Journal of Energy Technology and Environment Vol. 3(4) 2021 pp. 121-131 ISSN-2682-583x

NIPES

Journal of Energy Technology and Environment

Journal homepage: www.nipesjournals.org.ng

Modelling of an Agent Based-Job Shop Scheduling of Make Span Minimization in a Rigid Machine Setup

Chiagunye T. T. [a]****, Okey D. O.*** [b]***, Udeani H. U.*** [b]***, Ede C. C.*** [c]

[a,b,c] Department of Computer Engineering, Michael Okpara University of Agriculture, Umudike, Abia State, Nigeria

Corresponding Author: okeydanielogobuchi@gmail.com

Article Information

Article history:
Received 18 October 2021
Revised 21 November 2021
Accepted 30 November 2021
Available online 30 Dec. 2021

Keywords:
Makespan minimization, Rigid machine setup, Scheduler Agent, Bunching factor

https://doi.org/10.37933/nipes.e/3.4.2021.13

https://nipesjournals.org.ng

Abstract

This paper presents an agent-based model for scheduling job in a rigid machine setup. The model involved three sequential machines through which every job must pass followed by one out of three finishing machines used one per finishing type. For the type of product that is produced, the raw material must pass through the first three machines only in one order. Thus, the model developed took this sequential order into consideration. A well-crafted scheduler agent that 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 result of ten different monthly orders scheduled with bunching factor 2 had earliest release dates for eight out of the ten different orders and bunching factor 3 had earliest release dates for two orders while bunching factor 1 had none. The agent-based job shop scheduling model was validated with D.G. Kendall, classical method for poisson arbitrary distribution with nonpreemptive discipline where the agent-based model (ABM) compared favorably with the classical model. 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 makespan minimization in a rigid machine setup.

1. Introduction

This work explores the well-known n-by-m Job Scheduling Problem (**JSP**), in which n jobs must be processed exactly once on each of m machines. Each job i ($1 \le i \le n$) is routed through each of the m machines in a predefined order πi where $\pi i(j)$ denotes the jth machine ($1 \le ^{j} \le m$) in the routing order. The processing of job ¡ on machine $\pi i(^{j})$ is denoted O_{ij} and is called an operation. The scheduling objective is makespan minimization, which means to minimize the completion time of the last operation of any job [1] and [2]. Scheduling, understood to be an important tool for manufacturing and engineering, has a major impact on productivity of a process. 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 [3] and [4]. Existing deterministic shop floor schedulers work well for situation where n job must pass through m machine in any order while in the case study company, the n job must pass through the m machine in a given sequence which makes the job shop scheduling more complicated. The production process requires three machines in sequential order through which every raw material input must be processed and one-out-of three finishing machines 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 [4].

Yih and Thesen [5] formulated the scheduling problem as semi-Markov decision problems and used a non-intrusive knowledge acquisition method to reduce the size of the state space. The idea was to identify and update dynamically the states and transition probabilities that are used by an expert system to solve real time scheduling issue. However, the reduced state-space and the estimated parameters cannot fully represent the original problem but an approximation. It is possible for a state to appear which is not included in the reduced state space during the operation if the simulation process does not exhaust all the possible states which can result from user decisions.

The works of [5] and [6] 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 behavior by trial and error employing reinforcement learning algorithm. Gabel and Riedmille [7] 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.

Tao et al in [7] and [8] 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 is a sequential decision-making optimization technique. The system contains five (5) machines and produces two (2) products with two (2) operation flows. The operation flow in this model is not constrained to pass through each machine in series.

In this work, scheduler agent that uses a carefully crafted algorithm to schedule incoming orders for production has been developed. Scheduler agent carries out bunching of jobs either in 1 or 2 or 3 days per finishing type and selects the best out of the three approaches. Bunching is a scheduling technique adopted in this model to schedule an order in a queue. This technique allows a certain product type to be produced for 1 or 2 or 3…. n days before changing to another product type.

Thus, either a finishing type is done for only one day before changing to another order in sequence which typically is of another finishing type, or one finishing type is produced for 2 or 3 continuous days before changing over to another finishing type. The bunching is not fixed at 1 or 2 or 3 days for each finishing type but the best performing bunching type is selected for each set of orders being scheduled.

Hence a production scheduling and control that performs reactive (not deterministic) scheduling and can make decision on which job to process next based solely on its partial (not central) view of the plant becomes necessary. This requirement puts the problem in the class of agent-based model (ABM) where each job must be processed on three machines in series and the semi-processed product is passed on a one-of-three parallel finishing machine. Hence, this work adopts bunching technique to minimize the makespane.

2. Methodology

As part of the schedule Agent list of intentions, is the execution of the algorithm called Run Schedule Algorithm (RSA) intentions that satisfy all constraints. The production agent uses the projected distributions (worked out with Markov Chain) which is intention of the production agent to initiate the production process. The agent continues to run this process in the background while reacting to disturbances from the factory floor. For example, when a new order arrives, when a job or operation is preempted or a machine becomes unavailable, the agent updates the schedule and re-iterates the process.

Also, when backtracking is detected based on constraint checking, the agent adapts by either running another schedule from its schedule list or dumping the entire schedule and then re-computing the sequence. The objective of the algorithm is to schedule N jobs on M machines while taking stochastic conditions into effect so that the makespan MS is minimized.

The process scenario adopted for scheduling of jobs here uses three sequential machines followed by one out of three finishing machines. In this algorithm, jobs of varying sizes and levels are scheduled on the first three machines sequentially, then the output or the semi-processed product is passed on to any of the finishing (fourth) machine on a one out of three bases depending on the type of order. The algorithm allows for bunching of job in either three or two or one day, where the best bunch is selected except where the need for preferential consideration as in the case of government Job is highly needed, human/machine interaction will be invoked. Each complete run of the algorithm terminates once all jobs have been scheduled. A detailed description of the algorithm is given in section 2.1.

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 are:

- i. The orders gotten from the customers for thirty (30) days are grouped into three different finishing types.
- ii. Each finishing type is sorted in ascending order of job size with respect to the finishing type before scheduling.
- iii. Bunching of each finishing type of job with a bunching factor (Bf) of 1 or 2 or 3 was used to schedule the job.
- iv. Selecting the best bunching factor for each order, this means the bunching factor that gives the earliest finishing time for all the orders.
- v. Test running the carefully crafted algorithm on ten (10) separate orders.
- vi. Scheduling with the best bunching factor (Bf) for each of the ten different orders and that lead to the latest finishing dates at the bottom of the table.

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 is 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 of order 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. Software Sub-System Model

The flow chart showing the agent-based model for job shop scheduling and control is presented in Figures 1 and 2. It presents the order agent activity flow chart and the scheduler agent activity flow chart.

Figure 1 is the activity flow chart for the order agent. The Order Agent receives all incoming orders and on request passes the order records to the scheduler.

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

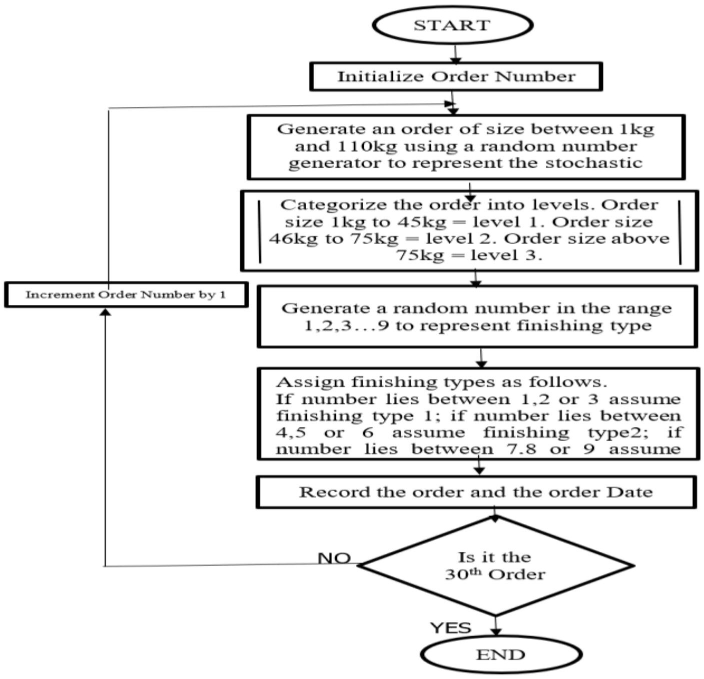

Figure 1: Order agent's activity flow chart

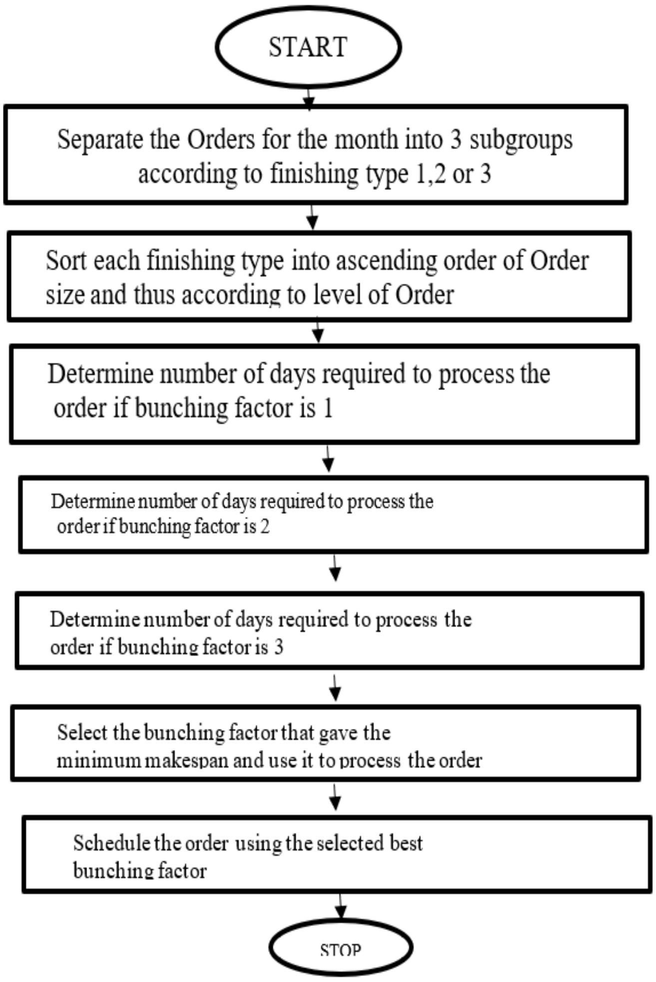

Figure 2: Scheduling agent's activity flow chart

3. Results and Discussion

3.1 Scheduling of Job Order Using Bunching Factors 1, 2 & 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 as shown in Table 1.

To present the concept in a more comprehensive way, 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 1 and 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. 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.

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. This is achieved in the reduction of days required to complete as set of orders using bunching techniques and which therefore reduces time and production cost. The graphs of Figure 3 and 4 were used to illustrate the performances of the three bunching factors.

Figure 3: Bar chart of Order 1 as Bf varies from 1-3

Figure 4: Bar chart of Order 2 as Bf varies from 1 – 3

The release dates for orders received in one month and then scheduled is shown in Figure 1. Referring to Table 1, the orders are scheduled using three different bunching factors (Bf1, Bf2 and Bf3) for the three finishing types. From the bar chart, 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 figure 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. Model Validation using D.G. Kendall queuing System

For the purpose of validation and testing of the agent-based job shop scheduling model, a classical method for poisson arbitrary distribution with nonpreemptive discipline by Kendall in [10] was used.

The mathematical model by D.G. Kendall is stated thus; $(\boldsymbol{M_i/G_i/1}) : (\boldsymbol{NPRP}/\propto/\propto)$**,** the symbol NPRP is used with the Kendall notation to represent the nonpreemptive discipline; M_i and G_i stand for poisson and arbitrary distributions [11].

Let $F_i(t)$ be the CDF of the arbitrary service time distribution for the ith queue (i=1, 2… M), and let $E_i\{t\}$ and $Var_i\{t\}$ be the mean and variance, respectively; let λ_i be the arrival rate at the ith queue per unit time. Define $Lq^{(k)}$, $Wq^{(k)}$, $Ws^{(k)}$ and $Ls^{(k}$ as;

Ls= expected number of customers in system
Lq= expected number of customers in queue
Ws = expected waiting time in system
Wq = expected waiting time in queue
Except that they now represent the measures of the kth queue.
The Model was evaluated using Equations 1 and 6

$$Wq^{(k)} = \frac{\sum_{i=1}^{n} \lambda_i (E_i^2\{t\} + Var_i\{t\}}{2(1-S_{k-1})(1-S_k)} \quad \ldots \quad 1\ [10]$$

$$Lq^{(k)} = \lambda_k Wq^{(k)} \quad \ldots \quad 2$$

$$Ws^{(k)} = Wq^{(k)} + E_k\{t\} \quad \ldots \quad 3$$

$$Ls^{(k)} = Lq^{(k)} + P_k \quad \ldots \quad 4$$

Where $P_k = \lambda_k E_k\{t\}$ … 5

$$S_k = \sum_{i=0}^{k} P_i < 1 \qquad K=1, 2\ldots M \quad \ldots \quad 6$$

$$S_0 \equiv 0$$

Where, $E_i^2\{t\}$= mean
$Var_i\{t\}$= variance
S = time interval
λ_k = constant service rate per a day
P_k = probability distribution.
Using sorted order approach, the values for the first order (first month) are given as follows and which was obtained from the case study of the job order comparison:

SL_1=	16	27	32	43	51	60	70	96	101				
SL_2 =	13	13	20	22	29	51	62	104					
SL_3 =	13	36	38	40	50	54	59	82	83	85	88	102	107

Where SL_1 is finishing type 1
SL_2 is finishing type 2
SL_3 is finishing type 3
Therefore, the mean for SL_1 is

$$\frac{16+27+32+43+51+60+70+96+101}{9}$$

$$\text{Mean} = \frac{496}{9} = 55.11$$

$$\lambda_1 = \frac{mean}{3} = \frac{55.11}{3} = 18.37$$

For SL_2

$$\text{Mean} = \frac{13+13+20+22+29+51+62+104}{8} = \frac{314}{8}$$

Mean = 39.25

$$\lambda_2 = \frac{39.25}{3} = 13.08$$

For SL_3

$$\text{Mean} = \frac{13+36+38+40+50+54+59+82+83+85+88+102+107}{13}$$

Mean = 64.38

$$\lambda_3 = \frac{64.38}{3} = 21.46$$

But $P_i = \lambda_i Ei\{t_i\}$

$$\therefore P_1 = \lambda_1 E\{t\} = 18.37 \left(\frac{1}{15}\right) = 1.2247$$

$P_2 = 13.08 \left(\frac{1}{19}\right) = 0.6884$

$P_3 = 21.46\ (\frac{1}{30}) = 0.7153$

Where 15kg, 19kg and 30kg are the maximum production capacity for product type 1, 2 and 3 per normal production day respectively.

$S_1 = P_1 = 1.2247$

$S_2 = P_1 + P_2 = 1.2247 + 0.6884 = 1.9131$

$S_3 = P_1 + P_2 + P_3 = 1.9131+0.7153 = 2.6284$

The due date for the complete schedule for order 1 is 2.6284 X 30 = 78.852

For the second order (order 2)

$SL_1 = \frac{22+23+23+25+34+39+45+46+51+56+71+75+83}{13}$

Mean = 45.61

$$\therefore \lambda_1 = \frac{45.61}{3} = 15.21$$

For SL_2

Mean = $\frac{30+30+35+38+55+60+63+78+101+104}{10}$

Mean = 59.4

$$\lambda_2 = \frac{59.4}{3} = 19.8$$

For SL_3

Mean = $\frac{47+62+72+79+87+98+109}{7} = \frac{554}{7}$

Mean = 79.14 $\therefore \lambda_3 = \frac{79.14}{3} = 26.38$

$P_1 = \lambda_1 E\{t_1\} = 15.21\ \left(\frac{1}{15}\right) = 1.014$

$P_2 = 19.8\left(\frac{1}{19}\right) = 1.042$

$P_3 = 26.38(\frac{1}{30}) = 0.879$

$S_1 = P_1 = 1.014$

$S_2 = P_1 + P_2 = 2.056$

$S_3 = P_1 + P_2 + P_3 = 2.052+0.879 = 2.935$

The Due date for the complete schedule for order 2 is 2.935 x 30 = 88.06

The complete value of the last release date for a queue of ten different orders using D.G. Kendall model is shown in Table 2, in comparison to that of agent-based job shop scheduling model.

Table 2: Comparison for the last release date for the proposed ABM and D. G. Kendal classical model

Order No	Release Date (Days) for Agent Model	Release Date (Days) for Classical Model
1	79	78.85
2	89	88.06
3	93	92.92
4	81	87.38
5	86	85.338
6	80	88.51
7	112	106.59
8	95	103.67
9	82	82.72
10	82	85.82

Table 2 presents the latest completion time for that of agent-based model and the classical model by D.G. Kendall. The result of the agent-based model shows a better result in comparison to that of classical model [4]. The graph of figure 5 shows the comparison of agent-based model to that of classical model.

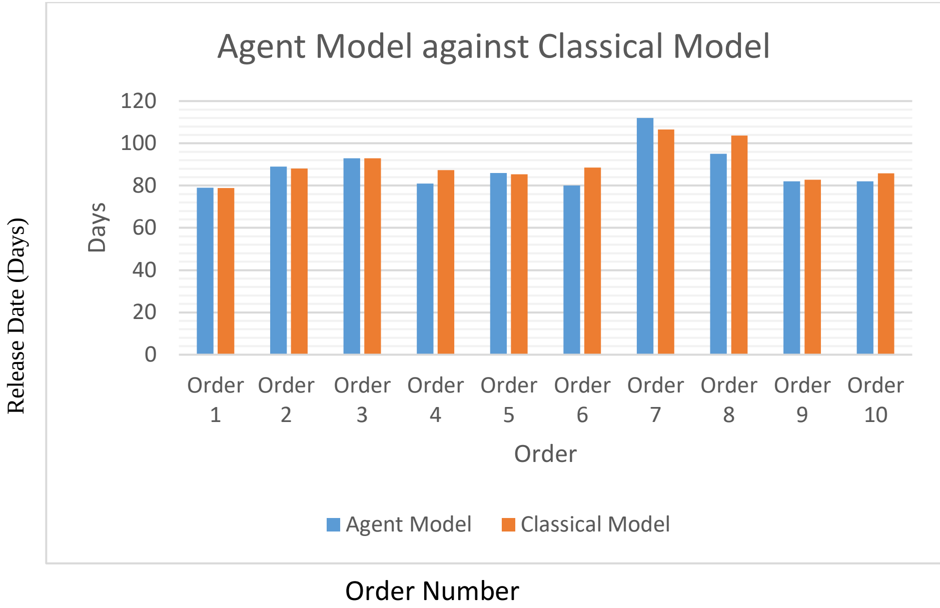

Figure 5: The comparison between Agent Model against Classical Model

Figure 5 shows clearly the performance of the ABM model, with the latest due date for the complete job out performing that of the classical model in orders 4,6,8,9 and 10 while it still relatively close to that of classical method in other once, as can be seen in order 1 that has the agent-based model result as 79 days while classical model had 78.85 approximately 79 days. The classical model seems better in order 7 with about 106 days against ABM's 112 days. This implies that the model developed achieves a better result up to 2.4% improvement which is makespan minimization (minimizing the time for the completion of a set of orders) as shown when evaluated with a classical model [4].

4. Conclusion

This developed model introduced an important technique (Bunching) that can choose, out of the several factors, the best factor that will give the minimum makaspan for scheduling a given set of orders. The developed model has a human/machine interaction that can adjust to the best schedule algorithm to take care of important jobs requiring preferential treatment. This made the model flexible and a better option for scheduling stochastic processes when compared to existing models which used classical model.

References

[1] S. V Sevastianov and G. J. Woeginger, "Makespan minimization in open shops: A polynomial time approximation scheme," *Math. Program.*, vol. 82, no. 1, pp. 191–198, 1998.

[2] M. Ghirardi and C. N. Potts, "Makespan minimization for scheduling unrelated parallel machines: A recovering beam search approach," *Eur. J. Oper. Res.*, vol. 165, no. 2, pp. 457–467, 2005.

[3] C. Byun *et al.*, "Node-based job scheduling for large scale simulations of short running jobs," in *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, 2021, pp. 1–7.

[4] E. W. Dijkstra, "Hierarchical ordering of sequential processes," in *The origin of concurrent programming*, Springer, 1971, pp. 198–227.

[5] Y. Yih and A. Thesen, "Semi-Markov decision models for real-time scheduling," *Int. J. Prod. Res.*, vol. 29, no. 11, pp. 2331–2346, 1991.

[6] T. Gabel, "Multi-Agent Reinforcement Learning Approaches for Distributed Job-Shop Scheduling Problems," *Univ. Osnabruck, Ger.*, p. 173, 2009, [Online]. Available: http://repositorium.uni-osnabrueck.de/bitstream/urn:nbn:de:gbv:700-2009081216/2/E-Diss925_thesis.pdf.

[7] T. Gabel and M. Riedmiller, "Adaptive reactive job-shop scheduling with reinforcement learning agents," *Int. J. Inf. Technol. Intell. Comput.*, vol. 24, no. 4, pp. 14–18, 2008.

[8] T. Zhang, S. Xie, and O. Rose, "Real-time batching in job shops based on simulation and reinforcement learning," in *2018 Winter Simulation Conference (WSC)*, 2018, pp. 3331–3339.

[9] T. Zhang, S. Xie, and O. Rose, "Real-time job shop scheduling based on simulation and Markov decision processes," in *Proceedings - Winter Simulation Conference*, 2017, pp. 3899–3907.

[10] D. G. Kendall, "Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain," *Matematika*, vol. 3, no. 6, pp. 97–112, 1959.

[11] H. A. Taha, *Operations research: an introduction*, vol. 790. Pearson/Prentice Hall Upper Saddle River, NJ, USA, 2011.