Design of Job Scheduling Using BAT Algorithm To Minimize Makespan in Hybrid Flowshop

Sumiharni Batubara, Debbie Kemala Sari, Dhimas Aryo Wicaksono
Industrial Engineering Department
Faculty of Industrial Technology
Trisakti University
Kampus A USAKTI, INDONESIA
sumiharni@trisakti.ac.id; debbie.kemala@trisakti.ac.id; dhmsaryo@gmail.com

Abstract

This research was conducted in a manufacturing company that produces high carbon wire. The company had issues in fulfilling demand where it had to work on overtime despite having a production capacity that is greater than the number of requests. This problem are due to greater actual production time than the standard production time and job scheduling prioritization using first come first serve. The purpose of this study is to create a job schedule using a bat algorithm in hybrid flow shop for high carbon wire products to minimize makespan and hence, to reduce overtime. In this study, a comparison was made between 4 conditions, namely one condition using time and sequence of work based on production reports (condition 1), second condition using actual time and sequence of work based on bat algorithm (condition 2), third condition using standard processing time and sequence of work based on production reports (condition 3), and forth condition using standard time and sequence of work based on bat algorithm (condition 4). Post data processing, we found that the condition 1 has a makespan of 551 hours and requires 82 hours of overtime; condition 2 has a makespan is 545 hours and requires 76 hours overtime; condition 3 has a makespan is 478 hours and requires 9 hours overtime; and condition 4 has a makespan is 439 hours and does not require any overtime 4. By using bat algorithm and standard processing time, the company can eliminate overtime and reduce makespan by 112 hours from condition 1.

Keywords
Scheduling, Hybrid flowshop, Bat algorithm

1. Introduction

1.1. Background

A steel manufacturing industry typically produces four types of products; high carbon wire, low carbon wire, cold heading wire and alloy steel wire. This research will observe high wire carbon production only. High carbon wire has two types of diameter, 2.24mm and 2.4 mm. High carbon wires will go through two processes. The first process is called pickling and coating and the second is the drawing process. The company has one pickling machine for the first process and three parallel and identical machines specifically allocated for drawing high carbon wire products. The scheduling method applied for high carbon wire is first come first serve (FCFS). FCFS is a scheduling method that schedules production based on the time that the orders were received.

One of the problems faced by the manufacturing industry is continuously having to do overtime to fulfil orders albeit having a production capacity that is higher than demand. Through preliminary studies at the company, we understand that the problem is caused by 2 factors. The first was the human factor where the operator decreased the speed of the machine and increasing the actual processing time to be higher than the standard processing time. The second factor is the ineffective scheduling method of FCFS that caused a high makespan. Based on this, we continue our research to analyze the high carbon wire production process at a specific observation period and apply production scheduling method using bat algorithm to reduce makespan and minimize overtime.

1.2. Research Question

Through observations and interviews, it is understood that high makespan affecting high overtime in fulfilling orders. Currently, the company uses first come first serve method to process orders and current actual processing time is higher than standard processing time. These have caused late order fulfilment due to high makespan and also production overtime.

1.3. Purpose of the Study

To compose a high carbon wire production scheduling in hybrid flow shop to obtain a job schedule that could minimize makespan and reduce overtime.

1.4. Research Limitations

This research is limited to the following:

1. Scheduling is only done for high carbon wire.
2. Data used for this study is from the observation period of September 2018.
3. Drawing machine used is continuous 5, 6 and 8.
4. Changeover time between machines is ignored.
5. Each job that has completed pickling and coating process is immediately transferred to drawing process.
6. An interview with the production manager was conducted to identify the cause of the problem.

2. Review of Related Literatures

2.1. Scheduling

Scheduling is used to allocate human resources and equipment to perform specific tasks in a specific period of time. Resources can be labeled as machines and tasks as jobs. [1]. Four objectives of scheduling are reducing waiting time, increasing productivity of available resources, reducing work delays and assisting in production capacity planning.

Based on the number of machines and its relationship with each other, scheduling is categorized into two which are single machine scheduling and multi-machine scheduling. Multi-machine scheduling is divided into three which are scheduling n jobs at m parallel machines, scheduling n job at m serial machines (flow shop), and scheduling n jobs at m job shop machines. The difference between flow shop and job shop is in the sequence of the production process. Flow shop has a consecutive sequence of production while a job shop does not.

Some terminologies used in scheduling are the following,[1]:

- Processing Time is the time required to finished a job.
- Due date is maximum date that is agreed by the customer to finish an order.
- Completion time is the time span to complete a job from the beginning (start) to finish.
- Lateness is a deviation between completion time and due date.
- Tardiness is a positive measure of lateness.
- Earliness is when job is complete prior to due date. It is also called negative lateness.
- Makespan is the total time required to process the whole operations of a scheduled component from the beginning process to the end process.
- Gantt chart is a visual representation of the sequence of a job and describe the situation when a job processing is started until the end.

2.2. Hybrid Flow Shop Scheduling

Hybrid flow shop scheduling is an expansion of a classis flow shop scheduling where there is only one machine for each production line, however in hybrid flow shop there are different number of machines in each production line. For example, line A only has one machine but line B has three identical machines, uniform and unrelated [3]. The following charts describe the difference between classic flow shop scheduling and hybrid flow show scheduling.

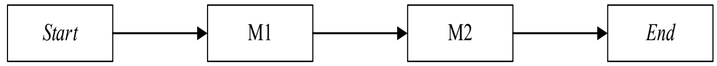


Chart 1. Classic *Flow Shop*

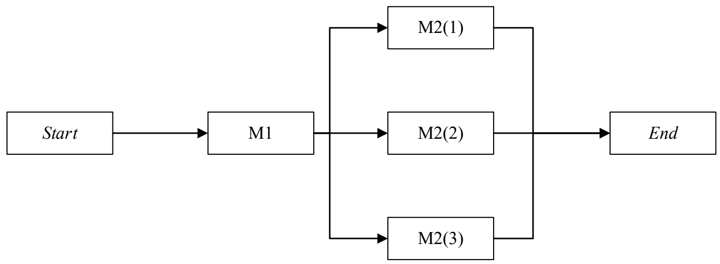


Chart 2. *Hybrid Flow Shop*

2.3. First Come First Serve (FCFS)

First come first serve or FCFS is a simple scheduling method where the job processing order follows the arrival of the order [4]. In this method, the first order received by the company will be processed first. FCFS scheduling method is likely to be used by companies because it is considered easier and more just.

2.4. Bat Algorithm

Bat Algorithm is an algorithm to optimize a function that is inspired by a ecolocation behavior of bats. Bat algorithm was first discovered by Yang in 2010. The ideal or approximate condition that is used by Yang to develop bat algorithm was the following:

- All bats used ecolocation to sense the distance with their food.
- Bats fly randomly with v_i velocity in position x_i with f_{mint} frequency, with a noise variance of A_0 to look for prey. They can automatically adjust the frequency from the pulse emitted and adjust the emission signal rate $r \in [0, 1]$, depending on the proximity of their target.
- Although noise varied, it is assumed that the variance range from a large (positive) A_0 to a constant minimum A_{min}.

All algorithm has an objective function. In this research, the objective function of the bat algorithm is to minimize makespan using the following equation [5]:

$$C_{max} \geq C_{js} \quad (1)$$

$$C_{js} = S_{js} + P_{sj} \quad (2)$$

Where:

C_{max} = *makespan*
C_{js} = *completion time* from job j in proses s
S_{js} = the start time of job j in proses s
P_{sj} = processing time of process s from job j

In bat algorithm, there are two important things to define, which are the bat position (x_i) and the bat velocity (v_i). The position and velocity of bat is an initial solution is randomly generated from zero to one for as many times as the number of bats, number of jobs or assignment and the number of processes. To optimize the objective function, each bat will collocate with different speed or velocity to obtain an optimal solution. The following is the equation of location change x_i^t with velocity of movement v_i^t to time (t)[3]:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (3)$$

$$v_i^{t+1} = v_i^t + (x_i^t - x_*)f_i \quad (4)$$

$$x_i^{t+1} = v_i^{t+1} + x_i^t \quad (5)$$

Where:

β = random number between 0 to 1

x_* = best temporary position from a bat from the previous iteration
x_i^{t+1} = bat position during an iteration
x_i^t = bat position at the previous iteration
v_i^{t+1} = bat velocity during iteration
v_i^t = bat velocity at the previous iteration
f_i = frequency of bat i
f_{min} = minimum bat frequency
f_{max} = maximum bat frequency

Meanwhile, the equation of movement to look for an optimum local solution is the following [5]:

$$x_{new} = x_* + \varepsilon\overline{A^t} \quad (6)$$

Where:
x_{new} = bat location from a local solution search
x_* = best temporary location of a bat
ε = random number between 0 to 1
$\overline{A^t}$ = mean of all bat noise rates

In addition to position and speed of bats, noise and pulse emission are important to define. Noise (A_t) and pulse emission (r_i) of a bat will always change in each iteration. This is caused by the fact that the noise will decrease as bats become closer to it's prey. While pulse emission will also change with the bat proximity with it's prey. The following is the equation of noise and pulse emission of a bat algorithm[5]:

$$A_i^{t+1} = \propto A_i^t \quad (7)$$
$$r_i^t = r_i^0[1 - \exp(-\gamma t)] \quad (8)$$

Where:
A_i^{t+1} = new noise
A_i^t = noise during an iteration
r_i^t = new pulse emission
r_i^0 = initial *pulse emission*

Similar to simulated annealing, α in bat algorithm show the decreasing factor of a bat's noise rate [6].

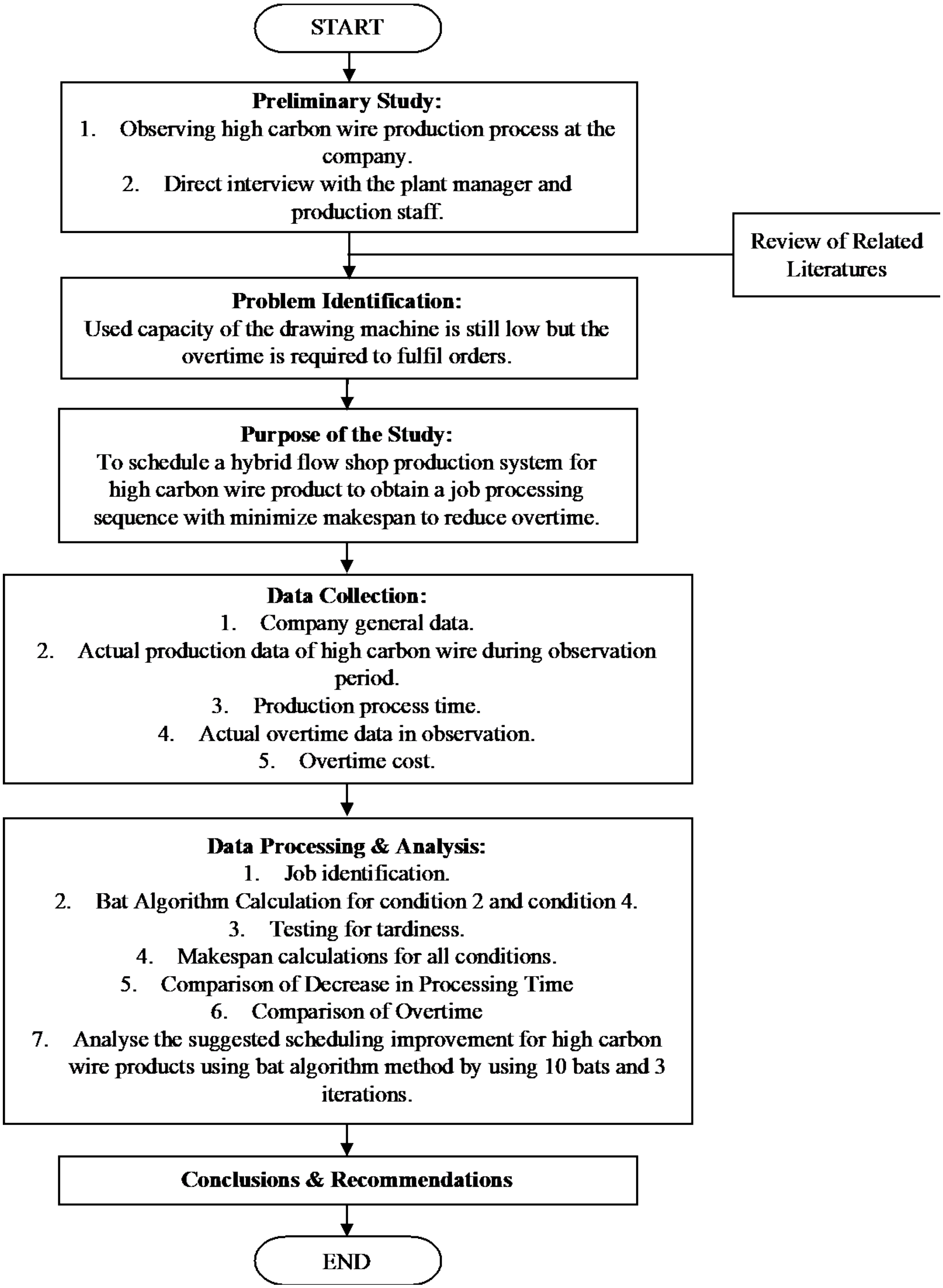
2.5. Overtime

Overtime is an effort to speed up the process of completing a task by increasing the number of working hours. Overtime can also be described as an effort to increase work productivity to accelerate the completion time of a job. According to the regulation, overtime means working hour that exceeds seven hours a day and forty hours in one week for six working days a week or eight hours a day and forty hours a week for five working days or any working hour in any rest day of the week or during an official holiday that is determined by the government.

In addition to the definition of overtime, the government has also issued regulations for overtime wage for all companies in Indonesia. All companies in Indonesia must comply to this rule when paying overtime wages to their employees. The regulations are the following:

- For the first hour of overtime, overtime wage is 1.5x of a worker hourly wage.
- For any consecutive hour of overtime passed the first hour, overtime wage is 2x of a worker hourly wage.

3. Research Methodology



4. Results & Discussions

As a basis for analysis, we would need to understand the scheduling characteristic that is applied by the company. Scheduling characteristic is categorized into two, job characteristics and machine characteristics.

Characteristics of the job are the following:

1. Each job has the same order of production.
2. Each job has different processing time.
3. Each job could be allocated to any machines depending on machine's availability.

Characteristics of the machines are the following:

1. Pickling machine works in batch with maximum weight per batch of 2.1 tonnes.
2. Pickling machine works simultaneously with 8 minutes interval time between batches.
3. All three drawing machines yield output in batches where each batch can produce a maximum of 700 kg.

In this research, we solve the problem using four conditions. The four conditions are based on the preliminary study findings where we find that the two core factors that are causing the problem are (1) machine operators who reduced machine capacity that causes a higher than standard production time and (2) ineffective scheduling method that caused a high makespan. The following table illustrate the conditions used in this study.

Table 1 Research Conditions

Condition	Processing Time		Scheduling Method	
	Actual	Standard	FCFS	*Bat Algorithm*
Condition 1	√		√	
Condition 2	√			√
Condition 3		√	√	
Condition 4		√		√

4.1. Job Identification

The first step of data processing is to define the job. Job definition is done based on the 22 work order letters issued by the company during the observation period. Each of the 22 orders have different weights. The result of job identification during the observation period is the following:

Table 2 High Carbon Wire Product Order Data

Job	Diameter (mm)	Weight (Kg)	Order Date	Job	Diameter (mm)	Weight (Kg)	Order Date
Job 1	2.24	10500	05/07/2018	Job 12	2.40	19400	23/08/2018
Job 2	2.40	98000	05/07/2018	Job 13	2.24	13300	28/08/2018
Job 3	2.40	39963	10/07/2018	Job 14	2.40	3400	28/08/2018
Job 4	2.40	59300	13/07/2018	Job 15	2.24	16600	25/07/2018
Job 5	2.40	28000	20/07/2018	Job 16	2.40	21700	09/08/2018
Job 6	2.40	14000	27/07/2018	Job 17	2.40	90600	16/07/2018
Job 7	2.24	22900	25/07/2018	Job 18	2.40	104800	30/07/2018
Job 8	2.24	20300	27/07/2018	Job 19	2.40	73100	03/08/2018
Job 9	2.24	27200	02/08/2018	Job 20	2.40	80650	01/08/2018
Job 10	2.40	14700	08/08/2018	Job 21	2.40	42900	13/08/2018
Job 11	2.40	17900	22/08/2018	Job 22	2.24	30400	15/08/2018

After jobs are identified, the optimum sequence of processing is obtained through scheduling using bat algorithm.

4.2. Bat Algorithm Calculation

When using bat algorithm in scheduling, we have to follow the steps below:

4.2.1. Initial Solution

Initial solution comprised of the initial position and initial velocity of a bat. Initial solution acts as a comparison for new solutions and also acts as a starting position of a bat. We randomly generate number between 0 to 1 for as many as a multiplication of the number of bats, number of jobs and number of processes.

4.2.2. Transformation and Makespan Calculation of the Initial Solution

This phase will change the initial position of the bat to become the sequence of jobs and machines by sorting from the largest to the smallest number in each element from each bats. Since there are 22 jobs and two processes, we obtained 1^{st} to 44^{th} rank. After the numbers are sorted from the largest to the smallest, 23^{rd} to 44^{th} rank is adjusted to be the second set of 1^{st} to 22^{nd} rank. The objective of this step is to obtain the sequence of the job processing. After the sequence of the job processing is obtained, then the makespan of each bat in the initial solution is calculated using equation 2. Makespan calculation is performed by allocation jobs to one pickling machine and three drawing machines based on machine's availability. To ease the process of job allocation, gantt chart is used.

The following is the result of the makespan calculation for the initial solution in the first iteration under the fourth condition.

Table 3 Initial Solution Makespan Calculation Result

Bat	Makespan (hour)	Bat	Makespan (hour)
1	485.832	6	465.733
2	459.231	7	465.337
3	478.514	8	471.455
4	474.187	9	469.274
5	463.962	10	455.827

4.2.3. Bat Transition

Each bat will then change position to find the most optimum solution. Equation 3, 4 and 5 then are used to obtain the bat's new positions. New makespan is then calculated based on the new position. We use the same equation 2 to calculate the new makespan.

4.2.4. Local Search

Local search is a process to find new solution amongst the best bats. In this phase not all bats will be picked. Only those bats that have a random number higher than the pulse emission of the bat will pass through local search. To find this position we use equation 6. Upon obtaining the position based on local search, then all bats that passed the local search will group together with the bats that did not make it through and create a new solution.

4.2.5. Comparison Between Initial Solution and New Solution

This stage will choose the optimum position of each bats as an updated solution to be used for the next iteration. A new solution is accepted as an updated solution if the makespan of the new solution is less than the current solution and the random number is smaller than the noise rate of the bat. The following the a table comparing the old solution with the new solution:

Table 4 Determining Updated Solution

Xi,j	Random	Ai	Makespan – Old Solution	Makespan – New Solution	Makespan – Updated Solution
X1,j	0.11	1	485.832	609.923	485.832
X2,j	0.786	1	459.231	467.92	459.231
X3,j	0.193	1	478.514	467.13	467.13
X4,j	0.292	1	474.187	491.63	474.187
X5,j	0.141	1	463.962	458.22	458.22
X6,j	0.352	1	465.733	467.92	465.733
X7,j	0.001	1	465.337	473.15	465.337

X8,j	0.606	1	471.455	474.81	471.455
X9,j	0.223	1	469.274	451.22	451.22
X10,j	0.197	1	455.827	475.98	455.827

4.2.6. Noise Update and Pulse Emission

Noise update and pulse emission is performed towards the bats that are chosen in the updated solution. Noise update and pulse emission is updated using equation 7 and 8. Results are the following:

Table 5 Noise Update and Pulse Emission

Xi,j	Ai	ri
X1,j	1	0.3
X2,j	1	0.3
X3,j	0.9	0.17803
X4,j	1	0.3
X5,j	0.9	0.17803
X6,j	1	0.3
X7,j	1	0.3
X8,j	1	0.3
X9,j	0.9	0.17803
X10,j	1	0.3
MEAN	0.97	0.26341

The table above showed that bats number 3, 5 and 9 experience a new noise rate and pulse emission. This happens because bats number 3, 5 and 9 were chosen as updated solution and hence the noise rate and pulse emission are updated.

4.2.7. Saving Optimal Solution

Updated solution is then saved to be used as in the next iteration as an old solution. If the iteration has reached its maximum then the bat algorithm is completed and will yield an output that has the smallest makespan. An iteration has reached its maximum when the next two iterations do not generate any updated solutions. As an example, the iteration for condition 2 and 4 stopped on the 3rd iteration because the 4th and 5th iteration do not generate any updated solutions. The job sequence for condition 2 and 4 after 3 iterations with 10 bats are the following:

Table 6 Processing Sequence

Condition	Machine	Processing Sequence
Condition 2	1	J1,J7,J19,J18,J20,J22,J11,J13,J4,J6,J2,J5,J12,J14,J15,J17,J21,J10,J8,J9,J3,J16
	2	J1,J20,J19,J13,J4,J5,J14,J18,J17,J6,J15, J12,J8,J9,J22,J3,J11,J7,J16,J2,J10,J21
Condition 4	1	J5,J3,J22,J6,J17,J4,J13,J18,J9,J7,J2,J1, J20,J15,J21,J10,J12,J16,J19,J8,J11,J14
	2	J5,J21,J2,J17,J6,J20,J9,J18,J22,J15,J3, J11,J1,J4,J8,J13,J19,J12,J7,J14,J10,J16

4.2.8. Testing for Tardiness

When we have obtained the best processing sequence for condition 2 and 4 based on bat algorithm, then we need to test each solutions for tardiness. Testing for tardiness aims to determine if the processing sequence deserve to the solution. If the test for tardiness yields positive, then the solution could not pass to the next step. If the solution does not pass, then we replace the solution with another updated solution with the second lowest makespan from the last iteration. The following the test results for tardiness in condition 2 and 4:

Table 7 Makespan Calculation

Job	Completion Date	Due Date

	Condition 2	Condition 4	
Job 1	02/09/2018	20/09/2018	25/09/2018
Job 2	27/09/2018	07/09/2018	30/09/2018
Job 3	22/09/2018	19/09/2018	22/09/2018
Job 4	09/09/2018	22/09/2018	25/09/2018
Job 5	10/09/2018	03/09/2018	14/09/2018
Job 6	17/09/2018	10/09/2018	19/09/2018
Job 7	23/09/2018	27/09/2018	28/09/2018
Job 8	19/09/2018	24/09/2018	24/09/2018
Job 9	20/09/2018	15/09/2018	22/09/2018
Job 10	28/09/2018	27/09/2018	28/09/2018
Job 11	23/09/2018	20/09/2018	28/09/2018
Job 12	18/09/2018	26/09/2018	29/09/2018
Job 13	08/09/2018	24/09/2018	29/09/2018
Job 14	10/09/2018	27/09/2018	29/09/2018
Job 15	18/09/2018	18/09/2018	21/09/2018
Job 16	24/09/2018	28/09/2018	28/09/2018
Job 17	17/09/2018	10/09/2018	24/09/2018
Job 18	14/09/2018	17/09/2018	22/09/2018
Job 19	06/09/2018	26/09/2018	29/09/2018
Job 20	04/09/2018	13/09/2018	24/09/2018
Job 21	29/09/2018	05/09/2018	29/09/2018
Job 22	21/09/2018	18/09/2018	28/09/2018

Based on table 7, we find that the solution from condition 2 and 4 is fitting and could pass to the next step because none of the 22 jobs yield positive for tardiness.

4.3. Makespan Comparison

After we calculate the makespan for condition 2 and 4, we then compare the makespan between all conditions. The following table is the comparison result:

Table 8 Makespan Comparison Table

Condition	Machine	Processing Sequence	Makespan (Minutes)	Makespan (Hours)
Condition 1	1	J2,J1,J3,J4,J17,J5,J15,J7,J6,J8,J18,J20, J9,J19,J10,J16,J21,J22,J11,J12,J13,J14	33049.98	550.833
	2	J2,J1,J3,J4,J17,J5,J15,J7,J6,J8,J18,J20, J9,J19,J10,J16,J21,J22,J11,J12,J13,J14		
Condition 2	1	J1,J7,J19,J18,J20,J27,J11,J13,J4,J6,J2, J5,J12,J14,J15,J17,J21,J10,J8,J9,J3,J16	32694.12	544.902
	2	J1,J20,J19,J13,J4,J5,J14,J18,J17,J6,J15, J12,J8,J9,J22,J3,J11,J7,J16,J2,J10,J21		
Condition 3	1	J2,J1,J3,J4,J17,J5,J15,J7,J6,J8,J18,J20, J9,J19,J10,J16,J21,J22,J11,J12,J13,J14	28685.35	478.45
	2	J2,J1,J3,J4,J17,J5,J15,J7,J6,J8,J18,J20, J9,J19,J10,J16,J21,J22,J11,J12,J13,J14		
Condition 4	1	J5,J3,J22,J6,J17,J4,J13,J18,J9,J7,J2,J1, J20,J15,J21,J10,J12,J16,J19,J8,J11,J14	26348	439.133
	2	J5,J21,J2,J17,J6,J20,J9,J18,J22,J15,J3, J11,J1,J4,J8,J13,J19,J12,J7,J14,J10,J16		

4.4. Comparison of the Decrease of Processing Time

We also compare the decrease of processing time for all conditions. The following is the comparison result:

Table 9 Comparison of the Processing Time Decrease

Condition	*Makespan* (Hours)	Processing Time Decrease (Hours)	Processing Time Decrease (%)
Condition 1	550.833	0	0%
Condition 2	544.902	5.931	1%
Condition 3	478.45	72.383	13%
Condition 4	439.133	111.7	20%

Based on table 9, we see that the largest rate of decrease happened under condition 4 where standard processing time is used and bat algoritm is used as a scheduling method. The decrease of processing time under condition 4 is 20%.

4.5. Comparison of Overtime

After comparing the decrease in production time, the last step of analysis is to compare the overtime required in each conditions. This step is done to align with the objective of this research of reducing overtime. The following is the comparison result:

Table 10 Required Overtime

Condition	Makespan (Hours)	Available Time (Hours)	Required Overtime (Hours)	Decrease in Overtime (%)	Savings in Overtime Cost
Condition 1	550.833	469	81.833	0%	0
Condition 2	544.902	469	75.902	7%	Rp 1,164,375
Condition 3	478.45	469	9.45	88%	Rp 11,643,750
Condition 4	439.133	469	0	100%	Rp 13,972,500

Based on the above comparison, then we can see that there is a 100% decrease in overtime under condition 4. By using the standard processing time and bat algorithm as a scheduling method, the company is able to complete orders without overtime.

5. Conclusions

1. Upon doing research and data processing, the shortest makespan is obtained in Condition 4.
2. The job processing sequence for Condition 4 is J5, J3, J22, J6, J17, J14, J13, J18, J9, J7, J2, J1, J20, J15, J21, J10, J12, J16, J19, J8, J11, J14 for machine 1 and J5, J21, J2, J17, J6, J20, J9, J18, J22, J15, J3, J11, J1, J4, J8, J13, J19, J12, J7, J14, J10, J16 for machine2. Total makespan with that processing sequence is 439 hours.
3. Based on the calculation using bat algorithm for Condition 4, the total production time is reduced by 112 hours or 20% from the actual production time.
4. With the decrease of production time by 112 hours, overtime is no longer required to fulfil orders and overtime is reduced by 100% from actual time.
5. The company is able to save production cost of IDR13,972,500,- every month if they apply bat algorithm and strictly supervise production to ensure that actual production time is not higher than standard processing time.
6. By applying Condition 4, the company is able to accept more orders due to a 30 hours time saved.

References

Baker K. R., *Introduction to Sequencing and Schedulling,* John Wiley and Sonsm New York,1974.

Bedworth, David D. And James E. Bailey, *Integrated Production Control System: Management analysis,* Design, 2nd ed, John Wiley and Sons, New York,1987.

M. K. Marichelvam, and T. Prambaharan, *A Bat Algorithm For Realistic Hybrid Flowshop Scheduling Problems To Minimize Makespan And Mean Flow Time*, vol.03, Oktober. 2012.

R. Hadi, dan Y. Nugrahaeni, *Aplikasi Penjadwalan Terapi dengan Metode FCFS pada Sixo reflexology,* vol.01, No. 2, Juni. 2016.

M. K. Marichelvam, T. Prambaharan, X. S. Yang, and M. Geetha, *Solving Hybrid Flow Shop Scheduling Problem Using Bat Algorithm*, vol.05, 2013.

H. M. Wang, F. D. Chou and F. C. Wu, "A simulated annealing for hybrid flow shop scheduling with multiprocessor tasks to minimize makespan", *The International Journal of Advance Manufacturing Technology*, Vol. 53, No. 5-8, pp.761-776,2011.

A. Nurhadi, dan M. Suryanto, Perbandingan Produktivitas Tenaga Kerja Konstruksi pada Jam Kerja Reguler dan Jam Kerja Lembur pada Pembangunan Gedung Bertingkat di Surabaya, vol.01,2015.

Biography / Biographies

Sumiharni Batubara is a lecturer in the Department of Industrial Engineering, Faculty of Industrial Technology, Trisakti University, Jakarta, Indonesia. She received her Bachelor's Degree in Industrial Engineering from Institut Teknologi Bandung in 1980 and completed her Master's Degree in Industrial Engineering and Management from Institut Teknologi Bandung in 1989. Now, she is the Coordinator for Production Planning and Inventory Control Module and Industrial Manufacturing Design Module. Her research interest is in production planning and inventory control, scheduling, and lean manufacturing.

Debbie Kemala Sari is a lecturer in the Department of Industrial Engineering, Faculty of Industrial Technology, Trisakti University, Jakarta, Indonesia. She received her Bachelor's Degree in Industrial Engineering from Universitas Islam Indonesia in 1998 and completed her Master's Degree in Business Administration from Gadjah Mada University in 2010 and her Master's Degree in Industrial Engineering from Indonesia University in 2015. Now, she is the Coordinator for Quality Control Module and Manufacturing Process Module. Her research interest is in production planning and inventory control, scheduling, quality control, supply chain.

Dhimas Aryo Wicaksono is a student in the Department of Industrial Engineering, Faculty of Industrial Technology, Trisakti University, Jakarta, Indonesia. He is a laboratorium assistant on Production Planning and Inventory Control Module and Industrial Manufacturing Design Module. His research interest is in production planning and inventory control and scheduling