Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

MULTI-AGENT SYSTEM MODEL FOR DYNAMIC SCHEDULING IN FLEXIBILE JOB SHOP SUBJECT TO RANDOM MACHINE BREAKDOWN

Akposeiyifa Ebufegha
Simon Li

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

ABSTRACT

This paper presents a model for dynamic scheduling in a smart manufacturing system that can be used in a manufacturing environment subject to random machine breakdown. We employ a multi-agent system (MAS) to schedule work on a system of machines in real-time. We propose that such a system should be less sensitive to unforeseen disruptions to the system whilst yielding good results with respect to the total flowtime for parts requested of the system. The approach employed is a completely reactive approach, and as such has the benefit of not requiring the determination of a nominal schedule. Rather, we take advantage of self-organizing nature of the MAS to guide work scheduling. To evaluate the efficacy of our proposed model, we compare its performance to that of a system using predictive-reactive scheduling to solve a furniture manufacturing problem.

1 INTRODUCTION

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 and 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 the purpose of 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 breakdown can occur randomly and at random intervals. This makes them difficult to account for whilst scheduling. They 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 type of damage to the machine as well as the resources available to direct towards the issue.

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

Literature on scheduling under uncertainty due to machine availability commonly focuses on completely reactive, predictive-reactive or robust proactive scheduling strategies. With respect to proactive scheduling, a common approach is to insert idle times into the schedule (Mehta and 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 and Portmann 2005). The issue with these proactive approaches is 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 and 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. Similar to Kutanoglu and Sabuncuoglu (2001), they found that the best policy to employ is dependent on the mean time to failure (MTTF) and meant time to repair (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 literature use 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; 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 its variability. This is because the flowtime includes the makespan.

Our proposed solution can be classified as under a completely reactive scheduling strategy. It is an extension of the model proposed by Ebufegha and Li (2021) to include machine breakdown. It involves using an MAS to schedule jobs in the facility in real-time (without need for a nominal schedule) 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 how 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 and Nof 1985; Abumaizar and Svestka 1997).

2 MODEL DESCRIPTION

The smart manufacturing system (SMS) is a cyber-physical production system. A cyber-physical production system commonly consists of a physical layer (smart machines, transporters and parts), a network layer (industrial Wireless Area Network or Local Area Network), 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. As such, we have chosen to breakdown our model into two domains: the physical domain, and the agent domain. A breakdown of our SMS model components can be seen in Figure 1. The physical domain is comprised of the physical resources. Whereas the agent domain consists of the agents that make decisions for these resources. In this model description, we will primarily focus on parts and machines. We assume that there are no other resource constraints and that all communication between the system's agents occurs in real-time.

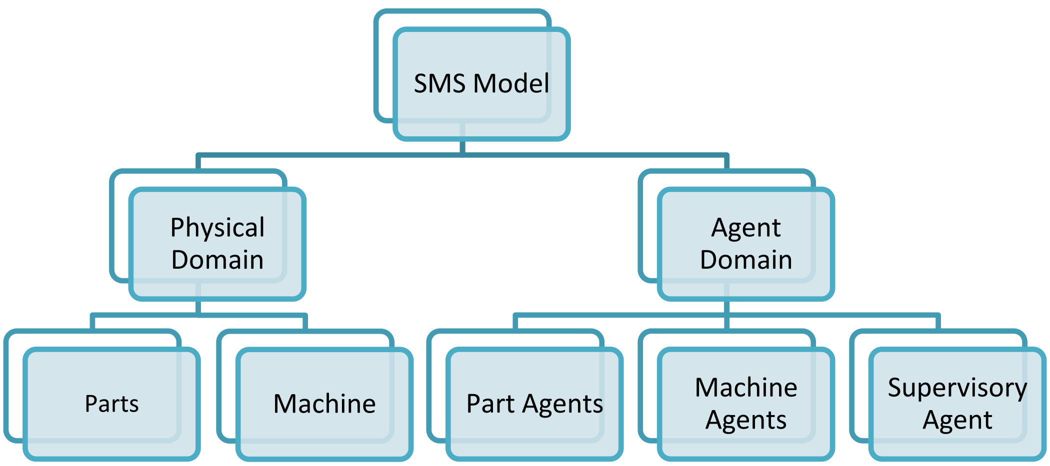

Figure 1: Overview of SMS model components.

2.1 Physical Domain of the Smart Manufacturing System

Our model focuses the system dynamics and characteristics of machines and parts. We consider these two elements (parts, and machines) to be the "core" elements of physical domain of SMS. In this section, we will provide a description of these how we modelled these elements in this research.

2.1.1 Machines

Allow $M = \{m_1, m_2, \ldots, m_m\}$ to represent the set of all m machines in the system. Also, allow $O = \{o_1, o_2, \ldots, o_o\}$ to represent the set of all operations that can be performed within a manufacturing system. Each machine in the system is capable of at least one operation and 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.

Our model allows for duplicate 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 are may either 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.

The work a machine must complete is held in the queue for that machine. When a part is sent to a machine for an operation, the part is first placed into a queue. This queue is a set of parts that have requested an operation from the machine and are waiting to be served by the machine. The order in which parts are released from the queue is dependent on the dispatching policy used by the system. In our system, we use a First-In-First-Out policy for parts in the queue. The queue length is the number of parts ahead of a given part in the queue. The queue length, along with the processing time left on the part being served by the machine, determines the amount of time that the part will have to wait to be serviced by a machine, this is the wait time. A machine will continue to perform value-adding operations on parts until its queue is empty.

The overall system dynamics for a machine are shown in Figure 2. When a machine in assigned a part to work on, the part enters into that machine's queue. It waits in the queue to be serviced by the machine. Once the part can exit the queue, the machine is setup to service the part. 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.

The key attributes that determine how well a machine performs an operation are the set up and processing times. Set up 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 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 set up times to perform the same operation on the same part type.

In real life manufacturing environments, the set up and processing times can vary from with each repetition of the same operation on the same machine. This could be due to the differences in the operator capabilities or other stochastic factors that influence operator performance (fatigue, errors, skill difference, experience, shift changes, etc.). As such these times are best represented in the form of a distribution. In this research, we have decided to represent them in the form of an exponential distribution. From our review of the literature, we found that an exponential distribution is the most commonly used. Also, an exponential distribution only requires the mean time to complete a task. This information can easily be obtained from a floor manager.

Our model allows for machine unavailability due to machine breakdown and repair. If breakdown occurs whilst a part is being worked on, we assume that the part is undamaged and will wait for the machine to be repaired to get the work it requires finished. For each machine, we assume we know the mean time to failure (MTTF) and the mean time to repair (MTTR). We represent both these variables in the form of an exponential distribution. This is typical in the literature on machine breakdown (Kececioglu 2002).

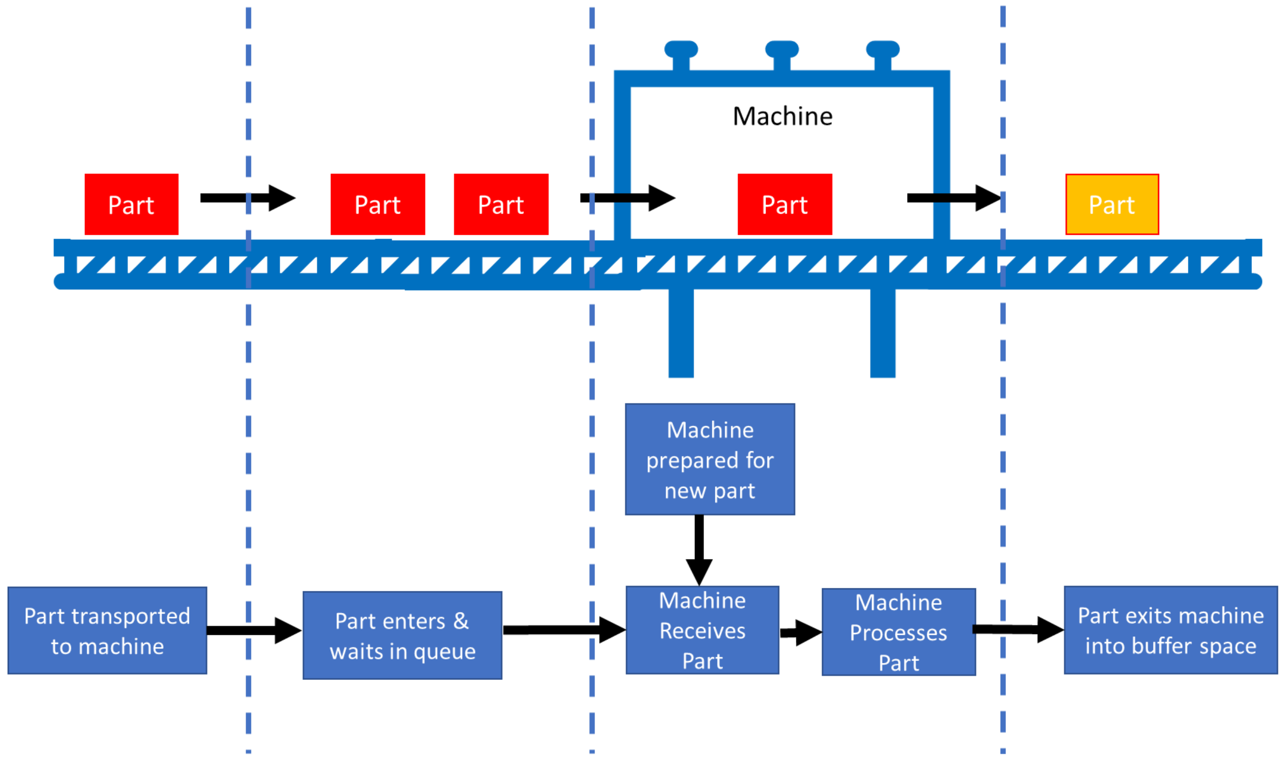

Figure 2: Machine system dynamics during normal operation.

2.1.2 Parts

We represent the set of distinct parts (or different part types) that can be produced by a system using $P = \{p_1, p_2, \ldots, p_p\}$. 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. However, each part may have multiple operation sequences that can be used to produce them. Each of these operation sequences (processing routes) can be used interchangeably during the production cycle. For example, Figure 3 shows the possibilities for work in process (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 by a change in color). In this example, 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).

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. 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 with respect to availability make it viable.

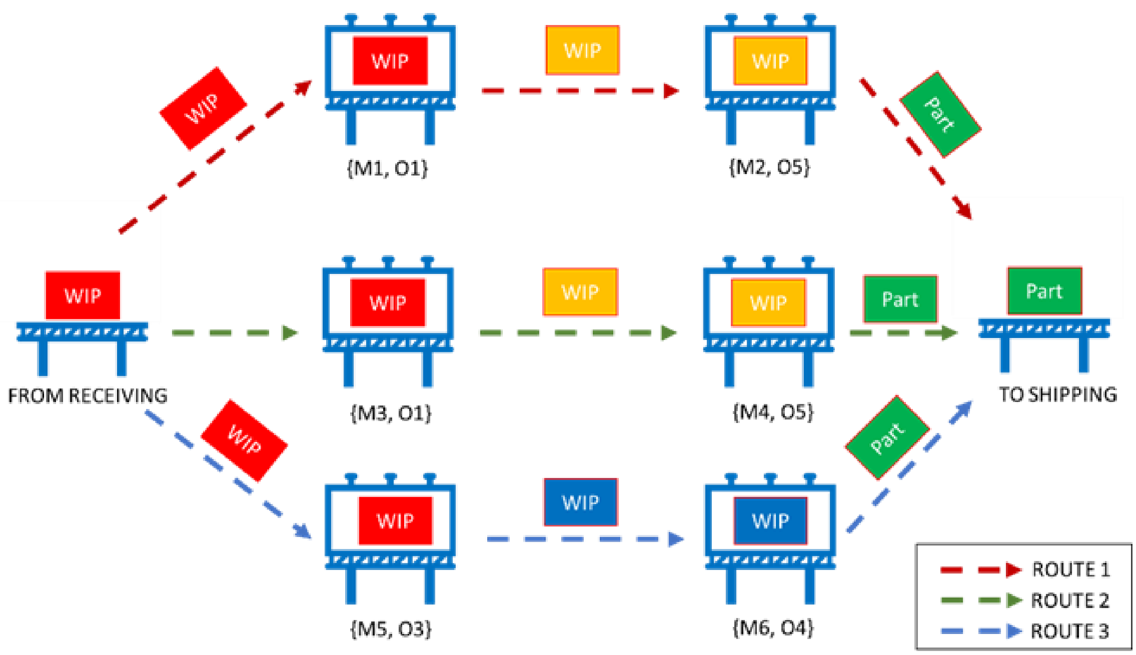

Figure 3: Machine route options for sample part in 6 machine system.

2.2 Agent Domain of the Smart Manufacturing System

The approach we use employs a hybrid control architecture which combine elements of heterarchical and hierarchical control. It is developed based on contract net protocol (Smith 1980) and the extension to contract net protocol presented by Wei et al. (2007). We use a hybrid control architecture as it 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). This protocol is similar to an auction, with agents auctioning and bidding on operations. An overview of each agent's functions, inputs and outputs can be seen in Figure 4. Figure 4 also shows the information flow between each agent.

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 propose using an agent-based approach. This requires determining the types of agents present in the system and the rules that govern their interaction. There are three types of agents that exist in our model: (1) part agents (PA), (2) machine agents (MA), and (3) the supervisory agent (SA).

For each part that is requested from the system, we have a PA. The PA's objective is to minimize the total flow time for the part it represents. This is the total of the time spent in transit (*TT*), and the time spent waiting (WT) and being processed (PT) on each machine in the system. We model the PA to be analogous to a subcontractor. It issues tenders for the next operation required by the part the PA represents. These tenders are bid on by the MA's who then return their bids. These bids are in the form of the estimated flow time (*FT*) required to complete the operation requested on the bidding machine. This is a summation of the estimated transfer times (*TT*), wait times (*WT*), processing times (*PT*), and set up times (*ST*). Once all bids are received, the PA decides the machine it wants to assign work to as well as a ranked list of alternate machines. This decision is made by selecting the lowest bid. The objective function that guides the PA decision making is as follows:

$$\min FT_P$$

$$FT_P = TT_P + \sum_{i=1}^{m} \left(WT_{ip} + ST_{ip} + PT_{ip} \right)$$

Each machine in the system has its own MA. The MA is analogous to a contractor in that it bids on the operations requested by all PA's on behalf of its machine. Its objective is to maximize the its machine's machine utilization (*MU*). As such, it tries to increase the amount of work assigned to its machine whilst minimizing idle time. MA's are free to bid on work only if the machine is available, otherwise they cannot bid on work. A machine is considered to be unavailable if the machine is down due to break down and waiting for repairs to be completed. The objective of the MA is as follows:

$$\min MU_m$$

The SA is analogous to a referee. If there is a conflict (i.e. two PA's awarding work to the same machine), the SA intervenes. Its objective is to ensure that system does not deviate too much from the global objective of minimizing total flowtime. To do this, the SA 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 (*p*) currently in the system whenever there is conflict. The objective function the SA uses is as follows:

$$\min\left(\max\left(FT_1, FT_2, \ldots FT_p\right)\right)$$

The auction process is continuous, with PA's initiating and closing tenders for work, and MA's bidding on work with SA's supervising all decisions. If at one point in time. 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.

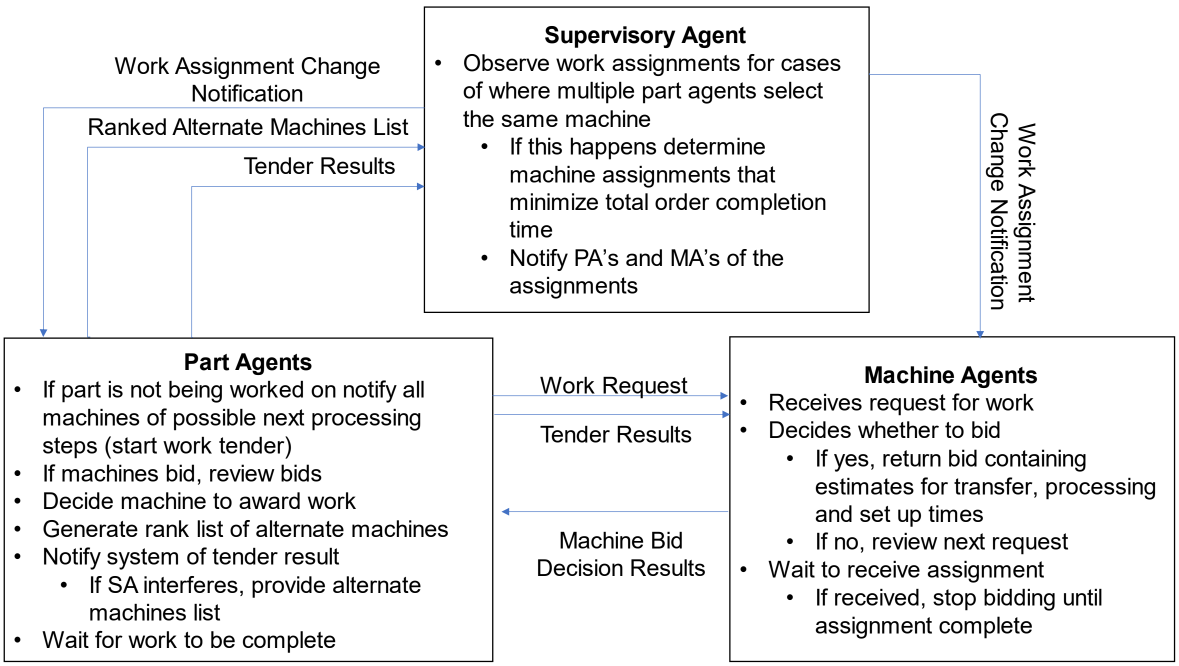

Figure 4: Agent behavior (Ebufegha and Li 2021).

3 SIMULATION EXPERIMENTS – EXAMPLE PROBLEM DESCRIPTION

In this study, the example problem used is based on a furniture manufacturing facility presented by Suzic et al. (2012). This facility consists of eleven (11) machines that can be 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. For the sake of this study, we focus purely on the parts being ordered from the system and ignore the products that can be assembled from the requested parts.

We have made two modifications to the base problem. The first is that we assigned each operation a distribution to represent its setup and processing times for each machine. The second modification is assigning each machine a MTTR and MTTF values. All machines are assumed to have the same MTTF and MTTR. The variables are represented using an exponential distribution. There are three levels for both variables; low, medium, and high. These mean values assigned to these levels can be seen in Table 1 below:

Table 1: High, medium and low settings for MTTF and MTTR (in time units).

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

In the scenario we examine, an order of six distinct types of parts is requested from the two systems consisting of the same set of machines at time zero (T = 0). These are parts p_1, p_2, p_3, p_4, p_5, and p_6. The machines in both systems are in the same layout, and have the same capabilities with the same performance. One system using predictive-reactive scheduling. First a nominal schedule is designed with the objective of minimizing the total flowtime for all parts requested. In the event of a disruption, the affected operations are right-shifted to the time when the machine becomes available again. The second system employs our proposed approach. Here, the schedule is determined in real-time using our MAS. Both systems are tasked to process the order, and are subject to the same machine breakdown conditions. Their performance is compared with respect to their order completion times and the variability in their performance.

3.1 Simulation Experiment Conditions

Simulation experiments were run for the predictive-reactive schedule, and our MAS-based approach. The simulations for both systems are run using a script developed in MATLAB R2021a. The conditions for the experiments are as follows:

- We execute a full factorial (3^2) experiment
    - All 9 combinations for levels of MTTF and MTTR are run
- Each experiment has 100 repetitions

3.2 Experiment Results

Table 2 shows the results for the simulation experiments run using both scheduling approaches. The key measures used to evaluate the results of the experiments are the mean and standard deviations. An analysis of variance (ANOVA) has been performed on the means to establish whether or not the means are the same.

Figure 5 shows the main effects plots for the simulation experiments. The top left plot depicts the effect of scheduling approach on the mean completion time for each different level of MTTF. The top right plot depicts the effect of scheduling approach on the standard deviation of completion time for each different level of MTTF. The bottom left plot depicts the effect of scheduling approach on the mean completion time for each different level of MTTR. The bottom right plot depicts the effect of scheduling approach on the standard deviation of completion time for each different level of MTTR.

Table 2: Simulation experiment results for machine breakdown.

MTTF Level	MTTR Level	Predictive-Reactive Approach		MAS Approach		P-Value
		Mean (μ_1)	Standard Dev. (σ_1)	Mean (μ_2)	Standard Dev. (σ_2)	(H_0: $\mu_1 = \mu_2$)
LO	LO	13.26	1.80	12.79	2.23	0.052
MED	LO	14.47	2.23	13.65	2.70	0.011
HI	LO	16.53	2.59	16.27	3.17	0.640
LO	MED	36.88	17.27	31.21	11.81	0.004
MED	MED	46.48	17.11	33.86	11.52	0.000
HI	MED	60.78	13.57	51.24	16.53	0.000
LO	HI	63.11	33.07	49.26	24.53	0.000
MED	HI	74.72	29.36	64.01	30.82	0.007
HI	HI	118.58	34.42	97.60	37.13	0.000

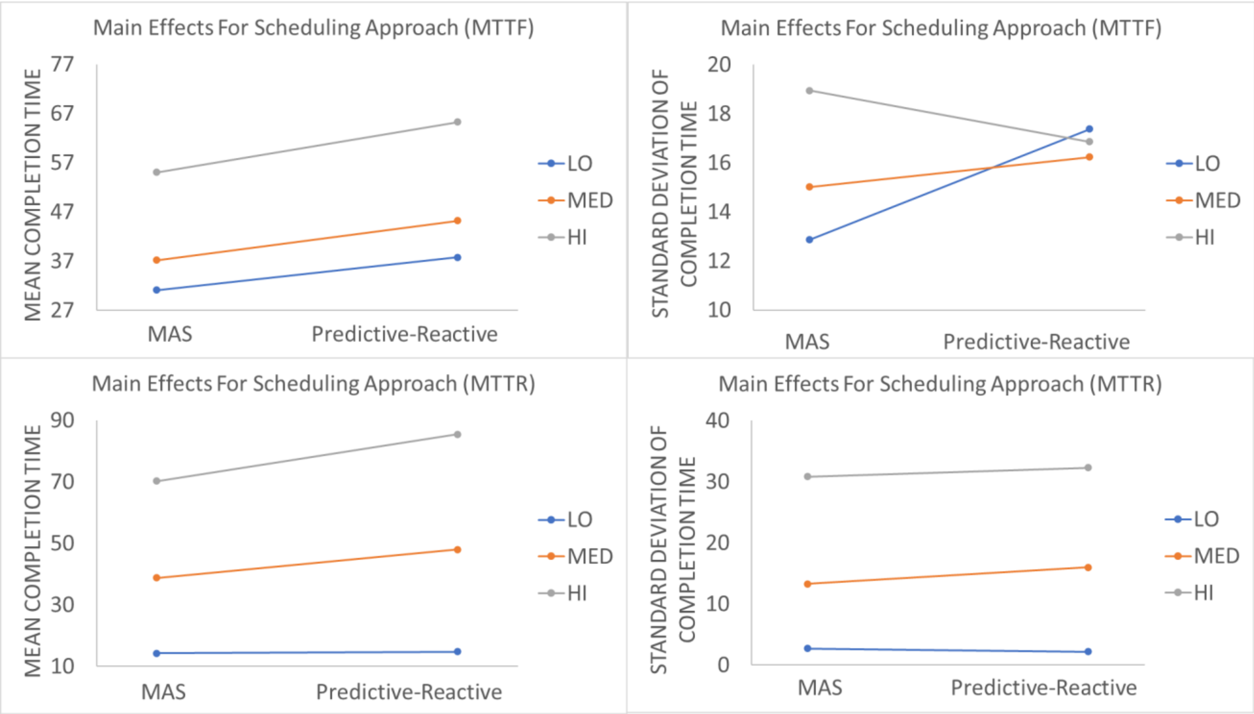

Figure 5: Main effects plots derived from simulation experiment data.

3.3 Discussion

The results of our experiments, as seen in Table 2, show that our proposed approach to dynamic scheduling yields lower mean completion times in all simulation scenarios examined for the part demand given. There is no discernable pattern with respect to the standard deviations. In some instances, the standard deviation is higher for our proposed approach that with the predictive-reactive approach. On the surface, this suggests that our proposed approach outperforms predictive-reactive scheduling for this specific problem.

An analysis of variance (ANOVA) was performed to confirm our initial findings. We used a null hypothesis of means being equal to determine whether the difference between the MAS-based approach and the predictive-reactive approach was statistically significant. The results of the ANOVA show that when MTTR is medium or high, the difference in the mean completion times are statistically significant ($P < 0.05$). This is true regardless of the MTTF level. However, this is not the case when MTTR is low. When MTTR is low, the statistical significance of the difference in mean completion times is dependent on MTTF level. With low MTTR and MTTF, the difference is the means is not statistically significant. However, this is a borderline case with $P = 0.052$. This result suggests that more information is needed to draw any conclusions. With low MTTR and medium MTTF, we see that the difference in the mean completion times is statistically significant ($P = 0.011$). With low MTTR and high MTTR, we see that the difference in the means are not statistically significant with $P = 0.640$ ($P>0.05$).

The main effects plot for completion times (seen in Figure 5) show that regardless of the setting for MTTF (low, medium or high), the MAS approach outperforms the predictive-reactive scheduling approach used. The slope for all three settings are basically parallel, suggesting that there is no interaction between the factors. We see that as MTTF increases, the completion time increases. The main effect plot for mean completion times looking at MTTR, suggests that our approach yields lower completion times regardless of the level of the MTTR. There is an interaction between the level of the MTTR and the scheduling approach used. The plots suggest that the MAS-based approach is less sensitive to changes in the level of MTTR. This can be seen as the slope increases as the level of MTTR increases.

Examining the main effect plots for the standard deviations for the completion times with respect to MTTF, we see that our MAS-based approach is more sensitive to changes in the level of MTTF than the predictive reactive approach. We can also see that the mean of the standard deviation is lower for the MAS approach than that for the predictive-reactive approach. This means that we should expect lower standard deviations using MAS but it will increase rapidly with MTTF level. The main effect plot for standard deviation of the completion times with respect to the MTTR show that the MAS approach has lower standard deviations. It also suggests that the MAS approach standard deviation is less sensitive to the level of MTTR than the predictive-reactive scheduling.

Overall, the results of our experiments suggest that the MAS approach we propose outperforms predictive-reactive scheduling in a manufacturing environment subject to random machine breakdown with respect to schedule stability and performance for the problem being examined. Our analysis suggests that the MAS scheduling approach yields lower mean completion times, and standard deviations. However, it is important to note that these results are preliminary. As such, it cannot be definitively said that our MAS-based approach would outperform predictive-reactive scheduling in most cases. Rather, the results of our experiments suggest that there are conditions under which our approach outperforms predictive-reactive scheduling. As such, further research is required.

4 CONCLUSION

In this paper, we present a MAS model for smart manufacturing systems that can be used for manufacturing environments subject to random machine breakdown. The model presented primarily focuses on dynamics of the parts and machines within system and agent intelligences that govern their interactions. Our intent was to develop a robust system capable of handling disruptions due to uncertainties arising from machine availability. The objective of our proposed model is to minimize the total flowtime for parts in the system and the variability in the system performance with different levels of uncertainty.

As a preliminary investigation, we compared the performance of our proposed system to one using predictive-reactive scheduling. Both systems were comprised of the same machines in the same layout, and were subject to the same machine reliability. Both systems were required to produce the same order of parts with the order arriving at each system at time zero ($T = 0$). The results of the investigation suggest that the MAS-approach outperforms the predictive-reactive scheduling approach with respect to stability and performance.

However, it is important to note that these results only suggest that our approach outperforms predictive-reactive scheduling under the conditions of the specific problem being examined. To draw a more conclusive observations, more experiments need to run under more varied experimental conditions (different problem sizes, different order mixes, different numbers of duplicate machines, etc.). That being said, it is our current hypothesis that our proposed approach is a viable approach from developing a robust manufacturing system.

REFERENCES

Abumaizar, R.J., and Svestka, J.A. 1997. "Rescheduling Job Shops Under Random Disruptions". *International Journal of Production Research* 35(7): 2065-2082.

Ahmadi, E., Zandieh, M., Farrokh, M., and Emami, S.M. 2016. "A Multi-Objective Optimization Approach for Flexible Job Shop Scheduling Problem Under Random Machine Breakdown by Evolutionary Algorithms". *Computers and Operations Research* 73: 56-66.

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.

Ebufegha, A. and Li, S. 2021. "Multi-Agent System Model for Dynamic Scheduling in Flexible Job Shops". In *Proceedings of the 2021 Winter Simulation Conference,* edited by S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, 1-12. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

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.

Kececioglu, D. 2002. "Reliability Engineering Handbook". Lancaster, Pennsylvania: DEStech Publications, Incorporated.

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.

Mehta, S.V., and Uzsoy, R.M. 1998. "Predictable Scheduling of a Job Shop Subject to Breakdowns". *Institute of Electrical and Electronics Engineers Transactions on Robotics and Automation* 14: 365-378.

Merdan, M., Moser, T. Vrba, P., and Biffl, S. 2011. "Investigating the Robustness of Re-Scheduling Policies with Multi-Agent System Simulation". *International Journal of Advanced Manufacturing Technology* 55 (1-4): 355-367.

Moratori, P., Petrovic, S., and Vázquez-Rodríguez, J.A. 2010. "Integrating Rush Orders Into Existent Schedules for a Complex Job Shop". *Applied Intelligence* 32(2): 205-215.

O'Donovan, R., Uzsoy, R.M., and McKay, K.N. 1999. "Predictable Scheduling of a Single Machine with Breakdowns and Sensitive Jobs". *International Journal of Production Research* 37: 4217-4233.

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 20th - 24th, 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.

Yamamoto, M., and Nof, S.Y. 1985. "Scheduling/Rescheduling in the Manufacturing Operating System Environment". *International Journal of Production Research* 23(4): 705-722.

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.