



Ant colony optimisation for solving university course timetabling problems

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

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

* Corresponding author E-mail: munimazlan@gmail.com

Abstract

Course timetabling is one of the most important activities faced by any educational institution. Furthermore, the course timetabling process is time-consuming and tiresome as it needs to be prepared for each regular semester. This paper aims to apply the Ant Colony Optimisation (ACO) method to solve the course timetabling problem. This approach is to optimise the properties of the course requirement and minimise various conflicts for the time slot assignment. This method is based on the life of the ant colony in generating automatic timetabling according to the properties (pheromones) such as time, student, lecturer and room, besides satisfying the constraints. The implementation of this method is to find an effective and better solution for university course timetabling. The result and performance evaluation is used to determine whether it is reliable in providing the feasible timetable.

Keywords: Ant Colony Optimisation (ACO); Constraints; Course timetabling; Optimisation.

1. Introduction

Course timetabling is one of the core operations faced by educational institutions [1]. Numerous approaches have been performed in the past decade in solving the problem of constructing timetables for universities. The university timetabling problems actually differ from each other based on the type of institution, the entities being scheduled and the constraints involved [2]. Both course and examination are part of a set of events that need to take place in the timetable. Timetabling is the process of producing a timetable, while satisfying the soft and hard constraints. The course timetabling problem usually deals with effective distribution of resources where the resources are lecturers, student, courses and particular rooms to a fixed time period. Apart from course timetabling, examination timetabling [3] is also employed in the university. Both course and examination timetabling are seamlessly integrating timetables to reduce conflicts of students and allotment of classes, which can be distinguished according to convenient timing and matters.

Besides, the problem of timetabling is to allocate the whole events to fix stipulated time slots and rooms which can satisfy the constraint. There are basically two types of constraints namely soft constraints and hard constraints [5]. Hard constraints must be satisfied in the problem. Soft constraints are rules that improve the quality of timetable but not absolutely essential.

In recent years, many research works have been devoted to ant colony optimisation (ACO) techniques in different areas and have been successfully used in many applications, especially problems in combinatorial optimization [4]. In ACO, each ant constructively builds a completion by several stepwise probabilistic decisions until the solution is revealed and reached. Due to its manageability and good performance, ACO has shown promising results for various timetabling applications and permutation scheduling problems. In this paper, we developed an Ant Colony

Optimisation algorithm to solve the university course timetabling problem. University course timetabling is classified as a non-deterministic polynomial (NP) hard problem. Solving a large course timetabling problem with a manual approach is extremely difficult and time-consuming. Nowadays, various methods are proposed to produce a better solution of the timetabling process. The course timetabling problem relies upon proper decisions and systematic plan strategies in order to obtain the best result. The remainder of this paper is organised as follows. Section 2 introduces the actual timetabling problems and section 3 elaborates the approach and algorithm used. Section 4 presents the results, while section 5 depicts the concluding remarks and future works.

2. University course timetabling problem

Timetabling in educational institutions is a crucial activity to schedule courses or examinations that consists of four finite sets namely a set of meetings, a set of resources (e.g. rooms, staff and students), a set of constraints and a set of available timeslots. The problem is to assign resources into appropriate time slots for students, lecturers and classrooms subject while maintaining the constraints. A feasible timetable should satisfy all the hard constraints, whereas soft constraints are desirable but not essential. The constraints need to be satisfied as much as possible. Thus, the problem description defines a broad range of problems where complexity significantly depends on the specific constraints defined.

A feasible timetable is one in which all events have been assigned a time slot and a room, so that the following hard constraints are satisfied: (1) No student attends more than one course at the same time; (2) The room should satisfy the features required by the course; (3) The number of students attending the course should be less than or equal to the capacity of the room and (4) Only one

course is assigned for each room at any time slot. In addition, a candidate timetable is penalised equally for each occurrence of the following soft constraint violations: (1) Students should not have exactly one event timetable on a day; (2) Students should not have to attend more than two consecutive events on a day and (3) Students should not have to attend an event in the last time slot of the day.

3. Ant colony optimisation method

Ant Colony Optimisation algorithm represents special solution approaches for combinatorial optimisation problems introduced by Marco Dorigo et al in early 90's [6-8]. The original ACO algorithm is known as the Ant System (AS). AS is a general purpose algorithm inspired by the study of ant colony behaviour. The basic idea of AS is to keep a population of ants that iteratively builds a solution by continually applying a probabilistic decision until a solution is found. The ants that find the best solution mark their path by putting some amount of pheromone on the edge of the path. Ants of the next iteration are attracted to follow the traversed good paths that contain pheromones resulting in a higher probability. AS ants have a memory (tabu list) that accumulates the visited components of their current path. There are two other successful variants of the ACO algorithm namely MAX-MIN Ant System (MMAS) and Ant Colony System (ACS). Nowadays, these approaches have been widely applied on variety of combinatorial problems. The depth introduction of Ant Systems is described by [9].

Ant Colony System [10] is different from other ant system algorithms, by which the state transition rule provides a very direct way to stabilise between the expedition of new edges and accumulated knowledge about the problem. Besides, the global trail updating rule is applied only to the edges that belong to the best ant tour. Meanwhile, a local pheromone updating rule is applied while ants construct the best solution. ACS is significant in the introduction of local pheromone update, while the pheromone update is performed at the end of the construction process. The local pheromone update is performed by all the ants after each construction tour step. MAX-MIN Ant System [11-12] is an improved version of the basic Ant System and widely used for its application in solving various combinatorial problems, especially the Traveling Salesman Problem. Therefore, MMAS is proven to have the property of effectively guiding towards promising regions of the search space by generating good initial tours which are often used as benchmarks to test the algorithmic ideas. MMAS has shown a good performance on various combinatorial problems.

A colony of k ants is used. Each ant constructs a complete timetable at algorithm iteration, where all of the hard constraints are satisfied at the highest possible extent in each of the generated solutions. The probability P_{ij}^k that the ant k , currently at node i will choose the node j is calculated using the random proportional rule given by,

$$P_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, j \in N_i^k \quad (1)$$

where τ_{ij} is the pheromone trail value on the edge connecting to node i to node j , η_{ij} is the heuristic value of that edge, α and β are the parameters that determine the relative influence of pheromone trail and the heuristic information and N_i^k is the feasible neighbourhood of ant k when it is at node i . After each ant completes its tour, the pheromone amount on each path will be adjusted with the pheromone trail decay Equation (2) and the edges that have been used by the ant (3),

$$\tau_{ij}(t) \leftarrow (1 - \rho)\tau_{ij}(t - 1) + \Delta\tau_{ij}(t) \quad (2)$$

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k} & , \text{if } (i,j) \in \text{tour done by ant } k \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} \quad (4)$$

where ρ is constant and L_k is the length of global best tour. The pheromone trail value is updated at the end of each iteration algorithm and evaporation is applied for each edge using rule (4).

4. Results and discussion

In this study, the ACO Algorithm approach is used and tested in the web based computer system of UniSZA Course Timetabling System that supported with CPU Intel Core i7-3610QM 2.3 GHz and RAM 4GB under Windows 7. We used the test case from the Faculty of Informatics and Computing (FIC) Datasets as shown in Table 1. The number of subjects in this faculty is 32. Besides, the number of students and the number of student enrolled are both 1600 respectively. The time slots that are applied for the course timetable are 10 timeslots.

The results that are based on the FIC datasets are successfully applied in the system. The performance of the algorithm on the FIC datasets is presented in Table 2 and 3. These outputs have different durations and contain wide varying numbers of events and attending students. Note that these datasets are instances of the timetabling problems that have appeared only at the Faculty of Informatics. From the results, it is shown that the gaps for each subject have been fairly generated.

Table 1: The FIC datasets

Test Case	Subjects	Students	Student Enrolled	Timeslots
FIC	32	1600	1600	10

Subjects \ Timeslot	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
S1	█	█						█		
S2						█	█			
S3	█						█			
S4			█	█						
S5		█	█					█		
S6	█									
S7			█	█						

Fig. 1: Course timetabling using ACO Approach in FIC per course

Subjects \ Timeslot	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
S8	█	█				█				
S9				█						
S10						█	█			
S11			█	█						
S12	█	█					█	█		
S13			█	█						
S14			█	█						

Fig. 2: Course timetabling using ACO Approach in FIC per course 2

5. Conclusion

In this research paper, it is concluded that the Ant Colony Optimisation algorithms prove to be capable of solving large real-world course timetabling problems. The implemented algorithm has comparable results like other high performance algorithms in the literature. The presented results might not be the best, but it is reliable to be applied for university course timetabling. This approach can produce better results according to the dataset instances problem involved. The results are analysed for future data use, which promise the effectiveness of the approach used and the decrease of time constraint. Thus, this approach helps to solve the course timetabling problems.

Acknowledgement

This work was supported in part by UniSZA under Grant Nos. R0008.

References

- [1] Thepphakorn T & Pongcharoen P (2013), Heuristic ordering for ant colony based timetabling tool. *Journal of Applied Operational Research* 5, 113–123.
- [2] 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.
- [3] Dimopoulou M & Miliotis P (2001), Implementation of a university course and examination timetabling system. *European Journal of Operational Research* 130, 202–213.
- [4] Hlaing ZC & Khine MA (2011), An ant colony optimization algorithm for solving traveling salesman problem. *Proceedings of the International Conference on Information Communication and Management*, pp. 54–59.
- [5] Burke EK, McCollum B, Meisels A, Petrovic S & Qu R (2007), A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research* 176, 177–192.
- [6] Maniezzo AC (1992), Distributed optimization by ant colonies. *Proceedings of the 1st European Conference on Artificial Life*, pp. 134–142.
- [7] Colomi A, Dorigo M & Maniezzo V (1992), An investigation of some properties of an "ant algorithm". *Proceedings of the Parallel Problem Solving From Nature Conference*, pp. 509–520.
- [8] Dorigo M, Maniezzo V & Colomi A (1996), Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26, 29–41.
- [9] Blum C (2005), Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews* 2, 353–373.
- [10] Dorigo M & Gambardella LM (1997), Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* 1, 53–66.
- [11] Stützle T & Hoos HH (2000), MAX-MIN ant system. *Future Generation Computer Systems* 16, 889–914.
- [12] Socha K, Knowles J & Sampels M (2002), A max-min ant system for the university course timetabling problem. *Proceedings of the International Workshop on Ant Algorithms*, pp. 1–13.