

Heuristic Algorithm for Job Shop Scheduling Problem to Minimize Makespan Considering Bottleneck Machine

Mark Jordan Songgigilan¹, Rahmi Maulidya^{1*}, Agung Sasongko¹

¹Production System Laboratory, Industrial Engineering Department, Industrial Engineering Faculty, Universitas Trisakti, JL. Kyai Tapa no 1 Jakarta, 11440, Indonesia

Corresponding Author: rahmimaulidya@trisakti.ac.id
marksonggigilan@gmail.com; agung.sasongko@trisakti.ac.id

Abstract. In the job shop scheduling problem, orders are processed in a unidirectional sequence, some machines have a queue, and a job may be processed in a machine several times. Orders are accepted for a certain period and the due date is set at once to please the customer without considering the resource. The problem that the completion time cannot meet the due date and the orders are not completed as promised can cause industries to lose the orders or get a penalty. The objective of this research is to formulate a heuristic algorithm for job shop scheduling problems to minimize the makespan considering bottleneck machines. The illustration in this research is based on a case in the stamping parts industry. The first step in the algorithm is to check the available time for processing the jobs. Then, the machine with the maximum processing time is defined as the bottleneck machine. The step for finding the solution is based on an algorithm for bottleneck scheduling. In this research, the result is compared using the solution of a non-delay algorithm and LEKIN software for job shop scheduling. The algorithm shows the selected orders processed in a production period and a schedule with a minimum makespan. The benefit of considering the bottleneck machine is still giving the best solution than the solution of the non-delay algorithm in this illustration.

Keywords: stamping parts, job shop, machine capacity, bottleneck

29. Introduction

In the job shop environment, manufacturers must handle customer orders with all their resources. Every order has a unique and challenging processing route to handle. Receiving orders with different processing routes can cause the utility of machines to differ [1]. Though this situation is challenging, manufacturers sometimes receive all orders at once. Problems may occur after they promised the customer and set the due date without checking first the available capacity of their resources. In this paper, the gap is accommodated to set how many orders can be handled in the planning horizon.

Once the orders are set in a particular planning horizon, then the bottleneck machines are beneficial to be identified beforehand [2] [3]. The workload distribution on each machine will vary due to the unidirectional process of jobs. Some machines may be skipped, and some machines may appear more than once in processing a job [4]. The available capacity will become the constraint for achieving the production target. Some manufacturers can apply the capacity cushion to save production with the amount of reserve capacity and the cushion will vary for some manufacturers [5]. Hence, the load of a machine still cannot exceed the available capacity and can only be controlled when receiving orders.

In terms of the theory of constraints, the production system circulation is controlled by the throughput rate resulting in bottleneck machines [6]. So, the bottleneck machine with the most workload will be scheduled first then followed by the most related machine with the bottleneck machine [7]. In contrast with the job shop scheduling problem, the schedule is generated by following the routing process and the available machine for processing the job [1]. The methods are obtained from the solution of heuristic algorithm such as the non-delay algorithm [8] [9] [10], and the active algorithm

[10]. The objective function of this method is to minimize the makespan. In this paper, the job shop scheduling problem is compared to the solution of bottleneck scheduling with the objective function of minimizing the makespan.

This paper deals with a job shop environment and formulates a heuristic algorithm to solve the job shop scheduling problem considering the bottleneck machines with the objective function of minimizing the makespan. The illustration in this paper comes from the case of the stamping industry where the orders are customized spare parts for automotive companies.

Several assumptions are made as follows:

- Machines are available in good condition
- Travel time is neglected
- Pre-emption job is not allowed.
- Jobs are available for processing at a stage immediately after processing completion at the previous stage.
- Each machine can process at most one job at a time
- Each job can be processed on at most one machine at a time

This paper is organized as section 2 for methods, section 3 for the proposed algorithm, section 4 for the result and discussion, and section 5 for the conclusion.

30. Methods

The bottleneck machine is defined using the theory of constraint approach to finding the maximum capacity that can be handled by the system [6] and the step for scheduling the bottleneck machine is adopted from [7].

Notation

- n = number of jobs
 m = number of machines
 S = the scheduled jobs
 U = the unscheduled jobs
 t = the current time
 b = the scheduled machine
 r_i = the release time for job i
 r_i^b = the arrival time for job i in machine b
 d_i = due date for job i in machine b
 d_i^b = the completion time for job i in machine b
 p_i = the processing time for job i in machine b
 $j(b)$ = operation of job i in machine b

The proposed algorithm are as follows.

- Step 0. Initialize the problem.
Step 1. Set the available capacity of each machine for the planning horizon.
Step 2. Store the time needed in each machine for the received order.
Step 3. Is there any available capacity left?
 If Yes, add more orders and repeat Step 2.
 If No, Go to Step 4.
Step 4. Define a machine with the maximum capacity.
Step 5. Set as machine b .
Step 6. For machine b , set $U = \{1, 2, \dots, n\}$; $p_i = p_{ij(b)}$; $i = 1, 2, \dots, n$; and $t = \min_{i \in U} r_i^b$.
Step 7. Define the available jobs in $S = \{i | r_i^b \leq t, i \in U\}$. Schedule job i^* on machine b , select job i^* with the best priority among S .
Step 8. Set $U \leftarrow U - \{i^*\}$.

- Step 9. Is $U = \emptyset$?
 If Yes, then all jobs have been scheduled. Go to Step 10.
 If No, set $t = \max(\min_{i \in U} r_i^b, t + p_{i^*}')$. Go to Step 4.
- Step 10. Is there any machines to schedule?
 If Yes, define the most related machine to the bottleneck machine. Repeat to Step 5.
 If No, Go to Step 11
- Step 11. Output the schedule of each machine and define the makespan.

31. Result and Discussion

The illustration is developed for the job shop environment of a stamping company. The system has nine machines each of which performs a processing stage and produces components for automotive industries. Figure 1 shows the system represented that received the 15 jobs for a planning period of August 2022. Table 1 shows the routing process and the demand for each job. Table 2 shows the time required for each job.

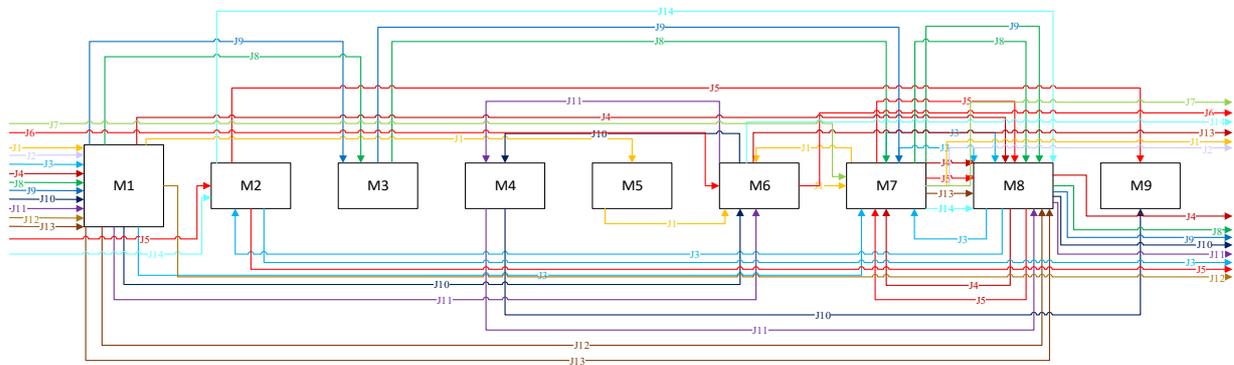


Figure 4. The job shop environment system

Table 1. Routing Process for the planning period

Job	Operation 1	Operation 2	Operation 3	Operation 4	Operation 5	Operation 6	DEMAND (unit)
J1	BLANK (M1)	PIERCING (M5)	RISTRIKE (M6)	BENDING (M7)			37000
J2	BLANK (M1)	PIERCING (M5)	RISTRIKE (M6)	BENDING (M7)			37000
J3	BLANK (M1)	EMBOSS (M7)	FLANGE (M7)	FLANGE 2 (M8)	SEPARATING (M7)	EMBOSS (M8)	17500
J4	BLANK (M1)	BENDING (M8)					9200
J5	BLANK (M2)	BENDING (M9)	RISTRIKE (M8)	FORMING (M7)	CAM PIERCING 1 (M8)		41000
J6	BENDING (M6)						15000
J7	BLANK (M1)	EMBOSS (M7)					15000
J8	BLANK (M1)	FORMING (M3)	RISTRIKE (M7)	PIERCING (M8)			5500
J9	BLANK (M1)	FORMING (M3)	RISTRIKE (M7)	PIERCING (M8)			8500
J10	BLANK (M1)	CHAMFER (M6)	BENDING (M4)	PIERCING (M8)			5500
J11	BLANK (M1)	CHAMFER (M6)	BENDING (M4)	PIERCING (M8)			8500
J12	BLANK (M1)						40000
J13	BLANK (M1)	FORMING (M8)	FLANGE (M7)	CAM PIERCING 1 (M8)	EMBOSS (M7)	PIERCING 2 (M6)	10500
J14	BLANK (M2)	FORMING (M8)	FLANGE (M7)	EMBOSS (M8)	CAM PIERCING 1 (M7)	PIERCING 2 (M6)	10500
J15	BLANK (M2)	FORMING (M8)	TRIMMING (M9)	RISTRIKE (M7)	PIERCING 1 (M8)	PIERCING 2 (M6)	7800

Table 2. The Process Time of Each Job Needed for the Requested Demand

Job	M1	M2	M3	M4	M5	M6	M7	M8	M9	Total
J1					122.56	122.56	148			393.12
J2	56.89				122.56	122.56	148			393.12
J3	54.85						128.33	108.38		236.71

J4	20.5						30.63		30.63
J5		75.65				163.97	327.94	163.97	731.53
J6					27.3				27.3
J7							27.3		27.3
J8	8.44		18.33				18.33	18.33	54.99
J9	13.04		28.33				28.33	28.33	84.99
J10	8.44			18.33		18.33		18.33	54.99
J11	13.04			28.33		28.33		28.33	84.99
J12	61.53								61.53
J13	16.14					35.04	70.08	70.08	175.2
J14		16.14				35.04	70.08	70.08	191.34
J15		12.01				26	26	52	147.21

The problem is initialized as the job shop environment with the routing process and the processing time to meet the demand. In the algorithm Step 1, the available capacity is defined for each machine. Then in Step 2, the available capacity will check the processing time of each job store in Table 2. The process is stopped when the available capacity cannot handle more jobs. So, the maximum number of jobs that can be processed is only 15 jobs. The total processing time for the jobs compared to the available capacity can be seen in Table 3.

Table 3. The Processing Time for the planning period

Time	M1	M2	M3	M4	M5	M6	M7	M8	M9
Processing Time	252.87	103.8	46.66	46.66	245.12	415.16	828.42	752.43	195.17
Available capacity	648	648	648	648	648	648	648	648	648
Capacity Cushion							180.42	104.43	
	Feasible								

Table 3 shows M7 and M8 are the machines with the total processing time. Though M8 has the most job to be processed, the total processing time shows that M7 with the most time is set as the bottleneck machine *b*. Figure 2 shows the arrangement of M7 as the bottleneck machine and the job routing.

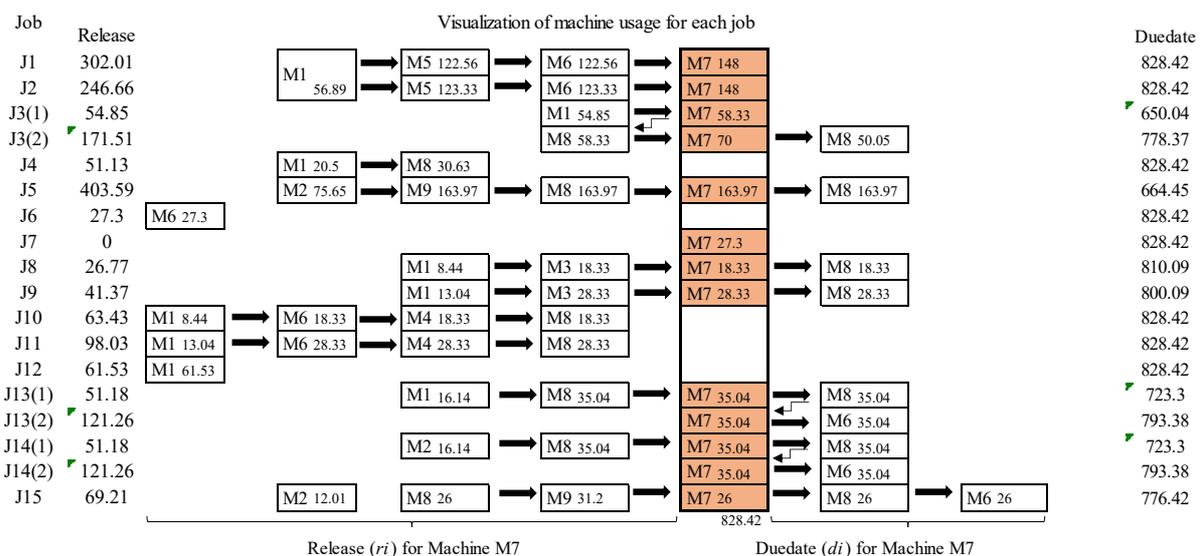


Figure 5. Step 6 for machine *b* (M7)

An arrangement is set for the re-entrant job J3 to become the separate job of J3(1) and J3(2). Figure 2 shows that J3 has a route of M1-M7-M8-M7-M8. In this case, the arrangement is set for J3(1) processed in M1-M7 and continues with J3(2) for processing in M8-M7-M8. The same arrangement is also set for J13 and J14. In Figure 2, the release time and due date for machine *b* are defined by using Equation (1) and Equation (2) which are adopted from [7].

$$r_i^b = r_i + \sum_{l=1}^{j(b)-1} p_{il} \quad (1)$$

$$d_i^b = d_i - \sum_{l=j(b)+1}^m p_{il} \quad (2)$$

Figure 4 shows the release time and due date obtained using Equation (1) and Equation (2) for machine M7. The next step is to schedule the job and the selection is made from closet release time to current time t . The schedule is obtained by selecting the job as can be seen in Table 5.

Table 4. The Release, Due date and Processing Time for Machine M7

Job	J1	J2	J3(1)	J3(2)	J4	J5	J6	J7	J8
r_i	302.01	246.66	54.85	171.51	51.13	403.59	27.3	0	26.77
d_i	828.42	828.42	650.04	778.37	828.42	664.45	828.42	828.42	810.09
p_i	148	148	58.33	70	0	163.97	0	27.3	18.33

Job	J9	J10	J11	J12	J13(1)	J13(2)	J14(1)	J14(2)	J15
r_i	41.37	63.43	98.03	61.53	51.18	121.26	51.18	121.26	69.21
d_i	800.09	828.42	828.42	828.42	723.3	793.38	723.3	793.38	776.42
p_i	28.33	0	0	0	35.04	35.04	35.04	35.04	26

The schedule for M7 is represented by the [S, C] stands for Starting Time and Completion Time, then the Gantt Chart can be generated for M7. The next machine to be scheduled is selected from Figure 2. Find the most machines that have the closest relation to M7, and the next step is to set M8 as machine b . Table 5 shows the release time, the due date, and the processing time for each machine when the machine is set as machine b . When scheduling the next machines, the previous machine's Gantt chart will control the next job in the system.

Table 5. The Release, Due date and Processing Time for Machine b

b	Job	J1	J2	J3(1)	J3(2)	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13(1)	J13(2)	J14(1)	J14(2)	J15
M7	r_i	302.01	246.66	54.85	171.51	51.13	403.59	27.3	0	26.77	41.37	63.43	98.03	61.53	51.18	121.26	51.18	121.26	69.21
	d_i	828.42	828.42	650.04	778.37	828.42	664.45	828.42	828.42	810.09	800.09	828.42	828.42	828.42	723.3	793.38	723.3	793.38	776.42
	p_i	148	148	58.33	70	0	163.97	0	27.3	18.33	28.33	0	0	0	35.04	35.04	35.04	35.04	26
M8	r_i	0	0	132.29	272.37	20.5	163.97	0	0	45.63	73.96	18.33	28.33	0	35.04	167.33	35.04	767.38	12.01
	d_i	828.42	828.42	708.37	828.42	828.42	500.48	828.42	828.42	828.42	828.42	828.42	828.42	828.42	688.26	758.34	688.26	758.34	719.22
	p_i	0	0	58.33	50.05	30.63	163.97	0	0	18.33	28.33	18.33	28.33	0	35.04	35.04	35.04	35.04	26
M6	r_i	122.56	123.33	0	0	0	0	0	0	0	0	8.44	13.04	0	0	35.04	0	35.04	26
	d_i	420.37	272.37	828.42	828.42	828.42	828.42	0	828.42	828.42	828.42	0	0	828.42	828.42	828.42	828.42	828.42	828.42
	p_i	122.56	123.33	0	0	0	0	27.3	0	0	0	18.33	28.33	0	0	35.04	0	35.04	26
M3	r_i	0	0	0	0	0	0	0	0	8.44	13.04	0	0	0	0	0	0	0	0
	d_i	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	27.33	73.96	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42
	p_i	0	0	0	0	0	0	0	0	18.33	28.33	0	0	0	0	0	0	0	0
M1	r_i	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	d_i	420.37	272.37	73.96	73.96	828.42	828.42	828.42	828.42	0	27.33	828.42	0	0	132.29	132.29	828.42	828.42	828.42
	p_i	56.89	56.89	54.85	54.85	20.5	0	0	0	8.44	13.04	8.44	13.04	61.53	16.14	16.14	0	0	0
M9	r_i	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	d_i	828.42	828.42	828.42	828.42	828.42	568.37	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	802.42
	p_i	0	0	0	0	0	163.97	0	0	0	0	0	0	0	0	0	0	0	31.2
M2	r_i	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	d_i	828.42	828.42	828.42	828.42	828.42	568.37	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	732.34	767.38
	p_i	0	0	0	0	0	75.65	0	0	0	0	0	0	0	0	0	16.14	16.14	12.01
M4	r_i	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	d_i	828.42	828.42	828.42	828.42	828.42	568.37	828.42	828.42	828.42	828.42	0	0	828.42	828.42	828.42	828.42	828.42	828.42
	p_i	0	0	0	0	0	0	0	0	0	0	18.33	28.33	0	0	0	0	0	0
M5	r_i	56.89	56.89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	d_i	420.37	272.37	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42	828.42
	p_i	122.56	123.33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The algorithm will stop after all processing route is scheduled and the Gantt Chart can be generated to show the schedule resulting from the algorithm. The Gantt Chart is shown in Figure 3 for the proposed algorithm.

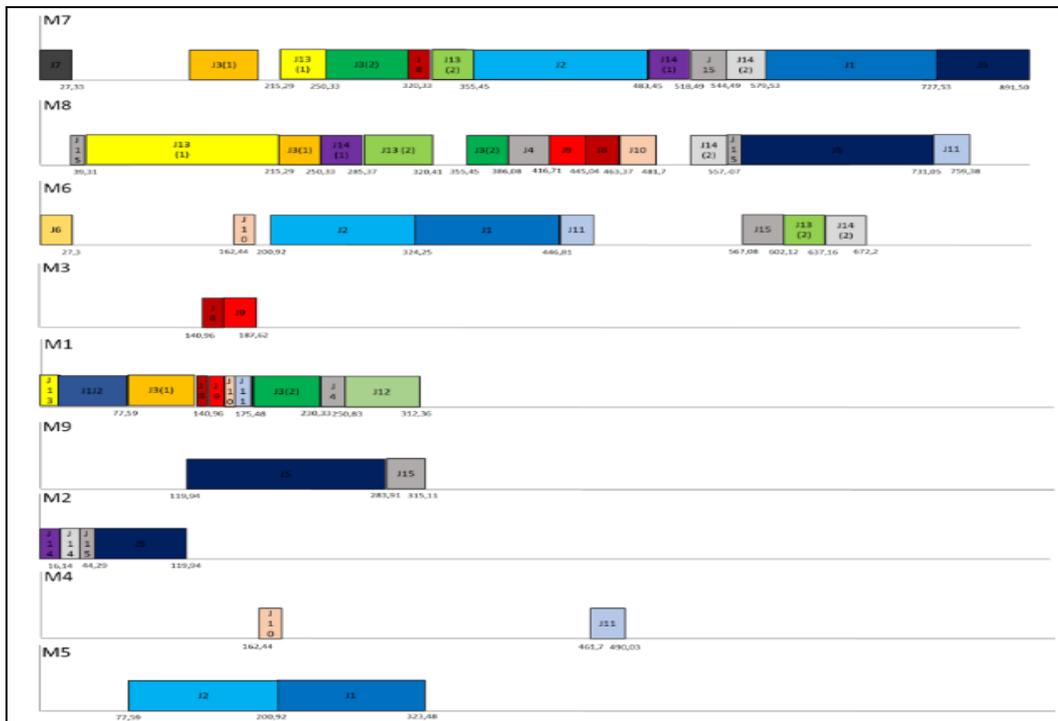


Figure 6. Gantt Chart for the proposed algorithm

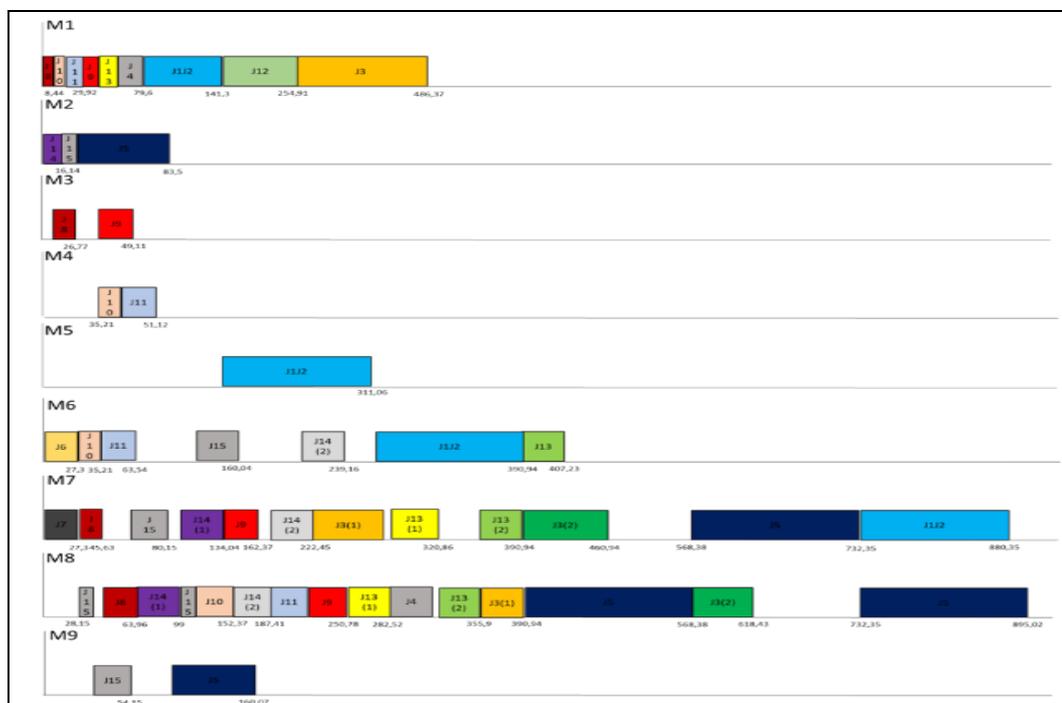


Figure 7. Gantt Chart for non-delay algorithm

As compared to the proposed algorithm, the non-delay algorithm is conducted to solve the solution of the job shop environment [1]. The Gantt chart for the non-delay algorithm can be seen in Figure 4. In this paper, we also compare the solution with the result from the software LEKIN 3.3 for job shop

scheduling problems (source: <https://lekin.software.informer.com/download/#downloading>). Figure 5 shows the Gantt chart for LEKIN 3.3 software.

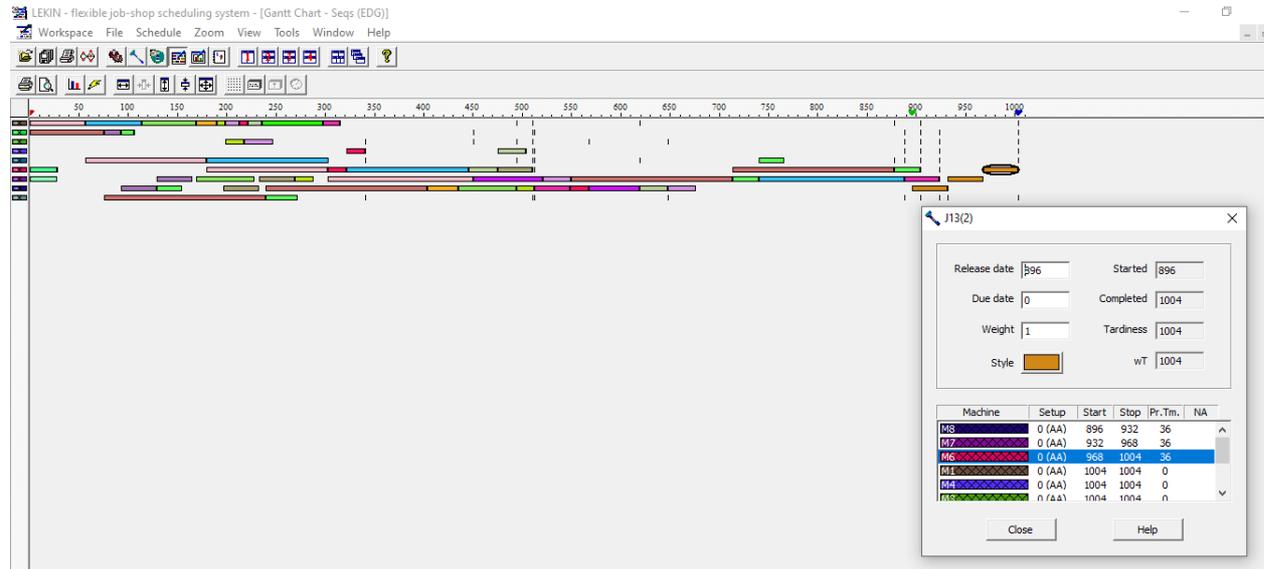


Figure 8. Gantt Chart for Software LEKIN 3.3

And the performance of all methods in this paper can be seen in Table 6. In Table 6, the result from the current condition is also presented in this paper. The non-delay algorithm gives a solution of 1.48% better than the current condition while the proposed algorithm that considers the bottleneck machine gives a solution of 1.82% better than the current condition.

Table 6. The Algorithm Performance

Methods	Makespan	Performance
Current Condition	908.05	-
No-Delay Alg.	895.02	1.48%
Proposed Alg.	891.5	1.82%
LEKIN 3.3	1004	-

The LEKIN 3.3 software for job shop scheduling problems cannot give a better solution than the non-delay algorithm. We suspect that the re-entrant job J3, job J13, and job J14 may affect the input data. This should be investigated more and become future research to be continued.

32. Conclusions

This paper shows the algorithm for receiving orders based on the available capacity of machines in the job shop environment and scheduling the job based on the bottleneck machine. The result is compared with the schedule of the non-delay algorithm and the schedule from LEKIN 3.3. This proposed algorithm is limited to minimizing makespan, therefore the result shows the better makespan but still leaves the tardy job. Another limitation is that the available capacity is set due to the available resources, and the proposed algorithm arranges the maximum number of orders that can be scheduled in the system. So, in the future, we can continue the research with the model formulation with the bi-objective function of makespan and the number of tardy job minimization.

33. References

- [1] K. R. Baker and D. Trietsch, Principles of Sequencing and Scheduling, New Jersey: John Wiley & Sons, Inc., 2009.

- [2] J.-Q. Wang, J. Chen, Y. Zhang and G. Q. Huang, "Schedule-based Execution Bottleneck Identification in a Job Shop," *Computer & Industrial Engineering*, vol. 98, pp. 308-322, 2016.
- [3] R. Zhang and C. Wu, "Bottleneck Machine Identification Method Based on Constraint Transformation for Job Shop Scheduling with Genetic Algorithm," *Information Sciences*, vol. 188, pp. 236-252, 2012.
- [4] I. G. Drobouchevitch and V. A. Strusevich, "Heuristics for The Two-stage Job Shop Scheduling Problem with a Bottleneck Machine," *European Journal of Operational Research*, vol. 123, no. 2, pp. 229-240, 2000.
- [5] L. J. Krajewski, M. K. Malhotra and L. P. Ritzman, *Operations Management: Processes and Supply Chain*, Boston: Pearson Education, 2016.
- [6] Q. Lei and T. Li, "Identification Approach for Bottleneck Clusters in a Job Shop Based on Theory of Constraints and Sensitivity Analysis," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, pp. 1-11, 2015.
- [7] D. Sipper and R. L. Bulfin, *Production Planning, Control and Integration*, New York: McGraw-Hill, 1997.
- [8] S. Harto, A. K. Garside and D. M. Utama, "Penjadwalan Produksi Menggunakan Algoritma Jadwal Non Delay Untuk Meminimalkan Makespan Studi Kasus Di CV. Bima Mebel," *Spektrum Industri*, vol. 14, no. 1, p. 79, 2016.
- [9] T. Al Barokah, E. Zaini and A. Saleh, "Usulan Penjadwalan Produk Menggunakan Algoritma Non Delay Dengan Mesin Paralel Pada PT. Adhichandra Dwiutama," *Reka Integra, Jurnal Online Teknik Industri Itenas, Bandung*, vol. 4, no. 2, pp. 102-113, 2016.
- [10] P. Fithri and F. Ramawinta, "Penjadwalan Mesin dengan Menggunakan Algoritma Pembangkitan Jadwal Aktif dan Algoritma Penjadwalan Non-Delay untuk Produk Hydrotiller dan Hammermil pada CV. Cherry Sarana Agro," *Jurnal Optimasi Sistem Industri*, vol. 12, no. 2, p. 377, 2016.