

# The Unrelated Parallel Machine Scheduling with a Dependent Time Setup using Ant Colony Optimization Algorithm

Farida Pulansari<sup>1\*</sup>, Muhammad Dwi Retno Triyono

**Abstract:** The unrelated parallel machine scheduling (PMS) problem is essential for the manufacturing industry. Scheduling will save company resources, especially time management. By solving scheduling problems quickly and precisely, the company can get more profit. On that note, this paper focused on unrelated PMS problems, which did not consider the inherent uncertainty in processing time and set up time by minimizing the makespan and tardiness. This paper aimed to minimize the makespan and tardiness using timing considerations. This paper described how to schedule unrelated parallel machines using the Ant Colony Optimization (ACO) Algorithm approach. The ACO is beneficial for inherent parallelism problems and can provide fast and reasonable solutions. This study revealed that the results of ACO Algorithm scheduling were obtained under a steady condition in iteration 30467. This condition can be interpreted that the makespan and tardiness value is close to 2.75%. By minimizing the makespan and tardiness, the delay of product delivery to consumers can be anticipated. Moreover, a company can maintain customer satisfaction and increase its profit.

**Keywords:** Ant Colony Optimization (ACO) Algorithm, makespan, tardiness, parallel machine scheduling (PMS).

## Introduction

Economic growth has been skyrocketing as more countries are open to global trade. Nations race to innovate and develop their respective industries to compete for market domination. Indonesia joins the bandwagon and continuously improves its capabilities in the manufacturing industry and development. Many companies adopt a production optimization approach. There are important aspects to optimize production. Production scheduling is one of the most important aspects to seize market opportunities with the shortest production cycle and maximum profit [1]. Besides, production scheduling is the decision-making process associated with allocating resources to tasks on machines to optimize one or more scheduling purposes [2]. Scheduling problems have an important role in recent years due to the increased consumer demand for variety, reduced product life cycle, market changes with global competition, and rapid development of new technologies [3]. Production scheduling also has a vital role in supporting company activities. Scheduling functions to support continuous and smooth production in filling up market needs. Parallel machine scheduling (PMS) is a problem related to allocating a set of jobs to several machines to fulfill demands. PMS can generally be classified into identical, uniform, and unrelated parallel machine scheduling problems [4].

Scheduling unrelated parallel machines is a generalization of two other categories of which different machines perform similar functions but have different processing capabilities or capacities. In real life, the scheduling of unrelated parallel machines is a challenge for researchers and practitioners because the problem of NP-hard can minimize the makespan time for several jobs in unrelated parallel machines [5]. Several studies present the problem of unrelated PMS, which involves the makespan considerations [6,7,8,9]. The novelty of this paper focuses on the problem of scheduling the parallel machines that are not related by considering the inherent uncertainty in processing and setting uptime. The practical implication of this paper is how to minimize the makespan by scheduling the appropriate number of jobs and machines. On the other hand, another benefit is that a company can perform other work with a lower makespan to make it more profits. This research road map is presented in Table 1.

PT. Asia Plastic is a company engaging in plastic packagings, such as plastic bags, plastic bottle caps, and other plastic materials. The company uses injection molding and blow molding machines and several machines using power jet technology in producing plastic packaging. The use of machines with new technology and modern equipment will support the production process. The company's operation is based on the Make to Order concept (products manufactured based on customer orders). Consequently, the company's production is highly dependent on the demand from other companies. Moreover, the company's productions and engine needs are not fixed. One of the frequent obstacles is unpredictable (non-deterministic) work, making the

<sup>1</sup>Faculty of Engineering, Department of Industrial Engineering, Universitas Pembangunan Nasional "Veteran" Jawa Timur. Jl. Rya Rungkut Madya, Surabaya 60294 Indonesia. Email: farida.ti@upnjatim.ac.id, triyono.mdr@gmail.com

\* Corresponding author

**Table 1.** Research position

No	Scheduling types	Previous research	Purposes	Methods
1	Single machine	Gagné <i>et al.</i> [10]	To minimize total tardiness	GA, Simulated Annealing (SA), and Branch and Bound
		Song <i>et al.</i> [1]	To minimize the makespan by employing SMS problem with sequence-dependent setup time	EDD, SDST, and ACO
		Jia <i>et al.</i> [12]	To minimize the makespan and the total rejection cost on BPMs	ACO, LACO, and PACO
		Li <i>et al.</i> [13]	To minimize the makespan	Weighted Shortest Processing Time (WSPT) and ACO
2	Flow shop	Huang and Yu [14]	To create computational time and stability	ACO
		Qin <i>et al.</i> [15]	To reduce the rescheduling frequency	Rolling Horizon Procedure (RHP) and ACO
3	Job shop	Huang <i>et al.</i> [16]	To minimize the makespan	ACO and PSO
		Khan <i>et al.</i> [17]	To minimize the makespan	ACO
		El Khoukhi <i>et al.</i> [18]	To minimize the makespan	Dual-Ants Colony (DAC)
		Wang <i>et al.</i> [19]	to optimize the makespan for FJSP	Improved Ant Colony Optimization (IACO)
4	Identical parallel machine scheduling	Tavares Neto <i>et al.</i> [20]	To minimize the sum of outsources and delay costs	ACO
		Öztürkoğlu, [21]	To increasing processing times of jobs	ACO and SA
5	Unrelated parallel machine	Arnaout, <i>et al.</i> [22]	To minimize the makespan	Two-stage Ant Colony Optimization (ACO) and TS
		Jia <i>et al.</i> [23]	To minimize the makespan	A Fuzzy Ant Colony Optimization (FACO)
6	Unrelated parallel machine	Current (Pulansari and Triyono, 2021)	To minimize the makespan and tardiness	ACO, First Come First Out (FCFO) rules, and Earliest Due Date (EDD) rules.

planned production schedule needs to be changed. These constraints also lead to a longer completion time. So far, the companies' scheduling tends to apply the First Come First Out (FCFO) and Earliest Due Date (EDD) rules. These types of scheduling are deemed ineffective because the length of order fulfillment time often results in late order fulfillment, frequently idle machines, and frequently sudden schedule changes that can disrupt the company's smooth production activities. The company typically purchases raw material and starts the production planning process only when a client's order is confirmed. Because the company does not make forecasts and initial preparation, the number of orders is beyond capacity. Consequently, many production processes are exceeding the due date. The researchers were encouraged to investigate scheduling unrelated parallel machines to time processing and time management based on the previous discussion. This study aimed to plan order-based production scheduling in the production process and to facilitate completion. The researchers employed the Ant

Colony Optimization (ACO) Algorithm to solve parallel machine scheduling problems. In some cases, different ACOs, such as single machine scheduling [10,11,12,13], flow shop scheduling [14,15,16,17, 18,19], and parallel machine scheduling identical [20,21] could provide best results. ACO Algorithm was used to plan the production schedule according to the standard time required by the machine of each order type. This scheduling was implemented to get the best implementation time. Client orders were adjusted by the machine types and the production target's required numbers.

## Methods

Scheduling is necessary to reduce the allocation of operator power, machinery, and production equipment and make other aspects more efficient. Scheduling is pivotal to make decisions in a sustainable production process. The scheduling problem refers to allocating work to the machine with limited capacity

and amount on that condition. In general, scheduling problems can be explained as  $n$  jobs ( $J_1, J_2, \dots, J_n$ ) to process on  $m$  machines ( $M_1, M_2, \dots, M_m$ ). The time needed to process  $J_1$  work on machine  $M$  is  $P$ . Each job must be processed without quitting during the processing time. The machine can only handle one job at the same time and is always available from zero time. Pinedo [2] classifies parallel scheduling into: single-engine scheduling and parallel scheduling of parallel machines.

### Problem Formulation

In the case of PMS, a set of independent jobs must be scheduled on the machine without preemption. In this paper, the problem situation referred to each job with a processing time, release date, and due date. This paper also considered the order-dependent setup time related to the setup time when the machine switched production from one job to another. Without eliminating the general nature, this research employed mathematical modeling according to the ACO algorithm parameter. The difference between this study and Guniet [22] lay in the machines' performance, creating different processing times, release dates, and due dates. The goal of the previous studies was only a shorter total of work completion.

We assumed that if the work was scheduled on the machine, no adjustment time was required. However, there were some restrictions and assumptions.

1. During the process, there was no stoppage.
2. The machine was ready to operate and produce.
3. There were no damage and repair to the engine.
4. Each machine could only do one job at a time.
5. Each job could only be processed once.

Some notations are defined below.

- $n$  : Number of jobs to process
- $m$  : Number of machines
- $i$  : Index of jobs
- $k$  : Index of machines
- $Cmax$  : The maximum completion time of jobs, i.e., the makespan
- $C_i$  : The completion time of job  $i$
- $P_{ik}$  : The processing time of job  $i$  on machine  $k$
- $ST_{ijk}$  : The setup times of processing job  $j$  right after job  $i$  on machine  $k$
- $ST_{ojk}$  : The setup times of processing job  $j$  first on machine  $k$
- $X_{ijk}$  : A binary variable equal to 1 if job  $j$  is processed right after job  $i$  on machine  $k$ , 0 otherwise
- $X_{ojk}$  : A binary variable equal to 1 if job  $j$  is processed at the first time on machine  $k$ , 0 otherwise

$X_{jok}$  : A binary variable equal to 1 if job  $j$  is processed at the last time on machine  $k$ , 0 otherwise

$V$  : A very large constant

Based on the previous assumptions and notations mentioned, the problem in questions can be formulated as follows (see [24, 25]):

$$\text{Minimize} = Cmax \tag{1}$$

$$\text{Minimize} = Tmax \tag{2}$$

$$\tag{3}$$

Subject to:

$$Cmax = \max \sum_{i=1}^k C_{i,j} \tag{4}$$

$$\sum_{i=0}^n \sum_{k=1}^m X_{i,j,k} = 1, \forall j = 1, 2, \dots, n, \tag{5}$$

$$\sum_{i \neq j}^n X_{i,h,k} - \sum_{j \neq h}^n X_{h,j,k} = 0, \forall h = 1, 2, \dots, n; \forall k = 1, 2, \dots, m, \tag{6}$$

$$C_j \geq C_i + \sum_{k=1}^m X_{i,j,k} (ST_{i,j,k} + P_{j,k}) + V (\sum_{k=1}^m X_{i,j,k} - 1), \forall i = 0, 1, \dots, n; \forall j = 1, 2, \dots, n, \tag{7}$$

$$\sum_{j=0}^n X_{0,j,k} = 0, \forall k = 1, 2, \dots, m, \tag{8}$$

$$X_{i,j,k} \in \{0,1\}, \forall i, j = 0, 1, \dots, n; \forall k = 1, 2, \dots, m, \tag{9}$$

$$C_0 = 0, \tag{10}$$

$$C_j \geq 0, \forall j = 1, \dots, n. \tag{11}$$

Equations (1), (2), (3) and (4) are objective functions. Equation (5) ensures that each job must be processed only once. Equation (6) implies that each work has a predecessor and successor. Note that,  $J_0$  job is used to present the first job on the machine. Equation (7) ensures that the processing of a job must begin after the completion of its predecessor. Equation (8) ensures that each machine only has one job during the process. Equation (9) defines binary variables. Equation (10) implies that the completion time for a job is zero. Equation (11) ensures that the completion time of regular work is not negative.

### Scheduling-Based Ant Colony Optimization

Marco Dorigo first introduced the ant colony algorithm. It is a probability technique for solving computational problems by finding the best path through graphs. The behavior of ants inspired this algorithm in finding pathways from their colonies from looking for food.

In the real world, ants travel around randomly. When finding food, they return to their colonies while giving signals with pheromone traces. If other ants find the path, they will not travel randomly anymore but will follow the trail, return and strengthen the pheromone if they have found food in the end.

When an ant leaves its nest, it begins to schedule production orders until it finds food (which means that the required work has been scheduled). Ants'

order scheduling means that the given node is selected and put on the path to food.

An ant randomly chooses the next operation (or the next node to go) based on the processing time of the operation and on the number of pheromones available at the edge to connect the node of the ant's current position and other possible nodes. The more pheromones in the bow (tip) are the more likely the ant will choose it.

An ant's scheduling of all possible operations indicates that it has reached a food source, and one trip is completed. The solution found by each ant is analyzed and depends on the quality of the responses. Each ant pathway created will get a certain amount of pheromone according to the quality criteria previously set (the better the schedule is, the more pheromones will be stored in each arch on the track).

Since the probability of selecting a node also depends on the number of pheromones in the arc, better solutions tend to influence other ants to choose the same path in the future. When all ants complete the required number of trips, the scheduling process ends, and the best responses are presented [26]. Dorigo *et al.*, [27] revealed the ant algorithm steps, which were then adjusted to the parallel machine production scheduling application. These steps are presented as follows.

1. Initializing the parameters of algorithms

In the ant algorithm, several parameters entered as an initial initialization to perform the optimization process. Some of these parameters are:

- n** : jobs
- τ** : Pheromone constant
- α** : The controlling constant of the ant trail intensity
- β** : The visibility controller constant
- k** : ants
- m** : machines
- φ** : Pheromone trail evaporation
- t** : Index of iterations

2. The visitor route arrangement of each ant to each node.

3. Calculation of changes in the price of ant footprint intensity between nodes.

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^\alpha \cdot [\eta_{ik}]^\beta} & \text{if } j \in allowed_k \\ 0 & \text{s for others} \end{cases} \quad (12)$$

where,

$\tau_{ij}$  = Pheromone intensity on the path between node (job) *i* and node (job) *j*

$P_{ij}^k$  = Probability of path from a node (job) *i* to node (job) *j* by machine *k*

4. Job sequence search with the optimal makespan and total tardiness

The objective function can be calculated using eq:

$$Minimize: Cmax = \max(C_i) \quad (13)$$

where  $C_i$  is completion time job on machine *i*.

$$Minimize: Tmax = \max(T_{i,j}) \quad (14)$$

where  $T_{ij} = C_i - d_i$  is the tardiness time job on machine *i*, the  $d_i$  is the due date of *j* job.

5. Calculation of the price of ant footprint intensity of nodes in the next cycle

Update local pheromone

$$\tau_{ij} (1 - \varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0 \quad (15)$$

where,

$\tau_0$ : Initial pheromone value

Update global pheromone

$$\tau_{ij}(t + n) = \varphi \tau_{ij}(t) + \Delta \tau_{ij} \quad (16)$$

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (17)$$

where,

$\Delta \tau_{ij}^k$  : The quantity per unit length of the pheromone trace substance left on the arch (*i,j*) by ant-*k* at the time interval between *t* and *t+n*

The pheromone quantity is formulated by:

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ use an arch } (i, j) \\ 0 & \text{for others} \end{cases} \quad (18)$$

where,

*Q* : A constant

$L_k$  : Tour length produced by ants *k*

The coefficient  $\rho$  must be less than 1 to prevent the accumulation of unlimited pheromone traces. In his experiments, Marco Dorigo uses  $\tau_{ij}(0) = c$  (where *c* is a small positive constant number). The intensity of the trace at time 0 is set equal to that of a small positive number.

6. If the stop is fulfilled (convergent conditions on the route is produced by the previous route) or the max number of iterations has been completed, take the job/operation sequence with the smallest makespan and tardiness; if not, return to step two. If iteration > maximum iteration, go to step seven.

7. Terminating the ACO algorithm.

## Results and Discussions

This paper employed the ACO algorithm because it is exact and can quickly resolve unrelated parallel problems. Here is a set of instances for the numerical experiments. For preliminary investigation, the researcher compared the ACO with other methods to

**Table 2.** Processing time

Job	Processing Time	Due Date
1	50	150
2	20	66
3	13	39
4	64	192
5	43	129
6	70	110
7	53	159
8	55	165
9	95	185
10	83	149
11	54	162
12	56	168
13	60	180
14	77	131
15	28	84
16	24	72
17	25	75

**Table 3.** Results comparison of five heuristics algorithms

Comparison of average makespan and tardiness experiment results (hours)					
	GA	PSO	SA	ACO	TS
Running time	0.244	0.298	0.136	6.7	0.52
Makespan	202.8	202.1	209.9	184.1	207.3
Tardiness	15.3	16.8	58.6	12.4	35.7

investigate its effectiveness and efficiency. The compared methods are the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Simulated Annealing (SA), ant Colony optimization (ACO) algorithm, and tabu search (TS). The data of processing time is presented in Table 2. Comparing between the five methods (Table 3) showed that the ACO Algorithm has the smallest makespan and tardiness. We compare five heuristics methods to test whether ACO can produce the smallest makespan and tardiness (Tables 4 and 5). We use three parameters, i.e., number of jobs, processing time, and due date. Moreover, we test the setting ten times.

Data were collected through three techniques. First, the observation technique was conducted to collect data by directing observation on the studied object. Second, the interview technique focused on the employees by holding a question and answer about the object to help or provide an explanation about the research problem. Third, documentation was conducted by recording and reviewing the company's documents related to the research object.

In this research, the production process began on 19 July 2019. First, the raw materials were collected from the warehouse; then, they were taken to the mixer machine to mix with dyes if color specifications had been on demand. If no color specifications had been required, the raw materials were sent directly to the molding machine. The prepared raw materials

**Table 4.** Makespan produces by five heuristic algorithms

No	GA	PSO	SA	ACO	TS
1	216	201	215	182	242
2	209	211	215	183	187
3	209	194	231	187	187
4	206	218	207	181	215
5	206	198	198	178	214
6	206	192	236	188	214
7	194	197	195	186	187
8	194	192	223	183	215
9	194	215	186	186	219
10	194	203	193	187	193
Average	202.8	202.1	209.9	184.1	207.3

**Table 5.** Tardiness produces by five heuristic algorithms

No	GA	PSO	SA	ACO	TS
1	120	53	84	51	129
2	11	0	47	0	8
3	11	98	82	56	3
4	11	0	42	0	47
5	0	0	18	0	37
6	0	0	105	3	37
7	0	17	27	6	10
8	0	0	92	0	39
9	0	0	6	6	39
10	0	0	83	2	8
Average	15.3	16.8	58.6	12.4	35.7

had been put in the machine, and the production process started. The product that came out of the machine was checked and tidied first and then packed. The packaged product was taken to the warehouse. It waited for the release schedule before being sent to the consumer. The required time was the cumulative time starting from the raw material collection, product specification setting, production process to finished goods. The first work on each machine was considered ready and did not count the setup time. Production time was only calculated until all product requests were made.

The data for the case study, which was collected from PT. Asia Plastic are presented in Table 6.

We need to determine how many product types should be scheduled and how long the product could be completely produced using the dataset presented in Table 6. Additionally, we also need to discover how many machines need to complete product orders. The processing time of each product was obtained from calculations using the processing time of each job on each machine is depicted in Table 7.

Table 7 refers to a condition set on the machine used by the company. The cycle time is the required time for one injection, while the number cavity is the number of patterns that affect the number of prints in one injection. The data was employed to calculate the production process time for each job on each machine. The followings are examples of the calculation.

**Table 6.** Job and the processing time

No.	Product names	Quantity (pcs)	Processing Time (Hours)			Date of entry	Due date
			M1	M2	M3		
1.	Bottle caps 1000cc	80,000	60	44	41	02 July 2019	30 July 2019
2.	Pet fertilizer caps	100,000	72	53	51	03 July 2019	26 July 2019
3.	Jerry can cap 5 l	95,000	69	66	48	03 July 2019	27 July 2019
4.	Clock frame	17,000	166	166	161	03 July 2019	31 July 2019
5.	Clock body	17,000	166	166	161	03 July 2019	31 July 2019
6.	Seashell basket	21,000	103	103	103	07 July 2019	30 July 2019
7.	Caps M 500cc	95,000	71	53	50	08 July 2019	31 July 2019
8.	Bottle caps 1000cc	120,000	89	65	61	09 July 2019	07 August 2019
9.	Pet fertilizer caps	120,000	87	63	61	13 July 2019	02 August 2019
10.	Clock body	15,000	146	146	142	16 July 2019	01 August 2019
11.	Clock Frame	15,000	146	146	142	16 July 2019	01 August 2019
12.	Caps M 500cc	110,000	82	62	58	17 July 2019	08 August 2019
13.	Jerry can cap 5 l	115,000	83	80	58	18 July 2019	10 August 2019
14.	Seashell basket	30,000	146	146	146	19 July 2019	06 August 2019
15.	Pet fertilizer caps	95,000	69	50	48	19 July 2019	04 August 2019

**Table 7.** Processing time of each job on each machine

No	Product names	Machines					
		M1		M2		M3	
		Ct (sec)	Nc	Ct (sec)	Nc	Ct (sec)	Nc
1	Bottle caps 1000cc	32	12	31	16	29	16
2	Pet fertilizer caps	31	12	30	16	29	16
3	Jerry can cap 5 l	31	12	30	12	29	16
4	Caps M 500cc	32	12	32	16	30	16
5	Clock frame	35	1	35	1	34	1
6	Clock body	35	1	35	1	34	1
7	Seashell basket	35	2	35	2	35	2

Ct is the cycle time (seconds), Nc is the number of cavity (number of print patterns)

**Table 8.** Matrix of machine setup time

From/		Job (hours)														
To		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Job (hours)	1	0	5	5	7	4	6	5	3	5	4	7	5	5	6	5
	2	5	0	5	7	4	6	5	5	3	4	7	5	5	6	3
	3	5	5	0	7	4	6	5	5	5	4	7	5	3	6	5
	4	7	7	7	0	4	6	5	5	5	4	3	5	5	6	5
	5	4	4	4	4	0	6	5	5	5	3	7	5	5	6	5
	6	6	6	6	6	6	0	5	5	5	4	7	5	5	3	5
	7	5	5	5	5	5	5	0	5	5	4	7	3	5	6	5
	8	3	5	5	5	5	5	5	0	5	4	7	5	5	6	5
	9	5	3	5	5	5	5	5	5	0	4	7	5	5	6	3
	10	4	4	4	4	3	4	4	4	4	0	7	5	5	6	5
	11	7	7	7	3	7	7	7	7	7	7	0	5	5	6	5
	12	5	5	5	5	5	5	3	5	5	5	5	0	5	6	5
	13	5	5	3	5	5	5	5	5	5	5	5	5	0	6	5
	14	6	6	6	6	6	3	6	6	6	6	6	6	6	0	5
	15	5	3	5	5	5	5	5	5	3	5	5	5	5	5	0

$$\begin{aligned} \text{Job 1 on machine 1} &= \frac{\text{demand} \times \text{Ct}}{\text{Nc} \times 3600} = \frac{80000 \times 32}{12 \times 3600} \cong 60 \text{ hours} \\ \text{Job 1 on machine 2} &= \frac{\text{demand} \times \text{Ct}}{\text{Nc} \times 3600} = \frac{80000 \times 31}{16 \times 3600} \cong 44 \text{ hours} \\ \text{Job 1 on machine 3} &= \frac{\text{demand} \times \text{Ct}}{\text{Nc} \times 3600} = \frac{80000 \times 29}{16 \times 3600} \cong 41 \text{ hours} \end{aligned}$$

**Real Scheduling of the Company**

In planning regular production scheduling, companies use the rules of First Come First Out or Earliest Due Date. The company scheduling uses the First

Come First Out method in which the ordering is based on the arrival time. Meanwhile, the Earliest Due Date scheduling is based on the due date. A comparison of work order sequences between the rules of the First Come First Out and th Earliest Due Date is presented in Figure 1 and Figure 2 in the form of a Gantt chart. In addition to the Gantt chart, the comparison of scheduling sequences is also presented in Table 9.

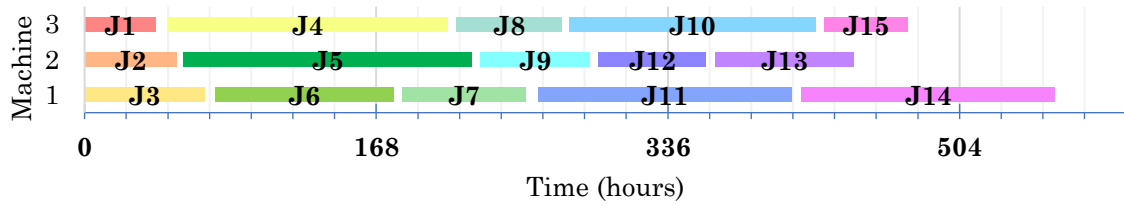


Figure 1. Gantt chart of first come first out method

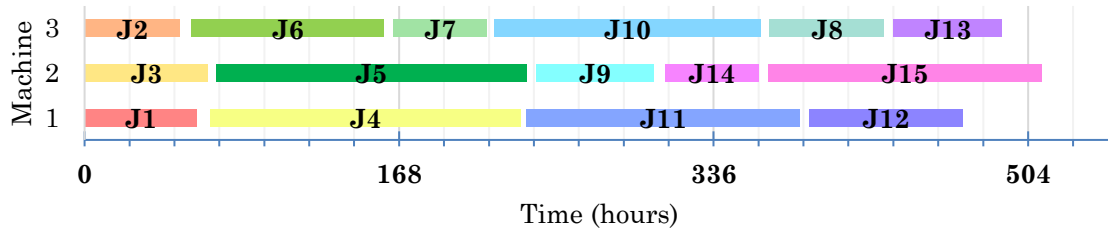


Figure 2. Gantt chart of early due date method

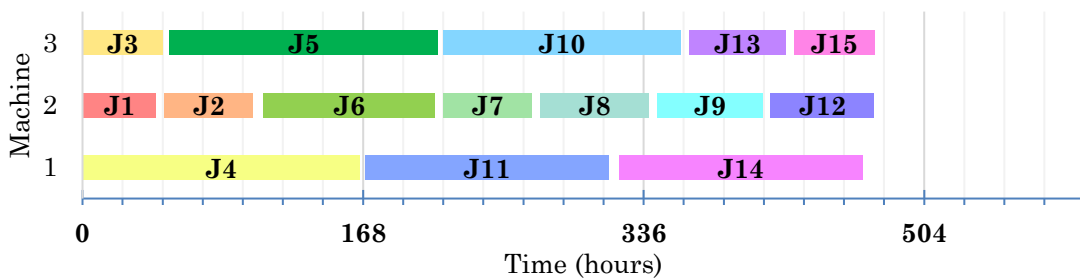


Figure 3. Gantt chart of the ant colony method

Table 9. FCFO vs EDD

Criteria	Method	
	FCFO	EDD
Maximum tardiness (hour)	175	85
Makespan (hour)	559	511
Number of late jobs (job)	4	5
Average tardiness (hours/job)	28.07	17.87

From Figure 1 and 2, it can be concluded that company scheduling using FCFO and EDD rules are quite different. The two figures have several differences. For example, the FCFO scheduling method in job 1 is done on machine 3 while the EDD scheduling method is done on machine 1. The comparison of the job execution sequence reveals that the EDD rules provide a more even machine loading compared to the FCFO scheduling method.

Table 9 shows the results of the company’s scheduling performance of FCFO and EDD methods. The scheduling results obtained four criteria of scheduling with the FCFO method. First, the maximum tardiness was 175 hours. This condition indicates that the total number of late jobs, the longest time, and work increased to 175 hours from the agreed limit. Second, the total of the makespan was 559 hours.

This condition indicates that the last work done on the injection molding machine was completed for 559 hours counted from the 0th time. Third, the number of jobs was more than four jobs. This condition means that the work on injection molding machines from the four jobs exceeded the predetermined time limit. Fourth, the average tardiness was 28.07 hours/work. This condition indicated that the average number of jobs to complete on a molding machine between July and August 2019 was related to the tardiness of 28.07 hours/work. The criteria for scheduling a company’s EDD method are 1) maximum tardiness of 85 hours, 2) the makespan of 511 hours, 3) the number of jobs exceeding five jobs, and 4) average tardiness of 17.87 hours/work.

### Ant Colony Optimization (ACO)

ACO method scheduling was done with the help of Matlab software. The parameters used in scheduling the ACO method were 1) pheromone constant ( $\tau$ ) of 0.5, 2) constant control of ant trail intensity ( $\alpha$ ) of 1, 3) visibility control constant ( $\beta$ ) of 3, 4) evaporation of pheromone traces ( $\rho$ ) of 0.5, 5) the number of ants ( $k$ ) of 30, and 6) maximum iteration of 50,000.

In this study, the company's initial machine scheduling applied the Ant Colony Algorithm method to produce more significant solutions than the company's current initial machine scheduling.

Pheromone is the weight on each trajectory and was used to place job  $i$  in the ant position  $k$ . The pheromone update calculation ( $\tau$ ) on the Ant Colony algorithm was employed to determine how the machines were used in the company. Then, the pheromone calculation was updated with several iterations until a pheromone value converged to the next iteration. When the stops were fulfilled (the value was converging) or the maximum iteration was completed, then the next sequence of production processes was taken with the smallest makespan. The scheduling based on ACO is presented in Figure 3.

The scheduling using the running program trial discovered that the makespan value was 473 hours. This result was better than that of the company's scheduling using FCFO and EDD, of which the makespan value was 559 and 511 hours. The ACO method scheduling sequence is different from the FCFO and EDD methods. The use of Gantt chart as output was more easily understood and looked more attractive. Job 1 ACO method was done on machine 2, FCFO is done on machine 3, and EDD job 1 was done on machine 2. In addition, Figure 4 describes the output graph to find out that the ACO iteration is already in a stable condition.

Table 10 was obtained from the program memory. It is known that the makespan value in the first iteration is 784 hours. The score changes in the second iteration for 590 hours and so on until the optimum value is obtained under steady conditions for 473 hours in iteration 30,467. Table 11 describes the consistency of the algorithm by running the algorithm multiple times and shows the mean and standard deviation of the ACO's performance.

Table 12 concludes that the company's scheduling using the FCFO rules and EDD rules differ from the ACO scheduling. The difference is shown by several examples. The FCFO scheduling of job 1 was done on machine 3, the EDD method was done on machine 1, and scheduling ACO job 1 was done on machine 2. Meanwhile, the comparison of the sequence of job execution can be seen from the Gantt chart showing that the ACO method gives more equitable machine loading compared to company scheduling using the FCFO and EDD methods.

Table 13 presents that the best scheduling performance was gained when the company applied the EDD method. The scheduling results obtained

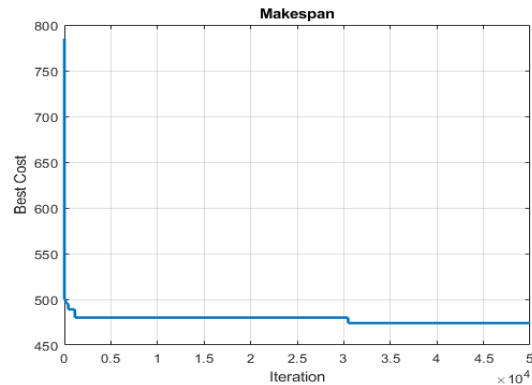


Figure 4. Chart Output of the Best Cost ant Colony Method

Table 10. The Process of finding the best makespan

Iteration	makespan
1	784
2	590
4	571
9	530
13	500
74	499
128	497
159	496
400	494
428	489
1158	480
30,467	473

four performance criteria for the company scheduling by EDD method. First, the maximum tardiness was 85 hours. This condition indicated that the completion of some jobs was late, and the longest time of a job exceeded the time limit for 85 hours from a predetermined limit. Second, the makespan was 511 hours. This condition indicated that the last job done on the injection molding machine was completed for 511 hours counted from the 0th time. Third, the completion of five jobs was late. This condition indicated that the completion of five jobs using the injection molding machines exceeded the determined time limit. Fourth, the average tardiness was 17.87 hours/job. This condition indicated that because the average number of jobs had been done on the injection molding machine from July to August 2019, the tardiness was 17.87 hours/job. The scheduling performance of the ACO method had four criteria. First, the maximum tardiness was 83 hours. Second, the makespan was 473 hours. Third, the number of late jobs was five jobs. Fifth, the average tardiness was 16.33 hours/job.

The next step was to compare the ACO method with the best scheduling method of the company (EDD) to find out the savings that the ACO method could offer. The comparison results are presented in Table 6. The



**Table 11.** ACO performance

Pheromone		Number of ants				
		10	20	30	40	50
0.25	Optimal solution	483	479	479	477	477
	Mean	484.8058	486.0711	483.4976	480.7344	478.8044
	Standard deviation	0.0181	0.0707	0.0450	0.0373	0.0180
	Running time (s)	13.8594	27.8125	42.8750	55.3594	73.2813
0.5	Optimal solution	480	480	479	478	476
	Mean	484.6061	484.7186	484.2551	483.1673	478.9825
	Standard deviation	0.0461	0.0472	0.0526	0.0517	0.0298
	Running time (s)	14.4688	28.3750	41.8906	58.8438	69.4063
0.75	Optimal solution	480	480	477	473	473
	Mean	483.1163	485.0188	483.8133	474.4594	473.7392
	Standard deviation	0.0312	0.0502	0.0681	0.0146	0.0074
	Running time (s)	13.9063	28.0625	44.5156	56.9688	69.0313

**Table 12.** Comparison of the job order in based on company scheduling method vs ACO

Machines	Jobs sequence		
	Company: FCFO	Company: EDD	ACO
M1	J3-J6-J7-J11-J14	J1-J4-J11-J12	J4-J11-J14
M2	J2-J5-J9-J12-J13	J3-J5-J9-J14-15	J1-J2-J6-J7-J8-J9-J12
M3	J1-J4-J8-J10-J15	J2-J6-J7-J10-J8-J13	J3-J5-J10-J13-J15

**Table 13.** Measure comparison of company methods and ACO

Criteria	Method		
	FCFO	EDD	ACO
Maximum tardiness (hour)	175	85	83
Makespan (hour)	559	511	473
Number of late jobs (job)	4	5	5
Average tardiness (hours/job)	28.07	17.87	16.33

comparison denotes that the ACO method had the best results because the ACO method was the smallest for all criteria of the company methods. The use of the ACO method enabled the division of jobs on all machines to be distributed more evenly. Consequently, the makespan values could be further minimized.

### Conclusions

This paper discussed the scheduling problems for parallel machines. A case study was conducted in PT. Asia Plastic. The company is currently applying FCFA and EDD. This study introduced the ACO to the company to improve its scheduling performance. The ACO had the makespan 38 hours fewer than EDD. Moreover, the ACO could reduce maximum tardiness up to two hours compared to EDD. Scheduling fair and equitable jobs on each machine could save the company's resources, such as time, energy, labor, and production costs. This condition indicated that the ACO algorithm could solve problems in unrelated parallel machine scheduling to non-identical.

This study has signified the practical implication in which the company could plan jobs and the appropriate machines. Thus, each job will not exceed the due date agreed by the company and consumers. On the other hand, the theoretical implication showed that many studies reported that ACO could be combined and compared with other methods according to the conditions and objectives. This study suggests that future researchers interested in investigating the same issue can transfer each job to another machine because the company's machines have the same function. The problem is that the company must continuously reset the machine setting if it produces high variance. The machine setup time is an essential factor in calculating the makespan and tardiness.

### References

1. Huang, M., Huang, R., Sun, B., and Li, L., Research on the Production Scheduling Optimization for Virtual Enterprises, *Mathematical Problems in Engineering* 2013, 2013, pp. 1–9.
2. Pinedo, M.L., *Scheduling Theory, Algorithms, and Systems*, 5<sup>th</sup> ed., Springer, New York, 2016
3. Sasikala, B., and Eswaramurthy, V.P., Ant Colony Optimization for Job Shop Scheduling Problem, *Proceeding of 8th IASTED International Conference On Artificial Intelligence and Soft Computing (ASC 2004)* 3(February), 2017, pp. 451–152
4. Torabi, S.A., Sahebjamnia, N., Mansouri, S.A., and Aramon Bajestani, M., A Particle Swarm Optimization for a Fuzzy Multi-objective Unrelated Parallel Machines Scheduling problem, *Applied Soft Computing Journal* 13, 2013, pp. 4750–4762

5. Lin, Y.K., Particle Swarm Optimization Algorithm for Unrelated Parallel Machine Scheduling with Release Dates, *Mathematical Problems in Engineering* 2013, 2013
6. Arnaut, J.P., Musa, R., and Rabadi, G., Ant Colony Optimization Algorithm to Parallel Machine Scheduling Problem with Setups, *4th IEEE Conference on Automation Science and Engineering, CASE 2008*, 2008
7. Low, C., and Wu, G.H., Unrelated Parallel-Machine Scheduling with Controllable Processing Times and Eligibility Constraints to Minimize the Makespan, *Journal of Industrial and Production Engineering* 33(4), 2016, pp. 286–293
8. Jia, Z., Yan, J., Leung, J.Y.T., Li, K., and Chen, H., Ant Colony Optimization Algorithm for Scheduling Jobs with Fuzzy Processing Time on Parallel Batch Machines with Different Capacities, *Applied Soft Computing* 75, 2019, pp. 548–561
9. Chen, S., and Wang, L., ACO-based Parallel Machine Scheduling Considering Both Setup Time and Run-based Preventive Maintenance with Reliability Constraints, *IEEE International Conference on Industrial Engineering and Engineering Management*, IEEE Computer Society (2019), 227–231
10. Gagné, C., Gravel, M., and Price, W., Scheduling a Single Machine where Setup Times are Sequence Dependent using an Ant-colony Heuristic, *Abstract Proceedings of ANTS'2000* (October), 2000, pp. 157–160
11. Song, H., Yi, S., Liu, M., and Qin, G., Single Machine Scheduling Based on EDD-SDST-ACO Heuristic Algorithm, *Journal of Intelligent Systems* 26(4), 2017, pp. 657–667
12. Jia, Z. hong, Pei, M. li, and Leung, J.Y.T., Multi-objective ACO Algorithms to Minimise the Makespan and the Total Rejection Cost on BPMs with Arbitrary Job Weights, *International Journal of Systems Science* 48(16), 2017, pp. 3542–3557
13. Li, H., Gajpal, Y., and Bector, C.R., Single Machine Scheduling with Two-agent for Total Weighted Completion Time Objectives, *Applied Soft Computing Journal* 70, 2018, pp. 147–156
14. Huang, R.H., and Yu, S.C., Two-stage Multiprocessor Flow Shop Scheduling with Deteriorating Maintenance in Cleaner Production, *Journal of Cleaner Production* 135, 2016, pp. 276–28
15. Qin, W., Zhang, J., and Song, D., An Improved Ant Colony Algorithm for Dynamic Hybrid Flow Shop Scheduling with Uncertain Processing Time, *Journal of Intelligent Manufacturing* 29(4), 2018, pp. 891–904
16. Huang, R.H., Yang, C.L., and Cheng, W.C., Flexible Job Shop Scheduling with Due Window - A Two-pheromone Ant Colony Approach, *International Journal of Production Economics* 141(2), 2013, pp. 685–697
17. Khan, S.A., Alam, M.A., and Umer, M., Resource Optimization in Job-shop Scheduling using Ant-Colony-Optimization Metaheuristic, *Proceedings of the Pakistan Academy of Sciences: Part A* 53(2A), 2016, pp. 131–144
18. El Khoukhi, F., Boukachour, J., and El Hilali Alaoui, A., The 'Dual-Ants Colony': A Novel Hybrid approach for the Flexible Job Shop Scheduling Problem with Preventive Maintenance, *Computers and Industrial Engineering* 106, 2017, pp. 236–255
19. Wang, L., Cai, J., Li, M., and Liu, Z., Flexible Job Shop Scheduling Problem Using an Improved Ant Colony Optimization, *Scientific Programming* 2017, 2017
20. Neto, R.F.T, Godinho Filho, M., and da Silva, F.M., An Ant Colony Optimization Approach for the Parallel Machine Scheduling Problem with Outsourcing Allowed, *Journal of Intelligent Manufacturing* 26(3), 2015, pp. 527–538
21. Öztürköglu, Ö., Identical Parallel Machine Scheduling with Nonlinear Deterioration and Multiple Rate Modifying Activities, *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)* 7(2), 2017, pp. 167
22. Arnaut, Jean-Paul., Rabadi, Ghaith., and Musa, Rami., A Two-stage Ant Colony Optimization Algorithm to Minimize the Makespan on Unrelated Parallel Machines with Sequence-Dependent Setup Times, *Journal of Intelligent Manufacturing*, 21(6), 2010, pp. 693-701.
23. Jia, Z., Yan, J., Leung, J. Y. T., Li, K., and Chen, H., Ant Colony Optimization Algorithm for Scheduling Jobs with Fuzzy Processing Time on Parallel Batch Machines with Different Capacities, *Applied Soft Computing Journal*, 75, pp.548-561.
24. Guinet, A., Scheduling Sequence-dependent Jobs on Identical Parallel Machines to Minimize Completion Time Criteria, *International Journal of Production Research* 31(7), 1993, pp. 1579–1594
25. Wang, L., Wang, S., and Zheng, X., *A Hybrid Estimation of Distribution Algorithm for Unrelated Parallel Machine Scheduling with Sequence-Dependent Setup Times*, 2016
26. Santos, L.P. dos, Vieira, G.E., Leite, H.V. dos R., and Steiner, Maria Teresinha Arns, Ant Colony Optimisation for Backward Production Scheduling, *Advances in Artificial Intelligence* 2012(i), 2012, pp. 1–12.
27. Dorigo, M., Maniezzo, V., and Coloni, A., Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 26(1), 1996, pp.29–41.