

An ant colony algorithm for universiti sultan zainal abidin examination timetabling problem

Ahmad Firdaus Khair, Mokhairi Makhtar, Munirah Mazlan, Mohamad Afendee Mohamed, Mohd Nordin Abdul Rahman

Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia

Article Info

Article history:

Received Aug 20, 2018

Revised Oct 30, 2018

Accepted Nov 19, 2018

Keywords:

Ant colony optimisation

Constraints

Examination timetabling

Heuristic

Pheromone

ABSTRACT

The real-life construction of examination timetabling problem is considered as a common problem that always encountered and experienced in educational institution whether in school, college, and university. This problem is usually experienced by the academic management department where they have trouble to handle complexity for assign examination into a suitable timeslot manually. In this paper, an algorithm approach of ant colony optimisation (ACO) is presented to find an effective solution for dealing with Universiti Sultan Zainal Abidin (UniSZA) examination timetabling problems. A combination of heuristic with ACO algorithm contributes the development solution in order to simplify and optimize the pheromone occurrence of matrix updates which include the constraints problem. The implementation of real dataset instances from academic management is applied to the approach for generating the result of examination timetable. The result and performance that obtained will be used for further use to evaluate the quality and observe the solution whether our examination timetabling system is reliable and efficient than the manual management that can deal the constraints problem.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Ahmad Firdaus Khair,
Faculty of Informatics and Computing,
Universiti Sultan Zainal Abidin,
Terengganu, Malaysia.
Email: firdauskhair@yahoo.com

1. INTRODUCTION

In the last decade, several of research regarding examination timetabling has been conducted over the worldwide by previous researchers. Generally, the definition of timetable can describe as schedule planning on particular events that take place in certain places. In academic, beside timetable for examination, it also can be used for organizing a particular subject or course. For this study, examination timetabling is describing a list that shows the times in the week at which specific examination is held. Typically, the distribution of examination timetables is depended on how the staff management is managing the information and set of data in every educational institution respectively. Regarding the timetables problem, the survey on the previous research concluded as non-deterministic polynomial-time hard (NP-hard) [1]. The examination timetabling problem is known as (ETP) that arises around the world which involves educational institutions whether schools, colleges or university. Based on the real-life examination timetable solving the scheduling problem for the high educational institutions are more complicated than the school. It appears that the problem of examination timetabling is to allocate examination into a limited number of timeslot and while the satisfying additional constraints. Typically, the examination timetable problem involves two types of hard and soft constraints that need to be satisfied in order to produce feasible timetable [2]. A conflict between two or more examinations assigns into limited timeslot causing difficulties

among the students. Previous researchers have suggested a solution where a penalty is given when there are circumstances that violation of conflict on two or more examinations is assigned at the same time and day [3]. To measure the gap and identify the availability of free amount of time, spreading the paper much as possible over examination duration could provide enough time for preparation and study that leads to increase the success of the student.

Survey on the literature, most of the constraints and problems were studied and solved by previous researchers using various types of the approach proposed. Some Genetic algorithms have been used to solve population of a chromosome which is represented as a solution for finding the feasible timetable. The cross-over operator tends to find suitable and produce an appropriate result for next population which depends on both parents and child [4–6]. Heuristic solution particularly proved that this kind of method provides solution examination time tabling by dealing the constraints even though not guarantee not optimal and perfect [7, 8]. A heuristic ordering is used to examination timetabling problem by comparing five different strategies and evaluate the best result [9]. Combination of Genetic and Heuristic also are great strategies which can overcome difficulties by maximizes the allocation and minimize as much as possible the violations of constraint in determining the best solution for timetabling problem [10]. A Constructive heuristic is presented for solving Universiti Malaysia Pahang examination timetabling by produce a good quality solution compared to existing software system [11]. The recent survey also found that memetic algorithm which is metaheuristic approaches have been constructed for deals exclusively with fixed length timetables [12]. Other examination timetabling problem also have been solved by the previous work with metaheuristic approaches such as Tabu Search (TS), GRASP (G), Great Deluge (GD) and etc. [13–15]. The evaluation performance of examination timetable has been compared with other variances of ACO and approach that has been proposed of Max-min ant system (MMAS) to determine the feasible solution and better result [16, 17].

ANCOTT program used two solutions which are ant rank-based system from ACO variance with heuristic ordering to determine the lowest number of soft constraints and reducing the amount of non-feasible timetables [9, 17]. A hybrid ant colony algorithm and a complete local search with memory heuristic were used for solving timeslot problems on the examination timetable that students assigned more than one examination simultaneously and maximize the available timeslot between two consecutive examinations [18].

In this research paper, the idea of the study is to satisfy all the constraints and provide a priority to assign exam events into a timetable. Therefore, the main objective of this study is to balance the distribution of timetable slots and student examination assignment. We presented with an algorithm solution which is based on ACO concept of real-life ant's behavior to solve our examination timetabling for UniSZA. This algorithm is implemented into the system where the result will be compared and tested with several datasets to analyze the effectiveness and flexibility of examination timetable. This paper is presented and organized into several sections. In the second section, an explanation about examination timetabling problem and discussion on UniSZA examination timetabling problem with details of constraints. A summary describes the definition of the problem in the third section. In the fourth section, the proposed method approach, ACO is explained. The experimental results are included in the fifth section and section sixth is the final result achieved. The last section presented overall conclusions and future work.

2. ANT COLONY OPTIMIZATION (ACO)

The ACO is a family of metaheuristic that can be used to solve the discrete optimisation problem [19]. Ant System (AS) is the earliest algorithm of ACO that has been proposed by Dorigo et. al in 1991 for solving traveling salesman problem (TSP) [20]. Then, this ACO evolved after a few years with emergence two variants of Ant Colony System (ACS) and the MMAS. In recent years, many researchers have been concentrated on this ACO algorithm that applied successfully on their various discrete optimisation problem and other combinatorial problems such as the Quadratic Assignment Problem (QAP), Vehicle Routing Problem (VRP), Graph Coloring Problem (GCP), and Job Scheduling Problem (JSP) [20].

ACO algorithm is inspired by trail laying for foraging and follows the highest concentration of pheromones in establishing the shortest path. Hence, the probability of using shorter paths from nest to resource depends on the high amount of pheromone. To describe the steps of ACO algorithm, an example of AS for TSP as showed in Figure 1 and three levels operation for execute the concept of ant behavior.

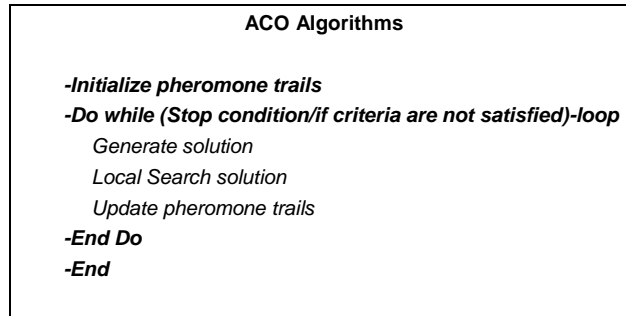


Figure 1. ACO algorithm

To initialize all the pheromones trails, the first level operation is to place ant randomly on node i with information that creates the solution. Then, ant moves through the node to node from its starting point of node i . At node i , by using transition probability, the ant v will decide the node which has not been visited yet for next node j . The formula of transition probability rules describe as follows:

$$P_{ij}^v(t) = \frac{(\tau_{ij}(t))^\alpha \cdot (n_{ij})^\beta}{\sum_{1 \in N_i^v} (\tau_{ij}(t))^\alpha \cdot (n_{ij})^\beta} \text{ if } j \in N_i^v \tag{1}$$

- N_i^v Set of nodes ant v which is at the state node i
- τ Amount of pheromone
- $\tau_{ij}(t)$ Pheromone trail intensity that link node i to j .
- $n_{ij} = 1/d_{ij}$ Visibility (heuristic information) desire for choosing closest node j when at node i ,
- α and β parameter to determine the influence factor the pheromone and visibility.

The second level operation is search solution that is optional if the ant finds the best solution. The pheromone trails update is the last level of operation ACO algorithm. There are circumstances in AS where the constant pheromone trails are evaporated to let the ants lay down pheromone after completion of its tour. The formula is defined and describe as follows:

$$T_{ij}(t) = (1 - \rho) \cdot T_{ij}(t - 1) + \sum_{v=1}^m \Delta T_{ij}^v \tag{2}$$

$$\Delta T_{ij}^v(t) = \begin{cases} Q / l^v(t) & \text{if } (i, j) \in T^v(t) \\ 0 & \text{if } (i, j) \notin T^v(t) \end{cases} \tag{3}$$

- m number of ants
- $(1 - \rho)$ evaporation rate where ρ is constant
- ΔT_{ij}^v amount of pheromone laid by ant ant v . on the edge i and j
- Q constant state and is the length of a tour that ant v
- $l^v(t)$ the length of a tour that ant

3. PROPOSE ACO APPROACH FOR OUR UETP

In this section, we briefly explained the flow process for managing the UniSZA’s examination timetabling problem with the representative solution and proposed approach. The ACO algorithm and heuristic is utilized UETP to find the solution to the heuristic information and pheromone trails for a feasible solution. The goal is to achieve best available time slot assignment for students by maximising the gap between two or more consecutive examinations and other respected constraints.

3.1. Representative Solution

We presented the example of timeslot indices on UniSZA examination week for our solution. Depend on each semester, the number of days and timeslots can be different for the examination week. For example, 3-week examination period in this semester and for the next examination may be different whether week or 2-weeks. Same goes to timeslot, if two timeslots in a day, means that two indices timeslots. So for example, with 3-week examination period has 45 timeslots. The total of 45 timeslots indices is

represented as 15 days without taking weekends (Saturday and Sunday) because there are no exams are held and in each day has 3 timeslots (morning, afternoon, evening).

(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,20,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57)

Figure 2. Timeslot indices

Based on represent indices timeslot in Figure 2, the timeslot number 1, 2 and 3 are denoted as day one, timeslot number 4,5 and 6 indicate as day 2, then followed by another timeslot number to denote as another day. This indexing timeslot is represented to provide and find a suitable timeslot for the students if a situation that they did not have gap-free time.

In Table 1, an example of room for the examination has been applied to a data set and with the details of its capacities. A process of heuristic is searching for the proper room to be chosen for examination based on the room details. The examination must assign into a room but if it is necessary, multiple of a room(grouping) can be utilized as long the examination is nearby to each other and capacity should be enough to fit the number of students. Distance matrix value between room is also given as shown in Table 1. In a particular case, there is an examination that needs to use more than one room to facilitate capacity of students. Therefore, a room grouping is provided by combining with other room to create a new total capacity shown in Table 2. To facilitate the room capacity, an arrangement for the room grouping is sorted decreasingly from large to small size. If the room grouping has been taken by an examination from the list, the other examination with a large capacity student can be assigned to any suitable room group according to the sorting capacity. It is necessary to organize the examinations for the large student with a priority so that during the assignation for choosing a room will be well arranged for the next examination to fit into the following capacity of the room. Note that, the value of information can be changed depending on the operations that involved

Table 1. Room Details and Distance Value Matrix

Room Group	Room	Capacity	DKU	AC21	AC19	AC20	AC23
FBIM	DKU	200	-	-	-	-	-
FIC	AC21	120	-	-	2	1	2
FIC	AC19	95	-	2	-	1	3
FIC	AC20	80	-	1	1	-	3
FIC	AC22	70	-	2	2	3	-

Table 2. Room Grouping Details

Room Grouping	New Total Capacity	Examination Split Value
(AC19,AC20,AC21,AC22)	565	3
(AC19,AC20,AC21)	495	2
(AC19,AC20)	200	1
(AC21,AC22)	190	1
(AC19,AC20)	175	1

3.2. Ant Strategies Movement

In order to demonstrate the ACO operation for the examination assignment, we provide an example matrix solution for timeslot and examination as shown in Table 3. Then, a sample of 7 examination datasets and set of number students taken an exam is provided to show the ant system iteration process which presented in Table 4.

Table 3. Matrix Solution of Timeslots and Exams

Timeslot \ Exam	E1	E2	E30
1					
2					
.....					
45					

Table 4. Example of 7 Sample Exams and Students

Student Group	Exams
{S1,S2,S3,S4,S5}	E1
{S6,S7,S8}	E2
{S6,S7,S8}	E14
{S3,S4}	E17
{S1,S2}	E16
{S1,S2}	E19
{S5}	E30

The Table 5 demonstrated the first stage of ant process to initialize the initial state randomly assigned the ant v is at the first node i in order to provide the timeslot and exams into proper day and session. As explained on the representative solution, D1 is assumed as the first day of the week consists of three timeslots (each day has three timeslots).

Table 5. Initial State Random Assign on Matrix

Date	1	2	3
Session			
D1	(E1,T1)		
D2		(E2,T5)	
D3			
D4			
D5			

Based on the previous section on ACO explanation, probability solution rule is used to determine the node which is not visited yet by ant v which initially was placed on node i examination will find other node j examination. The performance of ant v played an important role on the heuristic and pheromone information on the edge, where the probability preference of timeslot examination node j for the current node i depends on it. Table 6 showed the timeslots and exams that have been assigned suitable day after taking the probabilistic solution, and the conflict constraints in each examination are considered.

Table 6. Assign After Probabilistic Solution

Date	1	2	3
Session			
D1	(E1,T1)		
D2		(E2,T5)	
D3	(E19,T7)		(E30,T9)
D4			
D5		(E14,T14)	

3.3. Heuristics

The ACO approach for examination timetabling problem is supported with a heuristic method where taking the Largest degree and Largest Enrollment into account. This heuristic is utilized to speed up the process of finding a satisfactory solution even though the possibilities to achieve optimal solution does not guarantee but enough to provide a result. The description of the method is describing as follows:

LD: Exam is assigned first in the schedule if the exam has the most conflicts with other exams.

LE: Exam with large enrollment students is given priority to assign earlier.

3.4. Pheromone Update

A formula is given to interpret the best iteration according to the updated matrix value of pheromone trails. Hence, to get the strong concentration of the pheromone trails, additional some amount of value will provide result whether the best solution can be obtained based on the pheromone update rules. A new amount of pheromone is updated when the ant has been moved to all exam that already assigned to each timeslot on the timetable:

$$\Delta T_{ij}^v(t) = \begin{cases} Q / l^v(t) & \text{if } (i, j) \in T^v(t) \\ 0 & \text{if } (i, j) \notin T^v(t) \end{cases}$$

In order to make sure that the unlimited information pheromone at the end process of the ACO algorithm, the rules of pheromone evaporation process is applied to update and compute the trails:

$$T_{ij}(t) = (1 - \rho) \cdot T_{ij}(t - 1) + \sum_{v=1}^m \Delta T_{ij}^v$$

4. EXPERIMENTAL RESULT

The proposed ACO approach is executed in our simulation experiment for making an observation and identify most optimal result performance produced by data set. The algorithms were implemented and coded in a web-based computer system on Windows 10 with supported by CPU Intel Core i5-5200U 2.20 GHz and 4.00GB RAM. We were also using examples of real dataset which are applied to the simulation program system. The six of test case dataset instances are presented based on five details of exams, students, enrolled, timeslots and priority as shown in Table 7. All dataset is computed according to each number of experiment simulation that conducted to compare each result that has been obtained by the system.

Table 7. The FIC Datasets

ExpNum	Test Case	Exams	Students	Enrolled	Timeslots	Priority
1	FIC30_A	30	460	1416	45	Y
2	FIC30_B	30	460	1416	30	Y
3	FIC30_C	15	407	1058	30	Y
4	FIC30_A	30	460	1416	45	N
5	FIC30_B	30	460	1416	30	N
6	FIC30_C	15	407	1058	30	N

4.1. Discussion on Test Case

A priority is given on few of the test case to determine examination assigned for the large student's capacity or large enrollment of examination (LE) and to be scheduled earlier into the timeslots. The experiment has been carried out on the examination timetabling that applied with ACO approach to compute the examination timetable result with the test case information that has provided in the previous section. The result of the test case in Table 8 indicates that FIC30_A1 which is the priority given has the lowest value 0.96 of standard deviation and FIC30_B5 which is not given a priority produced 0.78 of standard deviation. Both results defined that the examination was scheduled inefficiently and provide good result rather than other test case result for the distributions of examination event in the timetable.

Table 8. Test Case Result on FIC Data Set Instances

Test Case	Density Conflict (%)	Mean	Var.	St.dev
FIC30_A1	2.12%	1.61	0.92	0.96
FIC30_B2	2.12%	1.91	0.98	0.99
FIC15_C3	1.42%	2.43	2.94	1.71
FIC30_A4	2.12%	2.00	1.17	1.36
FIC30_B5	2.12%	1.89	0.61	0.78
FIC15_C6	1.42%	2.62	1.90	1.38

4.2. Result

In Table 9, the final results of the FIC assignment examination timetabling were auto-generated to suit the relevant time slots and computed by considering all problems and solutions. The result also provides maximum separation for the consecutive exams and examination that has been given priority was assigned earlier into suitable timeslots. To depict examples of the data examination that has been assigned, information on student takes examination E1, E3, E16, and E19 time slots as shown in Table 10.

Table 9. Examination Timetable Result of FIC

Timeslots	Exams	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18	E19	...	E32	E33
T1																							
T2																							
T3																							
T4																							
T5																							
T6																							
T7																							
T8																							
T9																							
T10																							
T11																							
T12																							
T13																							
T14																							
T15																							
T20																							
T22																							
T23																							
T24																							
T25																							
T26																							
T27																							
T28																							
T29																							
...																							
T57																							

Table 9. Examples of Student Examination Assignment

Timeslot	T7	T2	T14	T22
Examinations	E1	E3	E16	E19

5. CONCLUSION

In this paper, it can be concluded that the implementation of ACO algorithm successfully obtained and solved the real practical examination timetabling problems faced by the FIC, UniSZA. Although the presented result does not guarantee the best scenario, at least the auto-generated processing is much better than manually process for the feasible solution on our UTime. It is proved that the approach can obtain good results depends on how the operation associated with the problem. Besides, the performance can be improved and enhanced by calibrating on the solution. Therefore, further testing and analyzing of this research will be able to ensure the establishment of the approach which helps resolve other variants of educational institution examination timetabling with the different requirement and specification for the future.

ACKNOWLEDGEMENTS

This work is partially supported by UniSZA (Grant RR008 CRIM/2016).

REFERENCES

[1] M. Doulaty, M. R. F. Derakhshi, and M. Abdi. *Timetabling: A State-of-the-Art Evolutionary Approach*. *Int. J. Mach. Learn. Comput.* 2013; 3(3): 255–258.

[2] W. K. Ho, A. Lim, and W. C. Oon. *Maximizing Paper Spread in Examination Timetabling using a Vehicle Routing Method*. 2001: 359–366.

[3] Ayob, M., Abdullah, S., & Malik, A. *A Practical Examination Timetabling Problem at the Universiti Kebangsaan Malaysia*. *International Journal of Computer Science and Network Security*. 2007; 7(9): 198–204.

[4] M. Golub and D. Jakobovi. *Exam Timetabling Using Genetic Algorithm*. 2009: 357–362.

[5] S. K. Jha. *Exam Timetabling Problem using Genetic Algorithm*. *Int. J. Res. Eng. Technol.* 2014; 3(4): 649–654,.

[6] W. Erben and P. Y. Song. *A Hybrid Grouping Genetic Algorithm for Examination Timetabling 2 The Hybrid Grouping Genetic Algorithm*. 2004.

[7] Z. Fu and A. Lim. *Heuristics for the exam scheduling problem*. 2000: 172–175.

[8] A. O. Adewumi, B. A. Sawyerr, and M. Montaz Ali. *A Heuristic Solution to the University Timetabling Problem*. *Eng. Comput.* 2009; 26(8): 972–984.

[9] T. Thepphakorn and P. Pongcharoen, “Heuristic ordering for ant colony based timetabling tool,” 2012, vol. 4, pp. 87–96.

-
- [10] N. D. Thanh. *Solving Timetabling Problem Using Genetic and Heuristic Algorithms*. *ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, 2007; 3: 472–477.
- [11] M. N. M. Kahar and G. Kendall. *The Examination Timetabling Problem At Universiti Malaysia Pahang: Comparison of a Constructive Heuristic with an Existing Software Solution*. *Eur. J. Oper. Res.* 2010; 207(2): 557–565.
- [12] E. K. Burke, J. P. Newall, and R. F. Weare. *A Memetic Algorithm for University Exam Timetabling* Springer, Berlin, Heidelberg, 1996: 241–250.
- [13] L. Di Gaspero and A. Schaerf. *Tabu Search Techniques for Examination Timetabling*. 2001; 2079 LNCS: 104–117.
- [14] S. Casey and J. Thompson. *GRASping the Examination Scheduling Problem*. Springer, Berlin, Heidelberg, 2003: 232–244.
- [15] H. T. Ā and S. Abdullah. *An Integrated Hybrid Approach To The Examination Timetabling Problem*. *Omega*. 2011; 39(6): 598–607.
- [16] A.R.Mirzaei and F.Djannaty. *Enhancing Max-Min Ant System for Examination Timetabling Problems*. *Int. J. Soft Comput.* 2008: 230–238.
- [17] M. Eley. *Ant Algorithms for the Exam Timetabling Problem*. *Pract. Theory Autom. Timetabling VI*. 2006; 3867: 167–180.
- [18] R. Abounacer, J. Boukachour, B.Dkhissi, A. El Hilali Alaoui. *A Hybrid Ant Colony Algorithm for The Exam Timetabling Problem*. 2016; 12: 15-42.
- [19] Dorigo, M., & Blum, C. *Ant Colony Optimization theory: a Survey*. *Theoretical Computer Science* 2005; 344(2–3): 243–278.
- [20] S. Katiyar and A. Q. Ansari. *Ant Colony Optimization : A Tutorial Review Ant Colony Optimization : A Tutorial Review*. 2015; 7 no. AUGUST: 35–41.