A Schedule Optimization of Ant Colony Optimization to Arrange Scheduling Process at Certainty Variables

Rangga Sidik[1], Mia Fitriawati[2], Syahrul Mauluddin[3], A.Nursikuwagus[4]
Dept. Information System
Universitas Komputer Indonesia
Bandung, 40132, Indonesia

***Abstract*—This research aims to get optimal collision of schedule by using certainty variables. Courses scheduling is conducted by ant colony algorithm. Setting parameters for intensity is bigger than 0, visibility track is bigger than 0, and evaporation of ant track is 0.03. Variables are used such as a number of lecturers, courses, classes, timeslot and time. Performance of ant colony algorithms is measured by how many schedules same time and class collided. Based on executions, with a total of 175 schedules, the average of a cycle is 9 cycles (exactly is 9.2 cycles) and an average of time process is 29.98 seconds. Scheduling, in nine experiments, has an average of time process of 19.99 seconds. Performance of ant colony algorithm is given scheduling process more efficient and predicted schedule collision.**

Keywords—Ant colony; optimization; scheduling; process; certainty variables

I. Introduction

Arranging schedule at university is more complex and difficult for a setting. Scheduling should be completed in time quickly and easily. Occupation of a classroom and lecturer should be avoided a collision. Meanwhile, the arrangement of the schedule should be submitted on time. Arranging a schedule at a university has to obtained by a regulation. Considering of schedule is depend on certainty variables. Regulation is stated to guide for ordering schedule [1][2]. In a digital era, all works want to be done quickly and easily. This cannot be denied considering that technology is growing and all facilities can be realized. In universities, one process that requires a touch of technology is scheduling. This variable is determined by taking into account certain boundaries and constraints. The process of making class schedules, each university has different terms and limits. While the process, accuracy is needed and must be on time. These problems are widely used as research using certainty variables [3].

Universitas Komputer Indonesia is one private university in Indonesia. There are many faculties at Universitas Komputer Indonesia. In managing of schedule, faculty of management has a problem in scheduling arrange. Time, classroom, lecturer activity, and timeslot are occupied as a factor success in managing schedule. Arranging of schedule collided between timeslot and another timeslot. Request for lecturer and lack of the classroom has caused a collision. In order to solve that problem, the research has been completed in many ways to solve. Using Ant Colony Algorithm (ACO), research has made efficient process and time reduction. ACO algorithm is used in much wide research [3]-[6]. Taking ACO in this research is made scheduling no collision and efficient in the process. In the execution of ACO, this research has been supported by many variables. The variables have included in ACO such as the number of lecturers, courses, classes, timeslot and time. Every execution in ACO, the process has always dependent by it variables. Completed research has made minimize collision in every timeslot and included lecture time submission.

II. Materials and Methods

A. Scheduling, Certainty Variables, Process

Refer to [4] has defined the schedule. Scheduling is an activity to manage the ordering job. Scheduling has many used by many fields. Research [2], [5]–[7], have presented about scheduling in ACO process. Research in robotic, traveling salesman problem, job shop scheduling, is introduced by [1], [8], [9].

In this research, we have proposed a term of certain variables. Certainty variables are parameterized in order to complete the schedule. We have used these variables in the ACO process. Considering, using these variables, is dependent on regulation at Universitas Komputer Indonesia. It also, the availability of ACO which can be processed in many variables.

The terminology of a process referred to manage the schedule. We have proposed these term, to ease of activity scheduling term. The process is also referred to as a sequence of a procedure. Process scheduling at Universitas Komputer Indonesia has regulated in optimization scheduling has always been following by regulation stated. It has made different with other scheduling. Every time, management was changed, it means, the process scheduling would be changed too.

B. An Colony Optimization (ACO)

Ant colony optimization is an evolutionary computational technique proposed by Dorigo et al. [10]. ACO is composed of three steps for a cycle: i) explore(), release ants for finding the destination, ii) pheromoneUpdate(), update pheromones the ants travel across the paths, iii) iteration(), comparing paths the ants found and find the best path if a predefined condition is met. To illustrated the ACO process, we have shown in Fig. 1. Fig. 1 is illustrated how pathfinding by ants ACO algorithm is taken from [10].

Directorate of Research and Community Service, Ministry of Research an Higher Education of Indonesia in 2018 Fiscal Year.

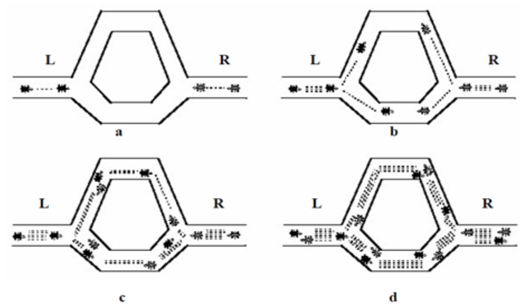

Fig. 1. (A) Ant Track is illustrated in two Ways, Right and Left Side. (B) Ants have Begun Track by two Paths. Each Side has Pictured Nature of Ant on Finding the Path. (C) Ant Sequence has followed the Track of Formerly Path. The Ant Track Previous has followed by other Ants with Pheromone Track. (D) The Figure has depicted the best Solution Path. The best Solution is the Path which has Minimize the Length of a Track. This Path has Chosen by Ant to Move between the Side and Another Side.

We have added some statements to align in scheduling problems. We have followed the step, to find the best solution in scheduling. At the algorithm below, we have adjusted the step to align in scheduling optimization at Universitas Komputer Indonesia.

Step 1

a) Parameters setting, defined as following

- Path intensity of ants and change (τ_{ij})
- City, we have defined as schedule order (*n*=175), including coordinat (*x*,*y*) or distance between timeslot and other timeslot (d_{ij})
- Schedule begin and target schedule (looping)
- Cycle ant constant (Q) = 1
- Ant track intensity constant (α) = 0.01, $\alpha \geq 0$
- Visibility constant (β) = 0.01 , $\beta \geq 0$
- Visibility between schedule =

$$1/[d_{ij} (\eta_{ij})] \quad (1)$$

- Ants (m)
- Evaporation of ant constant (ρ) = 0.03, $0 < \rho < 1$.
- Maximum cycle (*NCMax* = collision = 0) and never change as long as processing, for τ_{ij} always update for each *NC* =1 to *NC* = *NCMax*, or convergence reach.

b) Initialization for first schedule timeslot.

Afterward, τ_{ij} initialization done, and than ant *m* placed in the first schedule randomly.

Step 2

Put the first schedule containing the number of lecturers, courses, classes, timeslot and time in tabu list (temporary for searching schedule). Yields, for the first schedule from the first cycle, have to input as first element in tabu list. After completed in this step is defined indexing schedule by ant. It means variable tabuk1 can be contained indexing from 1 until n as defined in the first cycle

Step 3

Compilation of route for visiting schedule to each ant. An ant colony has distributed into the schedule, will be moved as many as 175 travels from origin schedule into the target schedule. Every ant has preferred schedule which did not in $tabu_k$. In final travel, $tabu_k$ has contained ant colony. If s is indexed visiting and schedule is stated N-$tabu_k$, then visiting probability would be counted as equation following [10]:

$$P_{ij}^{k} = \left[[\tau_{ij}]^{\alpha}.[\eta_{ij}]^{\beta}\right] / [\textstyle\sum_{k^{i\epsilon\{N-tabu_k\}}}[\tau_{ik'}]^{\alpha}.[\eta_{ik'}]^{\beta}] \quad (2)$$

$P_{ij}^{k} = 0, for\ next\ j$, where *i* is indexed for origin schedule and j is targeted schedule. $\tau_{ij}(t)$ means pheromone concentration of the schedule i and j (i,j) at time t. ηij is the heuristic factor that its value can be $1/d_{ij}$ (d_{ij} means the distance between schedule i and j) and is a constant, and we can see that the schedule of the shorter edge has a great probability to be chosen.

Step 4

a) Measure of route length for every ant.

Measure of length closed tour or L_k for every ant is done after one cycles is completed. Equation for count in one cycle completed as following [1]

$$L_k = d_{tabu_{k(n)},,tabu_{k(1)}} + \textstyle\sum_{s=1}^{n-1} tabu_{k(s),}, tabu_{k(s+1)} \quad (3)$$

Where d_{ij} is distance for schedule *i* and *j* that has formulated as following

$$d_{ij} = \left[\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2\right]^{\frac{1}{2}} \quad (4)$$

b) Searching minimum route.

In sequence of L_k has been counting, we have gained minimum closed length route or $L_{min}NC$ and whole closed length route or L_{min}

c) Updating pheromone in scheduling.

Ant footprint that has owned, would be stated as a track. Total of epavoration and difference ants, it has marked update of intensity. Equation for evaporation can be written as following [1]

$$\Delta\tau_{ij} = \textstyle\sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad (5)$$

$\Delta\tau_{ij}^{k}$ can be measured as following [1]

$$\Delta\tau_{ij}^{k} = [Q]/[L_k] \quad (6)$$

For *i* and *j* as origin schedule and target schedule. $\Delta\tau_{ij}^{k} = 0$, for other *i* and *j*.

Step 5

a) Counting intensity of ant footprints for next cycle.

Updating of intensity ant footprint has been caused by evaporation and differences of ant total. Updating value of for update global pheromone ant footprint has formulated as following [1]:

$$\tau_{ij} = \rho.\tau_{ij} + \Delta\tau_{ij} \quad (7)$$

b) Reordering intensity for ant footprint.

On the next cycle, we need to order intensity and than getting intensity value is 0.

Step 6

Empty of tabu list, repeat step 2. Tabu list of a temporary list has occupied into a tabu list for the new cycle. Every cycle, ant colony optimization, must empty the tabu list. It can occur if the sum of a cycle does not reach or does not convergence. The algorithm will be repeated at step 2 with the new footprint intensity. We have defined an equation for collision and bound of iteration as following

$$collision= \sum timeslot\ (l,d,t) + \sum timeslot\ (c,d,t) \quad (8)$$

Timeslot(l,d,t) and *timeslot(c,d,t)* is defined a value in schedule. Variable *l* means lecturer, *c* means classroom, *d* meand day, and *t* means time.

III. Result

A. Setting Variables Contsraint in Optimization Scheduling

In engineering knowledge for processing variables in ACO, our research has applied some assumptions. These assumptions have a pattern for the need of schedule and then justify the result. We are proposed some assumptions as following:

- Ant termonology is a variable combine among lecturer, courses, and classroom.
- Food resource is a pair of classroom, day, and time.
- Pheromone is a schedule that acquire by lecturer.
- Distance is a count from difference value among classroom, day, and time.
- Setting for first scheduling is activated by staff.
- Limitation of process is just implemented for theory subject and just in classroom.
- Setting day is just for Monday until Satuday with six days in total.
- Every schedule, result from ACO, can be implemented on three hours class.
- Limitation of occupancy in one day is only implemeted four timeslot and three hours class.
- Expecially for Friday, the third timeslot does not occupied.
- On Saturday, amount of timeslot is only three.
- Population ant colony is formulated as following

$$Population=c \times d \times t \quad (8)$$

c,d, and *t* are defined as paramater population like classroom, day, and time.

- Constraints of scheduling process as following

a) Lecturer is not permitted acquire in two schedules at the same day and time.

b) Scheduling do not process in two slot at the same day and time.

B. Ant Colony Optimization in Execution

In this section, how ACO is used by scheduling problem. In section 2, we have proposed six steps for completing the job. At the first step, we have shown at section 2.a for setting parameter. The continuous process, the next step, we can be tailored following procedural therein.

Step 1a

The result has shown at section 2, at step 1a. We continue to next step 1b.

Step 1b

Initialization first scheduling, we have constructed in randomize manner. Every variable, it has to transform with generating codification. A single value, we have preferred to code every variable. Afterward, the process continues to generate every single value into a tabular model. Combination table can be seen in table 1.

TABLE I. EXAMPLE GENERATE CODE INTO TABLE REPRESENTATIVE

Schedule	Lecturer	Course	Section	Classroom	day	time
J1	1	1	5	2	4	1
J2	4	3	4	1	2	2
J3	3	2	3	2	4	1
J4	1	4	2	1	2	2
J5	2	5	1	3	2	2
J6	5	1	2	1	1	3

At table 1, abbreviation J1 has meant a schedule at first timeslot. Single value, e.g 1,2, and 3, as codefication from variable that shown by column header.

Step 2

The process is started with implement randomize value into a table that called tabu list. Result at tabu list is the same as table 1. Difference both of table is, a tabu list is used for the transaction as long as the ACO process and the primary table is used for saving schedule from the ACO process.

Step 3

Compilation of route for visiting schedule to each ant. We have applied (2). Every timeslot can be calculated with using a measure of distance between timeslot and other timeslots. The result of this applied is probability visitation of track on every timeslot. Equation 2 has been applied into Table 1 and retrieved to count another equation. The result can be seen in Table 2.

TABLE II. PROBABILITY VISIT (P_{IJ}) TAILORED BY (2)

Schedule	*ηij J1*	*τij(t=0)*	$[\tau_{ij}]^{\alpha}.[\eta_{ij}]^{\beta}$	*Pij*	Cum.Prob
J1	0.00	0.01	0.0000	0.000	0.000
J2	0.22	0.01	0.0022	0.191	0.191
J3	0.50	0.01	0.0050	0.427	0.618
J4	0.00	0.01	0.0000	0.000	0.618
J5	0.29	0.01	0.0029	0.247	0.865
J6	0.16	0.01	0.0016	0.135	1

At Table 2 is shown about the result from applied (2). η_{ij} J1 is meant to factor heuristics at i and j times to timeslot schedule at J1. Concentration of pheromone at i and j, $\tau_{ij}(t=0)$ start from 0.01 at beginning of time (t=0). P_{ij} is symbolized for the probability of track visit.

Step 4a

Measure of route length for every ant. Arranging the route of each ant's visit to all points. At this stage, each schedule is attached to all schedules to calculate the distance. The distance between schedule can be calculated using (4). Example to calculate a distance between schedule and other schedules, we have used ordering pairwise among lecturer, classroom, day, and time. E.g J1 , lecturer is 1, classroom 2, day = 4, time = 1. If these values enter to table tabuk list, then distance will be calculated. The result on Step 4a can be seen at Table 3.

TABLE III. DISTANCE BETWEEN SCHEDULE USING (3) AND (4)

Schedule	**J1**	**J2**	**J3**	**J4**	**J5**	**J6**
J1	0.00	4.47	2.00	3.32	3.46	6.32
J2	4.47	0.00	3.36	3.00	3.46	2.00
J3	2.00	3.46	0.00	3.87	3.46	5.29
J4	3.32	3.00	3.87	0.00	3.00	4.36
J5	3.46	3.46	3.46	3.00	0.00	4.47
J6	6.32	6.32	5.29	4.36	4.47	0.00

Step 4b

This step has focused to find minimum distance. The process has tailored every value at Table 3 into a pairwise decision table. Table decision is a table that presented relation of between schedule and other schedules. We have calculated to find the minimum distance in every pairwise schedule with (3). Omitted pairwise, at Table 4 has presented with pair between schedule and other schedules as many as schedule. E.g, if the schedule has 175 timeslots, then the pairwise would be made 175 x 175 schedule. At Table 4, we have presented an example in the pairwise schedule. At Table 4, every pairwise route has used to find minimum distance L_k.

TABLE IV. AN EXAMPLE PAIRWISE SCHEDULE IN DISTANCE

Ant	Route Pairwise		Lk (Minimum Lenght)
1	J1	J5	3,46
2	J2	J4	3,00
3	J3	J5	3,46
4	J4	J2	3,00
5	J5	J3	3,46
6	J6	J4	4,36

Step 4c

Step 4c is implemented for updating pheromone in global. We have used (5) and (6) to calculate updating pheromone.

At Table 5, Q is taken from cycle constant=1. Column L_k is taken from table 4. Meanwhile $\Delta\tau_{ij}$ (5) defined as update for pheromone.

TABLE V. AN EXAMPLE UPDATIING PHEROMONE USING (5) AND (6)

Q	Lk	Q / Lk
1	3,46	0,14451
1	3,00	0,16667
1	3,46	0,14451
1	3,00	0,16667
1	3,46	0,14451
1	4,36	0,11468
$\Delta\tau_{ij}$		0,88154

Step 5a

Counting intensity of ant footprints for next cycle. At Table 5, we have presented global updating pheromone. On this step, we should counted updating value for new intensity. Using intensity as following:

τij new= (0.03 x 0.01) + 0.88154 = 0.8818

At the process, we have shown, using (7) has retrieved new intensity = 0.8818

Step 5b

In this step, running proces ACO, has only retrieved value of new intensity. New intensity is stored in memory process to use for the next process. Value of new intensity = 0.8818.

Step 6

The last step in ACO has repeated the process. After Step 6, all the table or tabu list must be clear. Iteration in ACO, we have defined that repeat would be iterated to Step 2 if there is no collision in the schedule. ACO which has designed, the iteration would be terminated with bound of collision = 0 in the schedule. As long as the collision is not 0, then ACO is iterated to the Step 2. Illustration for collision can be seen at Table 6 and Table 7.

At Table 6, at the grey cell, the table has presented lecturer who has collision with other schedule. Schedule *J2* and *J3* have known as collision. *Timeslot(l,d,t)* of *J2* = 4,4,3 and *timeslot(l,d,t)* of *J3* = 4,4,3. If ACO was found collision like Table 6, then ACO would iterate to step 2.

Another collision in *timeslot(c,d,t)*. At Table 7, we can be seen collision at classroom, day, and time.

TABLE VI. AN EXAMPLE OF COLLISION FOR LECTURER, DAY, AND TIME

Schedule	*l*	Courses	Class subject	*c*	*d*	*t*
J1	1	1	5	2	3	2
J2	4	2	4	1	4	3
J3	4	3	2	1	4	3
J4	6	5	1	3	4	3
J5	3	4	3	2	3	2

TABLE VII. AN EXAMPLE OF COLLISION IN CLASSROOM, DAY, AND TIME

Schedule	***l***	**Courses**	**Class subject**	***c***	***d***	***t***
J1	1	1	5	2	3	2
J2	4	2	4	1	4	3
J3	4	3	2	1	4	3
J4	6	5	1	3	4	3
J5	3	4	3	2	3	2

TABLE VIII. AN EXAMPLE OF SUMMARIZE COLLISION

Schedule	Collision		Total of Collision
	Timeslot(l,d,t)	*Timeslot(c,d,t)*	
J1	0	1	1
J2	1	1	2
J3	1	1	2
J4	0	0	0
J5	0	1	1

At Table 7, grey area at schedule *J2* and *J3*, it can be seen the schedule collision at classroom, day, and time. *Timeslot(c,d,t)* of *J2 = Timeslot(c,d,t)* of *J3*. *Timeslot(c,d,t)* of *J2* = 1,4,3 and *timeslot(c,d,t)* of *J3* = 1,4,3. It means, *J2* and *J3* has collided to each other.

At Table 6 and Table 7, the ACO process has shown a collision process. ACO has made the summarized table to retrieve where the schedule still reaches collision. We have prepared a summary table as storing table for schedule collision. At Table 8 is presented how many schedules have a collision on the first cycle.

At Table 8, in the column total of collision, there are still collision. ACO process is bounded by total of collision = 0. If the ACO process was presented like Table 8, then ACO process would be repeat to Step 2. In the process iteration, we have introduced a rule. The rule, only timeslot which have a collision, would be iterated.

IV. DISCUSSION

In this section, the results of the research that have been made, we used the term ant as a varabel of the term schedule. In this reasearch, teminology of Ant colony is known as a collection of schedules that have been arranged. The ACO application in the case under study has been changed at the time of the iteration. In ACO [10], iterations are carried out by comparing the value of the initial intensity with the intensity of the change. Iteration will stop when the difference in intensity meets the value of the conditions given. Whereas in the research conducted in this scheduling case, we used the number of collision for the iteration. The number of collision that we require is collision = 0. This is in line with research [1], [3], [10] which relies on intensity as a repetition requirement.ACO that has been implemented in the study, we made a test of the schedule used. The tests carried out can be seen at Table 9.

TABLE IX. TESTING PARAMETER ACO

Parameter Testing	Total
Total of Scheduling (testing scheduling)	175
Total of Classroom	9
Total of day	6
Total of timeslot	4
Population Scheduling	216

TABLE X. RESULT TEST ON 175 SCHEDULES

Number	Collision	Cycle	Time (Sec)
1	0	8	25,225
2	0	10	33,053
3	0	10	36,185
4	0	9	25,857
5	0	7	20,608
6	0	12	45,106
7	0	8	24,327
8	0	10	30,529
9	0	7	19,999
10	0	11	38,857

In Table 9, the testing parameter is a measure for optimizing ACO. Tests carried out, out of 216 population schedules, we only took 175 schedules. The optimization, for the ACO for the research conducted, can found the schedule with the minimum collision. This research was conducted by finding the value of optimization collision = 0. We can achieve the objectives of the study in accordance with the schedule optimization schedule.

The time given during the ACO process, provides efficiency in the preparation of SCHEDULING. We have tested 10 expereiment. Experiments conducted with a number of different cycles provide a significant time estimate [11]. More cycles give longer time. The experiments carried out also, have received collision = 0. Changes in parameters and intensities assumed by the number of collision provide fast time performance. In table 10, shows the experiment with the system that we have made.

In Table 10, we presented a conclusion that the application of the ACO algorithm can make scheduling optimal [12-14]. Furthermore, this research can make the scheduling process not occur in collision [15-17]. To reach the number of collision 0, on 10 trials the average requires as many as 9 cycles (rounding out of 9.2) with an average time of 29.98 seconds. The fastest time occurred in the 9th experiment, which was 19.99 seconds.

V. CONCLUSION

ACO implementation in research on scheduling optimization has an effect on the process of scheduling. Addition of calculations on intensity, provide experiment solutions by reaching the number of collision schedules to 0. Experiments carried out have reached an average time of 29.98 second. The time is reached on 9th cycles. We have concluded that improvement in ACO parameter would be improvement in scheduling. Primarily, on counting bound of iteration when process needs to loop. Proposed collision variable can caused looping reach collision = 0. Improvement of ACO for intensity have an influence to success achieve the schedule. This is properly aligned with problems that the aim of research.

References

[1] Marques, "Ant Colony Optimisation for Job Shop Scheduling," pp. 1–8.

[2] "SOLVING JOB SHOP SCHEDULING PROBLEM using AN ANT COLONY ALGORITHM Contribution / Originality," vol. 5, no. 5, pp. 261–268, 2015.

[3] E. Science, "Production scheduling with ant colony optimization Production scheduling with ant colony optimization," 2017.

[4] M. Tawfeek, A. El-sisi, A. Keshk, and F. Torkey, "Cloud Task Scheduling Based on Ant Colony Optimization," vol. 12, no. 2, pp. 129–137, 2015.

[5] W. Amir, F. Wajdi, A. Azman, and A. Wahab, "Solving Vehicle Routing Problem using Ant Colony Optimisation (ACO) Algorithm," vol. 5, no. 9, pp. 500–507, 2018.

[6] D. C. Ant, "Differential-Evolution-Based Coevolution Ant Colony Optimization Algorithm for Bayesian Network Structure Learning," 2018.

[7] Z. Wei, "The Research of Genetic Ant Colony Algorithm and Its Application," vol. 37, pp. 101–106, 2012.

[8] K. Akka and F. Khaber, "Mobile robot path planning using an improved ant colony optimization," no. June, pp. 1–7, 2018.

[9] J. A. C. Math and R. Bv, "Solving Traveling Salesmen Problem using Ant Colony Optimization Algorithm," vol. 4, no. 6, pp. 4–11, 2015.

[10] M. You-xin, Z. Jie, C. Zhuo, and A. B. Idea, "Production Scheduling," no. 1.

[11] WANG, Gang; GONG, Wenrui; KASTNER, Ryan. Operation scheduling: algorithms and applications. In: High-Level Synthesis. Springer, Dordrecht, p. 231-255, 2008.

[12] CHEN, Wei-Neng; ZHANG, Jun. An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol.39 no.1: 29-43, 2009.

[13] HUANG, Kuo-Ling; LIAO, Ching-Jong. Ant colony optimization combined with taboo search for the job shop scheduling problem. Computers & operations research, vol.35 no.4: 1030-1046, 2008.

[14] Atighehchian, A., Bijari, M., & Tarkesh, H. A novel hybrid algorithm for scheduling steel-making continuous casting production. Computers & Operations Research, vol.36 no.8, 2450-2461, 2009.

[15] Xing, L. N., Chen, Y. W., Wang, P., Zhao, Q. S., & Xiong, J. A knowledge-based ant colony optimization for flexible job shop scheduling problems. Applied Soft Computing, 10(3), 888-896, 2010.

[16] Arnaout, J. P., Rabadi, G., & Musa, R. 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), 693-701, 2010.

[17] Deng, G. F., & Lin, W. T. Ant colony optimization-based algorithm for airline crew scheduling problem. Expert Systems with Applications, 38(5), 5787-5793, 2011.