Uniport Journal of Engineering and Scientific Research (UJESR)

Vol. 6, Issue 1, 2021, Page 26-34
ISSN: 2616-1192

(www.ujesr.org)

Modelling an Agent-Based Job Shop Scheduling for Makespan Optimization in a Serial-Parallel Machine Setup

Chiagunye, T.T
Computer of Engineering Department, Michael Okpara University of Agriculture, Umudike, Abia State, Nigeria.
*Corresponding author's email: tchiagunye2@yahoo.com

Abstract
This research is concerned with the modeling of agent-based job shop scheduling to optimize a serial-parallel machine setup. The model agent-based job shop scheduling involved three sequential machines through which every order must pass followed by one out of three finishing machines used, one per finishing type. It was mandatory for the type of aluminum sheets being produced that the raw material be passed through the first three machines only in one order. Thus, the model developed took this sequential order into consideration. This model was executed using a combination of Markovian process, in working out the state of the machine and agent-oriented analysis that adjusts to the dynamics of the stochastic order processes. A well-crafted scheduler agent carries out bunching of sorted jobs either in 1 or 2 or 3 days' bunch (es) per finishing type and selects the best out of the three approaches. This scheduling technique allows a certain product type to be scheduled for 1 or 2 or 3 days before changing to another product type. The comparative result shows that the modelled agent-based job shop scheduling had 2.4% improvement to the existing classical model and should be applied in an industrial set up for optimum machine usage and customer satisfaction. The simulation results were also used to determine the optimum scale of plant, given the rate of order arrival per month.

Keywords: *Job shop scheduling, Serial-parallel machine, Makespan, Optimization, Agent based*

Received: 24th June, 2021 *Accepted: 19th October, 2021*

1. Introduction

The scheduling and planning of production order have an important role in manufacturing system. The diversity of products, increased number of orders, size of workshops and expansion of factories have made the issue of scheduling production orders more complicated, hence the traditional methods of optimization are unable to solve them. Past works have shown that most approaches to job-shop scheduling assume complete task knowledge and search for a centralized solution. These techniques typically do not scale with problems size, suffering from an exponential increase in computation time. The centralized view of the plant coupled with the deterministic algorithms characteristic of these schedulers do not allow the manufacturing processes to adjust the schedule, using local knowledge to accommodate disturbances such as machine breakdowns. This requirement puts the problem in the class of agent-based model (ABM). Hence this work adopts an alternative view on job-shop scheduling problem where each resource is equipped with adaptive agent.

Scheduling, understood to be an important tool for manufacturing and engineering, has a major impact on productivity of a process (Blazewicz et al. 1997). In manufacturing, the purpose of scheduling is to minimize the production time and cost, by telling a production facility what to make with which staff, and on which machine. Cited publications argued that agent-based modelling is used because only agent-based model can explicitly incorporate the complexity arising from individual behaviour and interactions that exist in the real-world. Nüske et al. (2017) worked on and represented ten well-known simulation models as a time homogenous Markov Chain. The author's idea is the formulation of the system stochastic transition as the state space of the Markov Chain. Despite the fact that all the information of the dynamics on the ABM is encoded in a Markov chain, it is difficult to learn directly from this fact, due to the huge dimension of the configuration space and its corresponding Markov transition matrix. The work of Nüske et al. (2017) mainly relies on numerical computations to estimate the stochastic transition matrices of the models. Shoukat and Moghadas (2020) and Sponer et al. (2018a) contributed to interweaving Markov Chains and ABM. The former represents the simplest form of a stochastic process while the latter puts a strong emphasis on heterogeneity and social interactions. In the model by Shoukat and Moghadas (2020) homogeneous mixing leads to a macroscopic Markov chain which underlines the theoretical importance of homogeneous mixing. An important prospect that is not exploited by Shoukat and Moghadas (2020) concerns the measure of practical emergence or discrepancy, the gap between the macro-structural properties of a system and internalized rules or intentions of the individual agents. The measure of this gap should lead to more elaborated gauges whose dynamics themselves call for new specific investigation.

Hallier and Hartmann (2016) showed how to construct Markov state models that approximate the original Markov process by a Markov chain on a small finite state space and represent well the longest time scales of the original model. The approach extracts the aggregated long-term dynamics of reversible Markov chains. The macrostates as well as transition probabilities between them can be estimated on the basis of short-term trajectory data. Apparent advantages of a reduced state space are that it is easier to compute eigenvalues and eigenvectors as well as other properties such as waiting times. One limitation to the Hallier and Hartmann (2016) is that the approach and its analysis depends on the original Markov chain that represents an agent-based model of interest, to be reversible. In general, it will be difficult to say whether it is reasonable to assume that an agent-based model results in a reversible Markov chain. One reason for this difficulty is that, if we estimate the transition matrix from simulated trajectory data, it does not need to fulfill the detailed balanced equation, even if the underlying Markov chain is reversible (Liu et al., 2019; Sponer et al., 2018b). Beyond that, an approach that applies also to non-reversible Markov dynamics need to be exploited. There are few approaches that apply to non-reversible Markov chains (Ward and López-García, 2019). Hameed and Schwung (2020) suggested a graph-theoretical framework for constructing reversible surrogates of non-reversible dynamics, based on a cycle decomposition of the underlying Markov chain. However, the application to agent-based models was not treated. Therefore, the construction of Markov state models for general agent-based models is still an open problem. Rüdrich et al. (2017) modeled the job shop scheduling problem by means of a multi-agent reinforcement learning and attached to each resource an adaptive agent that makes its job dispatching decisions independently of the other agents and improves its dispatching behaviour by trial and error employing reinforcement learning algorithm. Hallier and Hartmann (2016) gave some suggestions of state feature selection but did not consider whether these features are memoryless. The embedded Markov chain is also not mentioned in their work.

Zhang et al. (2017) modeled a real-time job shop scheduling based on simulation and Markov decision processes. The main task is to decide which job in a queue should be processed next. The model uses two algorithms, simulation-based value iteration and simulation-based Q-learning were introduced to solve the scheduling problem from the perspective of a Markov decision process. The real-time job shop scheduling model by Zhang et al. (2017) is a sequential decision-making optimization technique. The system contains five (5) machines and produces two (2) products with two (2) operation flows. The operational flow of job on a machine in this model is not constrained to pass through each machine in series. Hence, this research work presents a model where jobs must pass through the first three machine in series before the one-out-of-three finishing parallel machine.

2. Materials and methods

2.1 Materials

A combination of Markovian process, in working out the state of the machine and agent-oriented analysis that adjusts to the dynamics of the stochastic order processes were used in the analysis of the proposed agent-based model for the job shop scheduling. Using Markov chains, the amount of wastes in the production process were ascertained. Also, the machine states were evaluated, and cost of repairs and general machine maintenance were factored into the production costs. Scheduler agent that uses a carefully crafted algorithm to schedule incoming orders for production was developed at the second section. The production agent that produces according to the schedule except when interrupted by routine maintenance or machine breakdown which introduces some delays was developed. The release agent section was developed which ensures that orders are released as fast as possible.

The production process requires three machines in sequential order through which every raw material input must be processed and one-out-of-three finishing machine used, one per type of product. This is so because the production arrangement cannot be in parallel as the order must first be processed in machine one (1) before moving over to machine two (2). The same process has to be done on machine two (2) before machine three (3) in sequential order.

The case study company uses first come first served basis to schedule their jobs. This means that a small job may take unduly long before it may be delivered if it is positioned at the back of the queue and bigger jobs are at the front. This suggests that orders may be sorted in increasing order of size to accommodate smaller jobs first and increase the proportion of jobs that would be processed quickly. This was achieved using agents as presented in the proposed system architecture shown in Fig. 1.

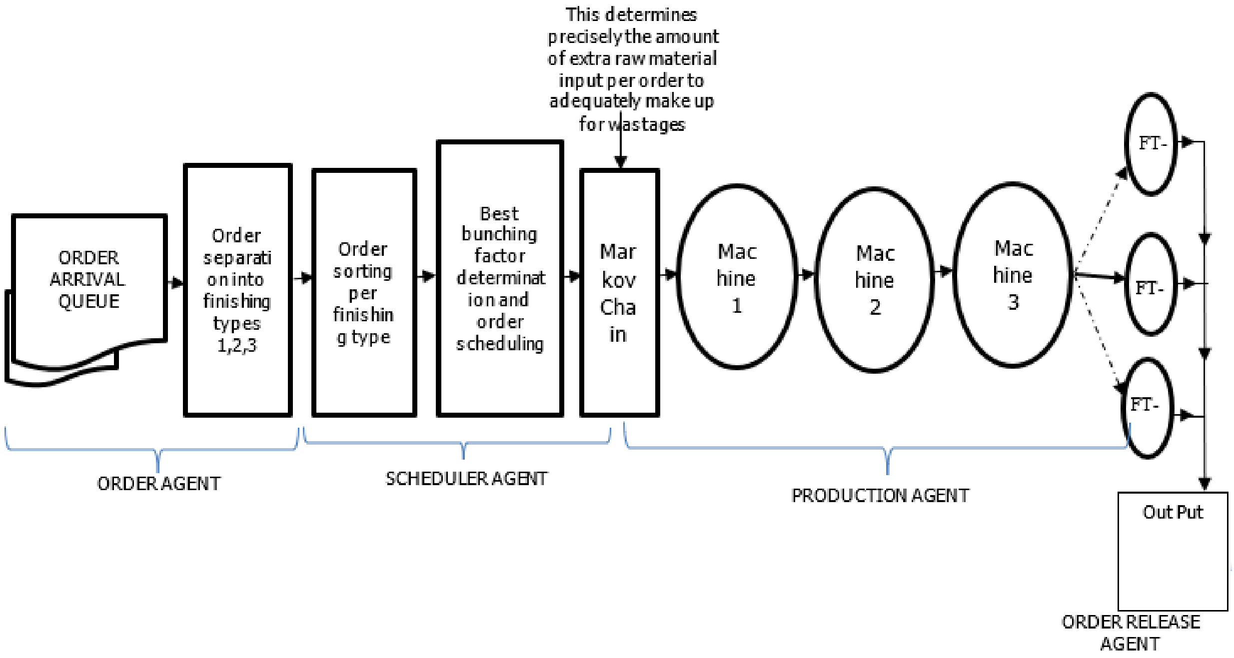

Fig. 1: Architecture for the proposed System.

2.2 Method

2.2.1 Schedule algorithm

A carefully worked out procedure used to achieve the set objectives is as follows:

a. The system sorts the entire order into three parts according to the type of finish desired of each. All the orders of finishing type one are together, finishing type two are together and finishing type three are together after the sorting.
b. Each finishing type is again sorted in ascending order of size in kilograms.
c. Thereafter the scheduler agent schedules the orders as follows: the first of type one is followed by the first of type two, followed by the first of type three. Then the second of type one follows in the schedule and after that, the second of type two and the second of type three, and so on.
d. Because the machine must be kept as busy as possible, slacks are introduced into each finishing type to ensure that the production of that type occupies only full days. Thus, once the machines start producing a particular finishing type, it must continue with that type throughout the working day before it can change to another type at the beginning of the next day if need be.
e. Simulate a schedule with bunching factor of 1, this means one type of finish is done each day, example:

   Day 1 = finishing type 1

   Day 2 = finishing type 2

   Day 3 = finishing type 3

   Day 4 = finishing type 1 and so on.

f. Simulating the scheduling using a bunching factor of 2 that is:
   Finishing type 1 = first two days
   Finishing type 2 = days 3 and 4
   Finishing type 3 = days 5 and 6
   Finishing type 1 = days 7 and 8 and so on.

g. Simulate the scheduling using a bunching factor of 3 this means:
   Finishing type 1 = days 1, 2 and 3
   Finishing type 2 = days 4, 5 and 6
   Finishing type 3 = days 7, 8 and 9
   Finishing type 1 = days 10, 11 and 12 and so on.

h. Select the bunching factor that yields the earliest finishing date for the order and use that bunching factor for scheduling the order.

2.2.2 Scheduler agent flow chart

Fig. 2 shows the activity flow chart for the scheduler agent. The scheduler follows this established algorithm to schedule the orders for production.

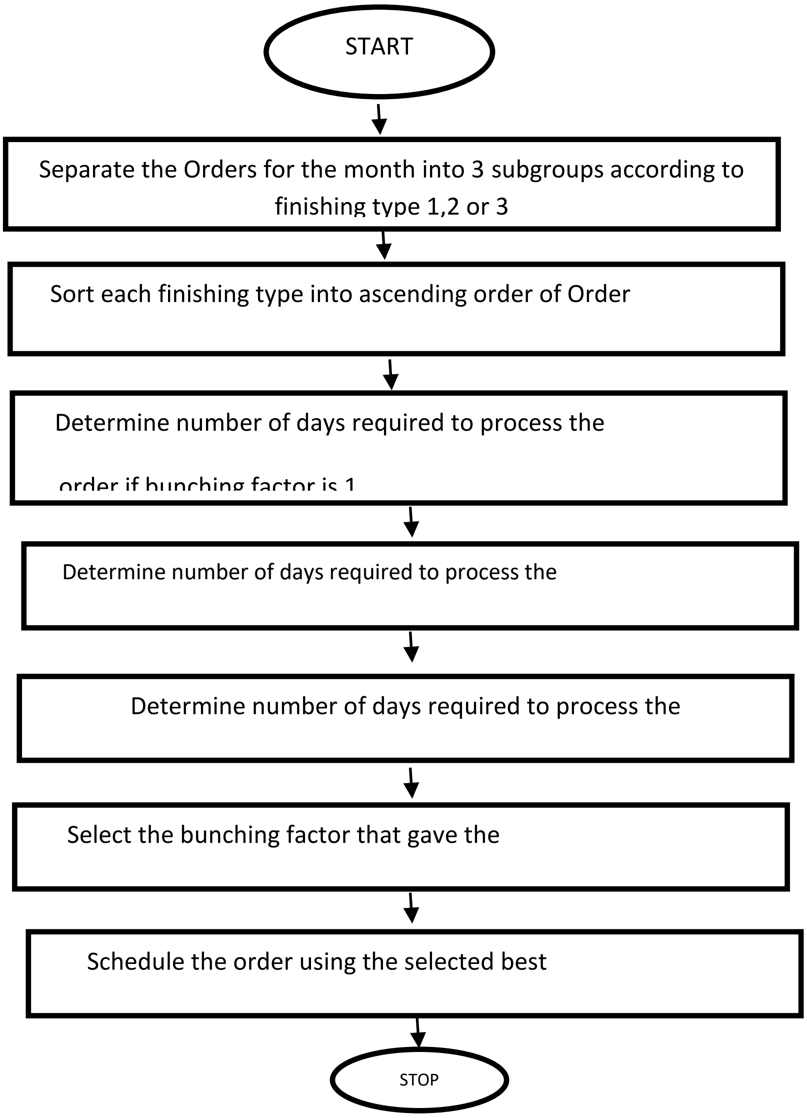

Fig. 2: Scheduling agent's activity flow chart

2.2.3 Algorithm for achieving the developed model

The proposed model seeks to obtain an agent-based scheduler that is optimized for handling job shop scheduling that ensures efficient and profitable manufacturing automation. The activities carried out to achieve the aim of the research work were:

1. The orders gotten from the customers for thirty (30) days were grouped into three different finishing types.

2. Each finishing type was sorted in ascending order of job size with respect to the finishing type before scheduling.
3. Bunching of each finishing type of job with a bunching factor (Bf) of 1 or 2 or 3 was used to schedule the job.
4. Selecting the best bunching factor for each order, this means the bunching factor that gives the earliest finishing time for all the orders (See Table 1).
5. Test running the carefully crafted algorithm on ten (10) separate orders.
6. Scheduling with the best bunching factor (Bf) for each of the ten different orders and that led to the latest finishing dates at the bottom of Table 1.

3. Results and discussion

3.1 Scheduling of Job Order Using Bunching Factors 1, 2 and 3

Bunching technique was adopted in this model to schedule job for processing. Bunching of the whole order queue with bunching factor (Bf) of 1 or 2 or 3 to determine the best bunching that gives the earliest finishing time or minimum makespan for all the orders. Table 1, shows the schedule result for ten (10) different orders, scheduled using Bf1, Bf2 and Bf3. Because of the stochastic nature of the order arrival, the best bunching factor may change with each order, for example, in an empirical study involving ten (10) different sets of orders (Table 1) the bunching factor of two (2) gave the best result in eight out of the ten (10) sets of orders while the bunching factor of three (3) gave the best result in two (2) out of the ten (10) sets of orders. The bunching factor of one (1) is consistently the worst-case scenario in all the ten (10) sets of order. To make it clearer, consider order one (1) in Table 1, the table shows that all the orders that need finishing type one (1) will be finished in 100 days using bunching factor 1, but 98 days using bunching factor 2 and 102 days using bunching factor 3. Also, all the orders that require finishing type 2 will be completed in 50 days using bunching factor 1, or 52 days using bunching factor 2 and 51 days using bunching factor 3. Similarly, all the orders requiring finishing type 3 will be completed in 84 days using bunching factor 2 but will take as much as 90 days if bunching factor 3 were used. In this scenario the best bunching factor is the one with the least number of days for completing the last job in a given order queue. Because the different finishing types in one set of orders do not have the same number of jobs, a finishing type may finish before others. For example, for order number 1 using bunching type (Bf1), finishing type 1 was the last to be processed up to the 100th day, finishing type 2 finished on the 50th day while finishing type 3 finished on the 84th day. In order 1 therefore, a bunching factor of 2 that finished the work in an order queue in 98 days is superior to bunching factor 1 that finished the work in an order queue in 100 days. The bunching factor of 3 gave the worst-case scenario for this order requiring 102 days to complete the order. Thus, the best is bunching factor 2 as shown in Table 1. Consider another example from Table 1 where bunching factor 3 is the best out of the three possible bunching factors. Consider order number 7, the latest finishing time to complete the order for bunching factor 1 is 157 days, that of bunching factor 2 is 152 days but the bunching factor 3 will get the work done in 147 days. Thus, the bunching factor to use when scheduling order number 7 is bunching factor 3.

Table 1: Schedule result for ten (10) different order with Bf1, Bf2 & Bf3

Order	Finishing Type	Bf_1 (days)	Bf_2 (days)	Bf_3 (days)	Best
1	1	100	98	102	2
	2	50	52	51	
	3	84	84	90	
2	1	118	116	120	2
	2	95	94	96	
	3	57	60	63	
3	1	134	128	129	2
	2	59	58	60	
	3	87	90	90	
4	1	97	92	93	2
	2	59	58	60	

	3	93	90	99	
5	1	166	164	156	3
	2	8	10	6	
	3	84	84	90	
6	1	40	38	39	2
	2	101	100	105	
	3	102	102	108	
7	1	157	152	147	3
	2	131	130	132	
	3	42	42	45	
8	1	97	92	93	2
	2	44	46	42	
	3	147	144	153	
9	1	115	110	111	2
	2	68	70	69	
	3	66	66	72	
10	1	118	116	120	2
	2	59	58	60	
	3	69	72	72	

The result of 10 different set of orders (Table 1) shows that bunching factor two (Bf2) has the smallest finishing times for orders 1,2,3,4,6,8,9 and 10; bunching factor three (Bf3) just had a better result in 5 and 7 while bunching factor one (Bf1) had none. A careful application of this bunching technique will help save time and cost in every industry that receives stochastic order. The graphs of Fig. 3 and 4 as shown were used to illustrate the performances of the three bunching factors.

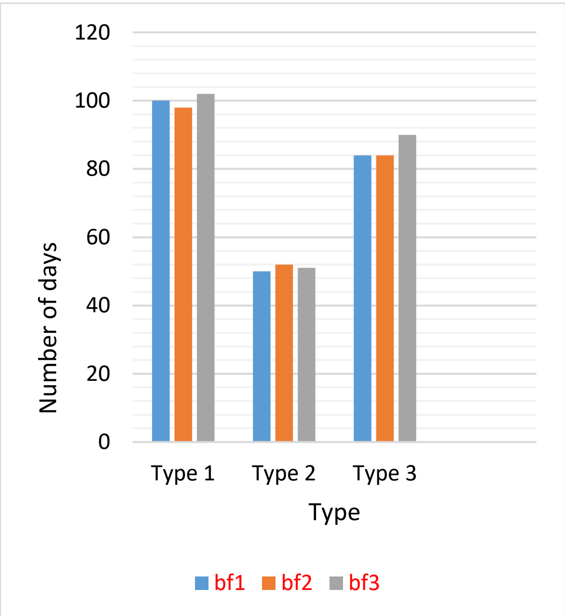

Fig. 3: Bar chart of Order 1 as bf varies from 1-3 (Use Bf in Fig. for uniformity)

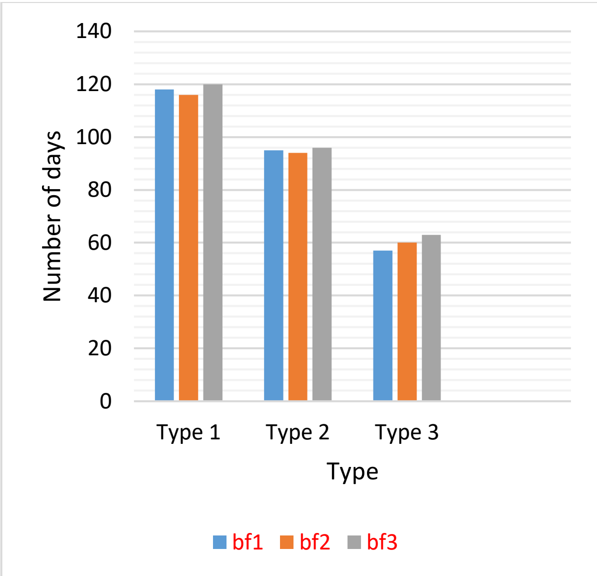

Fig. 4: Bar chart of Order 2 as Bf varies from 1-3 (Use Bf in Fig. for uniformity)

The release dates for orders received in one month and then scheduled is shown in Fig. 3. Referring to Table 1, the orders are scheduled using three different bunching factors (Bf1, Bf2 and Bf3) for the three finishing types. From Fig. 3, the finishing type 1 has Bf1 as 100 days, Bf2 as 98 days and Bf3 as 102 days; finishing type 2 has Bf1 as 50 days, Bf2 as 52 days and Bf3 as 51 days, while finishing type 3 has Bf1 as 84 days, Bf2 as 84 days and Bf3 as 90 days. The result shows that Bf2 had the earliest due date to complete the last operation, with the latest due date for the last release as 98 days while Bf1 has 100 days and Bf3 has 102 days. Similar thing happened in the second order of Fig. 4, with Bf2 having earliest due date for the complete process as 116 days while Bf1 uses 118 days and Bf3 uses 120 days to complete the process.

3.2 Release dates for sorted and unsorted order

Tables 2 and 3 show the results obtained when scheduling ten batch of orders that came on ten different months in an unsorted form (i.e., conventional method used by the company) and in a sorted form (i.e., using the agent-based approach). The finishing times for the unsorted orders represent what will happen if the orders were processed on first come first served (FCFS) basis. The finishing times for the sorted orders represent what the proposed ABM would achieve when the most favourable bunching factor is applied. This comparison is important because it shows the difference in the results obtained when the company used the unsorted approach which is now improved upon by the introduction of sorting and bunching factor in this model. Each finishing type machine has a capacity per day in kilograms. For example, finishing type one machine has a capacity of 15kg per day, finishing type two machine has a capacity of 19kg per day and finishing type three machine has a capacity of 30kg per day. By looking at the finishing type demanded by the customer and the capacity per day of the machine that produces it, the number of days each customer order will take to process is determined. The interesting point with sorting of order by the agent according to levels is on the area of customer satisfaction. The release dates for the sorted order meets the lead time. The sorting arrangement helps to clear undue delay of small order beyond the lead time.

Table 2: Release date for order 1-10 scheduled on first come first served (FCFS)

M 1	**M 2**	**M 3**	**M 4**	**M 5**	**M 6**	**M 7**	**M 8**	**M 9**	**M 10**
4	2	5	1	2	1	4	4	2	2
9	6	11	5	10	6	5	7	2	6
11	9	14	8	13	12	10	11	5	7
14	11	19	10	15	17	15	16	9	11
17	15	23	12	16	20	18	19	12	12
18	18	26	15	18	21	20	22	16	14
19	19	28	16	19	23	27	23	23	19
22	21	31	17	25	24	30	26	28	20
25	25	34	19	29	25	34	29	33	21
29	26	37	20	32	26	40	33	37	24
30	29	39	33	34	28	43	35	39	30
32	32	42	36	36	30	47	37	40	31
34	34	46	39	39	33	54	44	42	32
35	39	46	44	43	35	60	47	45	33
37	41	49	49	48	38	63	49	47	35
38	46	49	51	49	42	67	59	52	39
40	48	52	53	51	43	67	59	52	39
44	51	54	58	52	44	70	62	54	41
47	53	60	60	57	47	72	64	56	45
54	55	65	64	59	50	78	66	59	45
60	60	68	68	65	53	81	68	63	49
63	63	70	72	68	54	87	74	66	50
65	66	71	75	70	57	93	78	68	55
66	67	76	77	71	60	94	80	71	62
67	70	78	80	75	64	95	83	73	67
68	74	82	81	76	64	96	85	76	69
73	76	83	85	80	69	100	86	76	73
77	81	89	87	82	70	106	89	77	76
78	86	91	88	84	74	109	92	78	80
80	88	93	92	87	78	111	95	82	82

M 1 – M 10: Release days for orders in Months 1 to 10

Table 3: Release date for ABM scheduled order 1-10

M 1	M 2	M 3	M 4	M 5	M 6	M 7	M 8	M 9	M 10
2	1	1	1	1	1	2	1	2	1
3	3	3	2	2	2	3	2	2	3
4	5	3	3	3	4	5	4	4	4
5	7	4	4	5	5	6	6	5	5
6	8	6	6	7	6	6	8	6	6
7	10	8	8	8	6	8	11	8	7
8	11	9	9	10	7	12	15	9	8
9	13	11	11	11	10	14	17	10	9
10	15	13	12	12	12	15	19	12	10
12	17	15	14	14	13	17	22	14	12
14	19	17	16	17	14	21	26	16	13
16	21	20	19	19	17	24	28	18	16
17	24	23	21	21	21	27	31	20	18
20	27	25	23	24	23	30	35	23	19
22	30	28	26	27	25	33	40	27	22
24	33	31	30	30	28	39	42	30	26
26	36	35	32	32	29	42	45	32	28
30	39	39	36	35	35	46	51	35	29
34	42	42	40	38	37	51	54	37	33
37	45	46	42	41	39	55	56	40	38
41	48	49	45	44	45	61	62	43	40
44	52	55	49	47	48	65	65	47	43
49	55	59	54	52	51	70	68	50	47
51	59	63	57	55	56	76	75	54	53
55	63	66	62	58	59	80	78	59	56
58	69	71	65	63	64	87	81	62	61
62	73	75	69	68	67	92	85	66	65
69	78	80	73	73	71	98	88	70	69
72	83	86	76	79	77	105	91	75	75
79	89	93	81	86	80	112	95	82	82

The graphs of ten different orders for ABM and that of FCFS are shown in Fig. 5 and 6. From the graph of Fig. 5, it is observed that the release dates for the ABM scheduled order is smaller, maintaining the lead time expected to release each order for the set of jobs. The flow of the graph shows the ABM scheduled order having smaller release date (i.e. less time taken to release the order) while the FCFS scheduled order had higher release date (i.e. higher time taken to release the same order as that of ABM scheduled order).

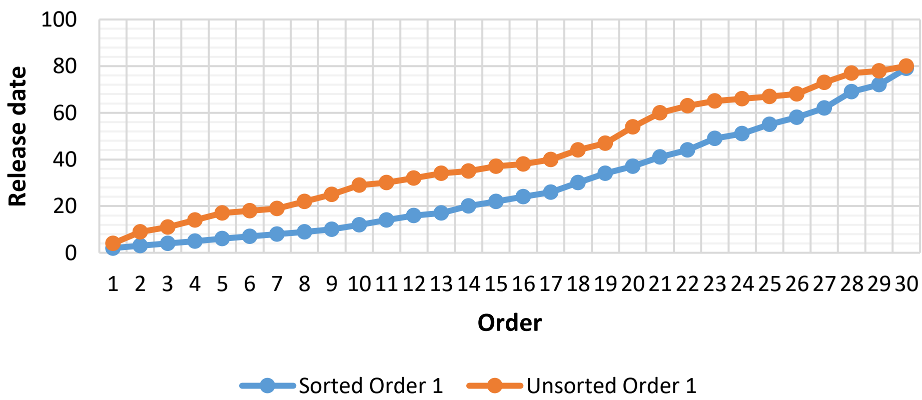

Fig. 5: Release date for the ABM scheduled job versus the FCFS scheduled job for Order 1

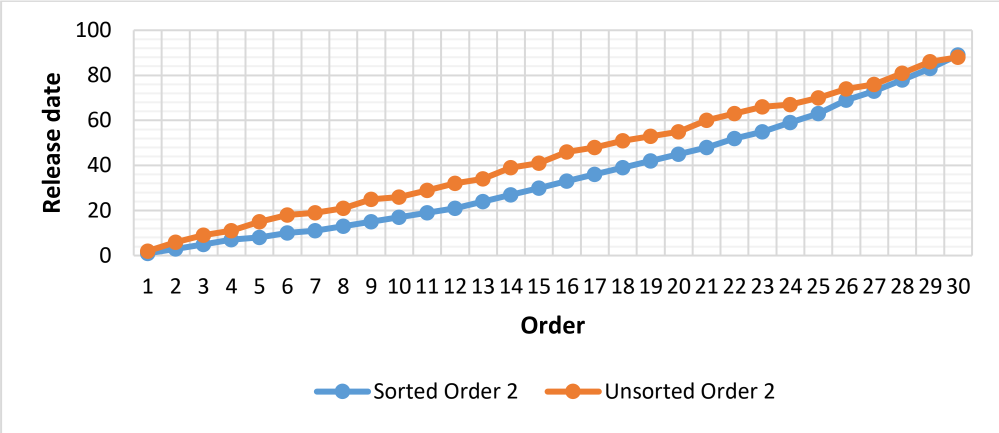

Fig. 6: Release date for the ABM scheduled job versus the FCFS scheduled job for Order 2

Figures 5 and 6 show the modelled agent-based job shop scheduler proposed in this research work with the conventional job scheduling process obtained from the case study companies. The release date for the ABM scheduled order shows that the agent-based model has a better result compared to the initial schedule process used by the companies in terms of customer satisfaction. Here, jobs are scheduled with respect to their type of finish and order size, which clears order queue. Smaller orders which their due date can be met in one day are processed first and released before large orders with acceptable large lead time.

4. Conclusions

The result shows that the modelled agent-based job shop scheduling is an improvement to the existing model and should be applied in a complex industrial set up for optimum machine usage and customer satisfaction. The simulation results were also used to determine the optimum scale of plant, given the rate of order arrival per month.

References

Blazewicz, J., Ecker, K.H., Pesch, E., Schmidt, G. and Weglarz, J. (1997) Scheduling Computer and Manufacturing Processes. Springer science and Business media, 48.

Hallier, M. and Hartmann, C. (2016) Constructing Markov State Models of Reduced Complexity from Agent-Based Simulation Data. Social Simulation Conference, 1-12.

Hameed, M.S.A. and Andreas, S. (2020) Reinforcement Learning on Job Shop Scheduling Problems Using Graph Networks. ArXiv Preprint, (ArXiv:2009.03836): 1–8.

Liu, Y., Lihui, W., Xi, V.W., Xun, X. and Lin, Z. (2019) Scheduling in Cloud Manufacturing: State-of-the-Art and Research Challenges. International Journal of Production Research, 57(15–16): 4854–79.

Nüske, F., Hao, W., Jan, H.P., Christoph, W., Cecilia, C. and Frank, N. (2017) Markov State Models from Short Non-Equilibrium Simulations - Analysis and Correction of Estimation Bias. Journal of Chemical Physics, 146(9): 94104.

Sarich, M., and Schütte, C. (2014) Utilizing Hitting Times for Finding Metastable Sets in Non-Reversible Markov Chains. ZIB Report, 14(32): 1–20.

Shoukat, A. and Seyed, M.M. (2020) Agent-Based Modelling: An Overview with Application to Disease Dynamics, (ArXiv:2007.04192): 2–18.

Sponer, J., Giovanni, B., Miroslav, K., Pavel, B., Sandro, B., Richard, A.C., Alejandro, G., Giovanni, P., Simon, P., Petr, J., Nils, G.W. and Michal, O. (2018a) RNA Structural Dynamics as Captured by Molecular Simulations: A Comprehensive Overview. Chemical Reviews, 118(8): 4177-4338.

Sponer, J., Giovanni, B., Miroslav, K., Pavel, B., Sandro, B., Richard, A.C., Alejandro, G., Giovanni, P., Simon, P., Petr, J., Nils, G.W. and Michal, O. (2018b) RNA Structural Dynamics as Captured by Molecular Simulations: A Comprehensive Overview. Chemical Reviews, 118(8): 4177-4338.

Ward, J.A. and Martín, L. (2019) Exact Analysis of Summary Statistics for Continuous-Time Discrete-State Markov Processes on Networks Using Graph-Automorphism Lumping. Applied Network Science, 4(1): 1–28.

Zhang, T., Shufang, X. and Oliver, R. (2017) Real-Time Job Shop Scheduling Based on Simulation and Markov Decision Processes. In Proceedings - IEE Winter Simulation Conference, 3899–3907.