigned only one operation is lower than those assigned more operations. This trend in machine utilization is consistent across all scenarios. The results indicate that machines m_1 and m_2 have the highest utilization and m_5 will have the lowest. In contrast, when using our proposed approach the machine utilization does not seem to adhere to a consistent pattern. As demand increases, the difference between the highest and lowest machine utilizations reduces. Also, it is important to note that the average utilization for all machines in the system is lower in comparison to that of the results from our proposed approach (42% - 54% compared to 44% - 71%). This information and the lower total order completion times suggest that our proposed approach uses its machine resources more efficiently.

The results show that the mean part wait times using a proactive scheduling approach increased in comparison to our approach. As the demand increases, we see drastic increases in part wait times. This observation is quite interesting as the comparative increase in wait times is more drastic than that for flowtimes. This is because, in our approach, it is possible to take a route which has longer processing times rather than wait. This results in parts spending more time being processed on alternate machines and less in the buffer. However, with a proactive approach, there is a fixed route and as such, there are more instances where parts are waiting for their assigned machine to become available.

The rest of this discussion will focus on the results using our proposed approach. The shortest processing time options of each part are $\{o_1, o_2, o_3\}$, $\{o_{10}, o_5, o_6\}$, $\{o_{13}, o_{14}, o_{15}\}$ for parts p_1, p_2 and p_3 respectively (without accounting for machine availability). In scenario 1, (only one order for each part is required of the system) we would expect that the highest utilization would occur with machines that can perform multiple operations that fall along the shortest processing operation sequence. These are machines m_3 (o_{12} and o_{13}), m_4 (o_3, o_5 and o_{14}) and m_5 (o_{10}, o_{15}). Looking at the machine uptime and cross referencing that information with the data from Table 2, it is clear that in most instances, the optimal route was selected (the machine uptime equals the total for each operation for each respective machine). However, the results show that m_1 has the highest utilization in this scenario. This is easily explainable when machine availability is considered. Firstly, note that machine m_1 can perform task o_{14} as an alternative to m_4 but requires a longer duration to complete it. Also note that m_4 is the only machine that can perform o_5 which is necessary for p_2. As such, the system's agents set p_3 along a less optimal part in order to favor shorter order completion time.

If the part waits for m_4 it could have to wait 6 hours for p_2 to be completed and then 6 hours to be completed for a total of 12 hours or, it can use m_1 and be completed in 11 hours. This indicates that the system is reacting and adjusting to real-time conditions and that the SA is intervening when necessary to adhere to the system's global objective.

With each successive scenario more parts of each type are required of the system. As expected, mean wait times increase as the demand increases. This is because the system now has more parts than the machines can service simultaneously. The wait time becomes more significant as the volume of parts demanded increases. The trend of durations increasing with increased demand is also observed with mean part flow times. However, note that the order of magnitude of the increase is not the same. The volume of parts demanded doubles between scenarios 1, 2 and 3. However the flowtimes only increase between 20% - 70% between scenarios. This supports the idea that utilizing the flexibility of the system by giving machines and parts autonomy over scheduling decisions could have beneficial effects in minimizing order completion time.

Whilst the trend of increasing duration with increasing demand holds true for machine, we observe that this increase does not hold true for machine utilization. As previously mentioned, there does not appear to be a consistent trend with machine utilization and this warrants further investigation. However, looking at the machine utilization, we can see that it falls between 30%-84% with value typically falling closer to 60%. These are relatively high machine utilizations which typically can cause longer queues. As such, it will be necessary to further examine the part wait times to ensure that system is providing the best balance of having the highest utilization possible without creating significant, avoidable bottlenecks within the system. This would require further investigation of PA and MA objective formulations.

Overall, from the results of our experiments, we see that with the proactive approach we have a situation where the order takes longer and uses machines less efficiently than with our proposed approach. These results, whilst preliminary, support our hypothesis that utilizing more of the system's flexibility by granting parts and machines autonomy in scheduling decisions can result in better system performance.

6 CONCLUSION

In this paper, we present a model for implementing a smart manufacturing system. The system is designed to utilize more of the flexibility in the routing and processing options available given a set of machines and parts to produce. It does this by offloading scheduling decisions to the parts and machines within the system, giving them agency. The parts make decisions on which available machines they will be processed on based on current system conditions and their objective to minimize their flowtime. Machines decide to accept jobs based on their current availability and capability, and their objective of maximizing their utilization. To ensure that these agents' local objectives do not greatly deviate from the global objective of minimizing order completion time, we implemented a supervisory agent.

As a preliminary investigation, we applied our smart manufacturing system model to a self-generated problem requiring the production of three different parts to be produced using a set of five machines that are collectively capable of fifteen operations. This investigation required the development of a multi-method simulation model consisting of agent-based modelling, system dynamics and discrete event simulation. The results of this preliminary investigation suggest that there may be potential benefits in implementing a smart manufacturing system over using conventional manufacturing systems with dynamic scheduling approaches. This benefit is in regard to reducing overall order completion time by simultaneously maximizing machine utilization whilst minimizing part wait times.

However, there is still a lot to investigate. First, there is a need to apply the model to more problems and perform more comparisons against proactive and reactive scheduling approaches in terms of factory performance measures such as wait times, flowtimes and machine utilization. This would provide a clear indication of the potential benefit and drawbacks of such a manufacturing system. Also, the current agent objectives are fairly simple and further investigation into the best objective for each agent (PA, MA and SA) is necessary. That being said, our current hypothesis is that smart manufacturing systems can provide better and more robust factory performance. The preliminary results of our investigation seem to support this hypothesis thus far.

REFERENCES

Aloulou, M. A., and Portmann, M. C. 2005. "An Efficient Proactive-Reactive Scheduling Approach to Hedge Against Shop Floor Disturbances". In *Multidisciplinary Scheduling: Theory and Applications*, 223-246. Boston: Springer.

Barbosa, J., Leitão, P., Adam, E., and Trentesaux, D. 2015. "Dynamic Self-Organization in Holonic Multi-Agent Manufacturing Systems: The ADACOR Evolution". *Computers in Industry* 66: 99-111.

Fahmy, SA, Balakrishnan, S and ElMekkawy, TY. 2008. "A Generic Deadlock-Free Reactive Scheduling Approach". *International Journal of Production Research* 46(1): 1–20.

Fazayeli, M., Aleagha, M. R., Bashirzadeh, R., and 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.

Gupta, Y. P., and Goyal, S. 1989. "Flexibility of Manufacturing Systems: Concepts and Measurements". *European Journal of Operational Research* 43(2): 119-135.

Kutanoglu, E., and Sabuncuoglu, I. 2001. "Routing-Based Reactive Scheduling Policies for Machine Failures in Dynamic Job Shops". *International Journal of Production Research* 39(14): 3141-3158.

Rahmani, D. 2017. "A New Proactive-Reactive Approach to Hedge Against Uncertain Processing Times and Unexpected Machine Failures in the Two-Machine Flow Shop Scheduling Problems". *Scientia Iranica* 24(3): 1571-1584.

Smith, R. G. 1980. "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver". *Institute of Electrical and Electronics Engineers Transactions on Computers* 29(12): 1104-1113.

Sun, J., and 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.

Wang, D. J., Liu, F., Wang, Y. Z., and 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, S., Wan, J., Li, D., and Zhang, C. 2016. "Implementing Smart Factory of Industrie 4.0: An Outlook". *International Journal of Distributed Sensor Networks* 12(1).

Wei, Y., Gu, K., Liu, H., and Li, D. 2007. "Contract Net Based Scheduling Approach Using Interactive Bidding for Dynamic Job Shop Scheduling". *In Proceedings of the 2007 Institute of Electrical and Electronics Engineers International Conference on Integration Technology*, March, Shenzhen, China, 281-286.

Xiong, J., Xing, L. N., and 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.

AUTHOR BIOGRAPHIES

SIMON LI is an associate professor in the Department of Mechanical and Manufacturing Engineering at the University of Calgary. He holds a BASc (Honours) in mechanical engineering, and MASc and PhD degrees in mechanical and industrial engineering, all from the University of Toronto. His research interests include intelligent and autonomous systems, sustainable manufacturing systems, energy engineering, HVAC systems and building energy modeling, engineering education, engineering design education. His email address is simoli@ucalgary.ca.

AKPOSEIYIFA EBUFEGHA is a PhD candidate in the Department of Mechanical and Manufacturing Engineering at the University of Calgary. His email address is akposeiyifa.ebufegha@ucalgary.ca.