International Journal of Engineering & Technology, 7 (2.15) (2018) 132-135

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

Solving examination timetabling problem in UniSZA using ant colony optimization

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

Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, 21300 Terengganu, Malaysia
**Corresponding author E-mail: firdauskhair@yahoo.com*

Abstract

At all educational institutions, timetabling is a conventional problem that has always caused numerous difficulties and demands that need to be satisfied. For the examination timetabling problem, those matters can be defined as complexity in scheduling exam events or non-deterministic polynomial hard problems (NP-hard problems). In this study, the latest approach using an ant colony optimisation (ACO) which is the ant system (AS) is presented to find an effective solution for dealing with university exam timetabling problems. This application is believed to be an impressive solution that can be used to eliminate various types of problems for the purpose of optimising the scheduling management system and minimising the number of conflicts. The key of this feature is to simplify and find shorter paths based on index pheromone updating (occurrence matrix). With appropriate algorithm and using efficient techniques, the schedule and assignation allocation can be improved. The approach is applied according to the data set instance that has been gathered. Therefore, performance evaluation and result are used to formulate the proposed approach. This is to determine whether it is reliable and efficient in managing feasible final exam timetables for further use.

Keywords: *Ant colony optimization; Ant system; Examination timetabling; Scheduling.*

1. Introduction

In the literature, almost every researcher has put his/her efforts in investigating and finding a solution to different kinds of examination timetabling as examination timetabling problem (ETP) appears to be a huge problem in constructing an appropriate timetable for university exams. Exam timetabling has been a major issue involving all educational institutions, especially higher education institutions across all countries. Numerous research papers about timetabling have been considered as non-deterministic polynomial-time hard (NP-hard) where the complexity of computational amount on time is exponential with the size problem [1-2, 9]. A timetable can be defined as a planning for a meeting which includes a set of requirements comprising time, place, person, an events [2, 13]. In every educational institution, the distribution of timetables is different based on the management of information and features. In fact, university timetabling is not only used for examinations but also for the course [13].

In this research paper, the problems in constructing university exam timetables take into account certain requirements such as space allocations for large numbers of students, room types and test time and exam subjects. The idea of the study is to satisfy all of the set requirements in order 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. A difficult situation arises where there is a conflict between two or more consecutive examinations taken by a student in a limited space of time. There are also instances where students have more than one examination in the same time slot. Previous researchers have suggested penalty action if there is a conflict of students who need to take the examination in two or more by considering a situation [10]. It is believed that examination times which are spread evenly as much as possible contribute to student and lecturer performance. In a practical timetable, the additional problem of hard and soft constraints needs to be considered to overcome examination timetable problem [7]. Hard constraints can be described as the violation of any conditions, which cannot be changed at all costs. Soft constraints can be described as the desire that needs to be satisfied in any conditions.

In most of the existing works by the previous researchers, these constraints and problems were managed using various approaches. Genetic and heuristic combinations are great strategies, which can overcome difficulties in determining the best solution for scheduling problem [6]. Recently, a number of metaheuristic approaches have been constructed, for example the Tabu Search (TS), GRASP (G), Great Deluge (GD) and other adaptive search technique [3-5]. Max-min ant system (MMAS) has been utilised to find feasible solutions for examination timetabling and produced better performance results compared to other variances of ant colony optimisation (ACO) [10-11]. The ant rank-based system has been applied to a program called ANCOTT in order to see feasible timetables by determining the lowest number of soft constraints combined with heuristic ordering to reduce the number non-feasible timetables [10, 12]. A hybrid ant colony algorithm and a complete local search with memory heuristic were used in order to maximise free time as much as possible between consecutive exams for each student and lessen the conflict students face, where they cannot sit for more than one exam in the same timeslot [9].

Inspired by the behaviour of the ant stigmergic colony system of ACO, an automatic timetabling assigning system was generated to draw up a final feasible examination timetable for the university. This paper is presented and organized into several sections. In the second section, a summary describes the definition of the problem.

In the third section, the proposed method approach, ACO is explained. The experimental results are included in the fourth section and the last section is the overall conclusion.

2. Problem definition

University Examination Timetabling Problem (UETP) is known NP-hard which is considered in this scheduling and optimising study. A survey of literature has concluded that constructing universities' examination timetable is much harder than schools' timetables because inevitable different terms and conditions [13]. The following notation is proposed and will be used in this paper to describe the examination timetabling problem.

- *E*: Examination set for *n*
- *R*: Room set for *r*
- T: Timeslot set for *p*
- S_i: Number of students take exam *i*.
- P: Penalty for student which take exam more than one exam at same periods.
- Q_{irt}: Number of student take exam *i* assigned to room *r* in timeslot *t*.
- C_t: Room capacity for available seats *r* in timeslots *t*.
- S_{ij}: Number of students take exam *i* and exam *j*.
- Y_{it}: If exam *i* is assigned into the timeslot *t*, binary variable is equal to 1 or Y_{it} = 1 and otherwise is 0 or Y_{it} = 0

With the notation provided, the formula for examination timetabling can be computed as follows:

To find: $Y_z\ i = 1,2, \ldots n$ where *t* = 1,2,…p

$$min = \sum_{\omega \in \Omega} \sum_{i,j \in E, i \neq j} \sum_{t \in T, t > \omega} P_\omega c_{ij} y_{it} y_j (t - \omega) \quad \text{(i)}$$

$$\sum_{t=1}^{p} Y_{it} = 1 \quad (1)$$

$$Q_{irt} \leq Y_{it} C_t \quad (2)$$

$$\sum_{t=1}^{p} S_{ij} Y_{it} Y_{jt} = 0 \text{ where } i \neq j \quad (3)$$

$$\sum_{i=1}^{n} Q_{irt} S_i Y_{it} \leq C_t \quad (4)$$

$$Y_z = 0 \text{ or } 1 \quad (5)$$

The first formula is to maximise a student's free time between two or more consecutive examinations and balance the student's assignment. The following hard constraints must be satisfied with no consideration in order to construct feasible timetables.

1) Only one examination must be assigned in each time slot.
2) If there are no exams, there will be no reserved time slots assigned to any room.
3) No students should take more than one exam in the same time slot and on the same day.
4) The number of students taking an exam must not be more than the room's capacity limits.

Besides satisfying the hard constraints, violations of each soft constraint are considered by penalising equally.

1) There may be more than one examination in one room which can support the total capacity of students.
2) Examination course that has a large number of students must be assigned earlier in the timetable

3. Ant colony optimisation (ACO)

The ACO algorithm is a special metaheuristic approach and great solution for combinatorial optimisation problem [7, 10]. Early in the 19th century, the first original ACO algorithm known as AS was introduced for solving traveling salesman problem (TSP). After a few years, two variants of ACO existed after AS which is the Ant Colony System (ACS) and the MMAS. Nowadays, these approaches are widely applied on various discrete optimisation and other combinatorial problems such as the Quadratic Assignment Problem (QAP), Vehicle Routing Problem (VRP), Graph Coloring Problem (GCP) and Job Scheduling Problem (JSP) [8].

ACO algorithms are inspired by trail laying and concentration of pheromones which follows the behaviour of real ants in establishing the shortest path when foraging. While walking from the nest to resources (food), ant colonies release a chemical substance on the ground known as pheromone. These ants tend to follow the strongest concentration of pheromone and leave the pheromone trail to let other ants decide the path for getting the identified resources. Hence, the probability of using shorter paths depends on the high amount of pheromone.

From the observation, an example of TSP has been implemented using AS algorithms as in Figure 2. Following is the explanation of the basic principles of TSP. There are three basic principles for standard ACO algorithms. The first basic principle is the generated solution where the ant is placed randomly on node *i* which stores information on the solution that has been created. Beginning from its movement from node *i*, an ant travels from node to node. At node *i*, a transition probability is given for ant *v* to choose and move to the next node *j* which has not been visited yet:

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

From the above information, the probability rule formula can be called as the pseudo-random-proportional action choice rule. N_i^v can be defined as a set of all nodes to be visited by ant *v* which is currently at node *i*. The amount of pheromone is defined as τ, while $\tau_{ij}\ (t)$ is pheromone trail that connects node *i* to *j*. Given n_{ij} = 1/ d_{ij} is desirable for choosing node j when at node *i*, and two parameters; α and β are used determine the factor influencing the pheromone and heuristic information. The second basic principle is the local search solution which is optional for finding the appropriate information needed to be satisfied. The basic principle for generating a solution will be completed if all ants achieve their best solution. Finally, the last basic principle is updating the pheromone trails. There are circumstances in AS where the constant pheromone trails are evaporated to let the ants lay down pheromone after completion of its tour.

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

The formula is defined as trail pheromone decay where *m* is a number of ants, ρ is the evaporation rate of pheromone trail (0 < ρ < 1) and $\Delta\ T_{ij}^{v}$ depends on the amount pheromone laid at each edge *i* and *j* by ant *v*.

$\Delta\ T_{ij}^{v}$ (t) = {Q / l^v (*t*) if (*i, j*) ∈ T^v (*t*)

0 if (*i, j*) ∉ T^v (*t*)

Q is a constant state and l^v (*t*) is the length of a tour that ant *v* has created. Figure 1 shows all steps of the ACO algorithm.

```
ACO Algorithms
      -initialize pheromone trails
      -Do while (Stop condition/if criteria are not satisfied)-loop
          Generate solution
          Local Search solution
          Update pheromone trails
      -End Do
      -End
```

Fig. 1: Basic principles of ACO algorithms

4. Proposed approach of ACO for UETP

In this section, the adaptation of the ant algorithm for UETP is computed and evaluated after the ant finds the solution to the constructive heuristic information and leave pheromone trails for a feasible solution. The objective is to get the best available time slot assignment for students by maximising the gap between two or more consecutive examinations by using formula (i) and formula (1), (2) and (3) for other conflict orders with respect to the constraints.

4.1. Representative solution

There are two solutions presented to be used to consider the timetabling situation. The first solution is to sort the exams according to the largest degree. The number of examination conflicts, *i* is defined as the number of other examinations that must be taken by students with examination *i*. The second solution is the largest enrollment, where a student's particular exam has a priority. Table 1 shows the matrix for time slots that will be assigned according to the ant solution approach. Then, a sample of 7 examination datasets and students from 33 dataset instances of Faculty of Informatics and Computing (FIC), University of Sultan Zainal Abidin (UniSZA) is represented in Table 2 to show how the ant system iteration process works for the exams and students.

Table 1: Matrix solution

Timeslot / Exam	1	2	….	….	33
1					
2					
……					
45					

Table 2: 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

3.2. Ant construction movement

Based on the objective function formula (i), the configuration of ant system is utilised by the matrix in the initial state which is randomly assigned if ant *v* is at the first node *i* as demonstrated in Table 3.

Table 3: Initial state random assign on matrix

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

As explained in section 3, ant *v* which initially was placed on node *i* examination will find other node *j* examination which has not yet been visited using the probability solution rule. The probability of choosing the time slot for examination node *j* for the current examination node *i* depends on how well ant *v* performs on the heuristic and pheromone information on the edge. Table 4 demonstrates the timetable that has been assigned after the probability is randomly assigned and the value between each examination is considered.

Table 4: Assign after probabilistic solution

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

4.3. Pheromone update

The matrix value of pheromone trails is updated with a given formula in order to interpret the best iteration algorithm by the ants. Therefore, after the ants have already visited all the examination assignment on each suitable time slot the pheromone that connects exam node *i* to time slot exam node *j* of each edge will be updated with the new amount values after all conditions are satisfied:

ΔT_{ij}^{v} (t) = {Q / l^{v} (*t*) if (*i*, *j*) $\in T^{v}$ (*t*)

0 if (*i*, *j*) $\notin T^{v}$ (*t*)

The evaporation pheromone process needs to be applied to the algorithm to avoid unlimited accumulation of pheromones and allow the algorithm to reset the previous decision process. The updating of trails can be calculated as equation:

T_{ij} (*t*) = (1 - ρ). T_{ij} (*t* - 1) + $\sum_{v=1}^{m} \Delta T_{ij}^{v}$

5. Results and discussion

In order to demonstrate the strength of ACO approach, all algorithms were tested and coded in a web-based computer system of the UniSZA examination timetable supported by CPU Intel Core i3 2.30 GHz and RAM 2.00GB under Windows 8. 33 dataset instances were utilised and applied to the examination timetabling problem. Examples of test case dataset instance characteristics are provided based on four values as shown in Table 5. According to the timetabling problem, 45 time slots for three weeks of examination were set up represented by T1, T2, …. T45. Each week comprises of five days and each day has three time slots. In Table 6, the 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 showed that the maximum separation for two or more consecutive exams was produced. For example, a student which take for an exams E1, E3, E16 and E19 time slots as shown in Table 7.

Table 5: The FIC datasets

Test Case	Exams	Students	Enrolled	Timeslots
FIC33	33	1600	1600	45

Table 6: FIC examination timetable with using ACO approach

Timeslots / Exams	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18	E19	E20	…	E32	E33
T1																							
T2																							
T3																							
T4																							
T5																							
T6																							
T7																							
T8																							
T9																							
T10																							
T11																							

T12
T13
T14
T15
T16
T17
T18
T19
T20
T21
T22
T23
T24
…
…
T45

Table 7: Example of gap time for two or more exams

Timeslot	T7	T2	T14	T22
Examinations	E1	E3	E16	E19

6. Conclusion

In this paper, it can be concluded that the implementation of ACO algorithm successfully solved the real practical examination timetabling problems faced by the FIC, UniSZA. The optimization of examination timetable has been generated with examination spread evenly and the priority for the examination with large number of student is assigned earlier in the schedule. Although the presented result does not guarantee the best scenario, at least a feasible solution and optimal solution for our ETP was obtained. It was observed that this approach can produce great results according to how the data set problem is dealt with. Besides, the performance can be improved and enhanced depending on the adjustment solution. Therefore, further testing and analysing of this research will be able to ensure the establishment of the approach which helps resolve other variants of examination timetabling in the future.

Acknowledgement

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

References

[1] Carter MW, Laporte G & Lee SY (1996), Examination timetabling: Algorithmic strategies and applications. *Journal of the Operational Research Society* 47, 373–383.

[2] Cooper TB & Kingston JH (1995), The complexity of timetable construction problems. *Proceedings of the International Conference on the Practice and Theory of Automated Timetabling*, pp. 281–295.

[3] Casey S, Thompson J. GRASPing the examination scheduling problem. *Proceedings of the International Conference on the Practice and Theory of Automated Timetabling*, pp. 232–244.

[4] Di Gaspero L & Schaerf A (2000), Tabu search techniques for examination timetabling. *Proceedings of the International Conference on the Practice and Theory of Automated Timetabling*, pp. 104–117.

[5] Turabieh H & Abdullah S (2011), An integrated hybrid approach to the examination timetabling problem. *Omega* 39, 598–607.

[6] Thanh ND (2007), Solving timetabling problem using genetic and heuristic algorithms. *Proceedings of the* IEEE *8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pp. 472–477.

[7] Matijaš VD, Molnar G, Čupić M, Jakobović D & Bašić BD (2010), University course timetabling using ACO: A case study on laboratory exercises. *Proceedings of the International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 100–110.

[8] Katiyar S, Ibraheem AQ & Ansari AQ (2015), Ant colony optimization: A tutorial review. *MR International Journal of Engineering and Technology* 7, 35–41.

[9] Abounacer R, Boukachour J, Dkhissi B & Alaoui AE (2010), A hybrid Ant Colony Algorithm for the exam timetabling problem. *Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées* 12, 15–42.

[10] Eley M (2006), Ant algorithms for the exam timetabling problem. *Proceedings of the International Conference on the Practice and Theory of Automated Timetabling*, pp. 364–382.

[11] Djannaty F & Mirzaei AR (2008), Enhancing max-min ant system for examination timetabling problem. *International Journal of Soft Computing* 3, 230–238.

[12] Thepphakorn T & Pongcharoen P (2013), Heuristic ordering for ant colony based timetabling tool. *Journal of Applied Operational Research* 5, 113–123.

[13] Doulaty M, Derakhshi MF & Abdi M (2013), Timetabling: A state-of-the-art evolutionary approach. *International Journal of Machine Learning and Computing* 3, 255–258.