2009 Ninth International Conference on Intelligent Systems Design and Applications

Heuristic Factors in Ant System Algorithm for Course Timetabling Problem

Djasli Djamarus
Informatics Department
Trisakti University
Jakarta 11440, Indonesia
e-mail: djasli@trisakti.ac.id, ddjamarus@gmail.com

Ku Ruhana Ku-Mahamud
College of Arts and Sciences
Universiti Utara Malaysia
06010 Sintok Kedah Malaysia
e-mail: ruhana@uum.edu.my

Abstract—**This paper presents an algorithm that is based on ant system to solve the course timetabling problem. The problem is modeled using the bipartite graph. Four heuristic factors are derived from the graph characteristic, are used to direct ants as the agent in finding course timetable elements. The concept of negative pheromone was also applied to ensure that paths leading to dead ends are not chosen. The performance of this proposed algorithm is promising when comparison of performance was made with the original ant system algorithm.**

Keywords: Ant System Algorithm, Course Timetabling Problem, Heuristic, Pheromone.

I. INTRODUCTION

Course scheduling is an activity that appears every semester in education institution. Due to its difficulty and variation, solving course scheduling problem has been studied extensively and it has been noted that no less than 700 academic writings have been published in this area [1] since it was first introduced by Gotlieb in 1962 [5]. [15] defines course scheduling as a sub-class of assignment problem for which the events take place at educational institutions. The goal of the problem is to arrange events, each of which conducted by lecturer in a certain room in a definite period of time to be offered for students [14]. The scheduling activity usually produced a kind of table called timetable [3],[9], therefore this problem is also known as the course timetabling problem.

For a limited number of data, the course timetabling problem could be solved by any deterministic algorithm. However, as the NP Problem, for large number of data this kind of algorithm needs unreasonable time to run to come up with a solution. When this happens, it is usually said that the method failed to give a solution [4]. This reality encourages people to find out the nearest solution of the problem by using approximation algorithms

Currently the most promising approaches to solve the non-deterministic polynomial (NP) problems are local search and constraint programming approach. Local search approach starts by proposing an arbitrary solution, and then looks for better solution from the neighborhood repeatedly. Methods for solving the course timetabling problem which is also an NP problem include the taboo search, simulated annealing, and genetic algorithm. These methods are included in a class of algorithm called meta-heuristic [2],[3],[6]. The latter declares the problem as a set of constraints that define relations among variables that must be obeyed in search of a solution [11]. This approach usually starts by defining the solution as an empty set and then gradually includes some components that bear with the defined constraints into the solution [2] also called this approach as the constructive approach.

The more recent meta-heuristic algorithm that uses the constructive approach in solving such combinatorial problems is the Ant System algorithm. A type of Ant System algorithm that is intended to solve an optimization problem is known as the ant colony optimization [8]. It is a probabilistic algorithm that can be used to solve many computation problems, especially the NP problem. The Ant System algorithm is based on the ability of ant colony to find a shortest path between two places known as nest and food location using a kind of chemical substance called pheromone as the communication medium. Based on its function, in the real world there are some types of pheromone used by animals such as aggregation pheromone, sex pheromone, and fight pheromone [16]. There are two types of aggregation pheromone i.e. positive pheromone that is used to signal the colony to follow pheromone producer and negative pheromone that is used to signal the colony not to follow the pheromone producer [10],[12],[16]. [7] developed a model of computation for course timetabling problem that includes negative pheromone. The method produces better results as compared with algorithm that includes positive pheromone only.

TABLE I. COURSE ENTITY

Course	Section	Credit Unit
C00203	11	2
C00258	14	2
C00329	9	3
C00330	10	3
C00355	13	3

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.62

TABLE II. LECTURER ENTITY

Lecturer	Load (Section)	Max Hours
L1002	5	13
L1007	2	5
L1008	5	14
L1025	3	9
L1033	4	10

The rest of this paper is organized as follows. Section 2 explains the graph model that is used to model the course timetabling problem and discussion on several heuristics factors in solving the course timetabling problem is presented in Section 3. The proposed ant system algorithm is presented in Section 4 while experimentation and results are presented in Section 5. Concluding remarks are given in the last section.

II. GRAPH REPRESENTATION FOR THE COURSE TIMETABLING PROBLEM

The course timetabling problem in this study consists of four entities namely course, C = $\{c_1, c_2, \dots c_i\}$, lecturer, L = $\{l_1, l_2, \dots l_j\}$, time slot, T = $\{t_1, t_2, \dots t_m\}$ and room, R = $\{r_1, r_2, \dots r_n\}$. Each course timetable requires one element of course, c_i, one element of lecturer, l_j, one or more element of time slots, $t_k \dots t_m$ where $k < m$, and one element of room, r_n.

The maximum sections of a certain course and the maximum teaching hours of the lecturer are taken as the hard constraint, while the expertise level and the priority level of time slots for the lecturer are considered as the soft constraint.

The timetabling problem can be modeled as a graph G (V, E), where V is a set of vertices represent the entities course, lecturer, time slot and room (refer Figure 1). E is a set of edges which represents the relations between those entities. Relations between course c_i and lecturer l_j indicates lecturer l_j is willing and capable to handle course c_i, relation between lecturer l_j and time slot t_m indicates availability of lecturer l_j to conduct in time slot t_m, and relation between time slot t_m and room r_n indicates the availability of room r_n in the tm time slot.

By adding one vertex at both ends, each of which as nest and food, this graph is very similar to the Ant System model. This computation model gives a freedom to choose a path that connect the nest to the food, through any vertex that construct a tuple <Course, Lecturer, Intervals, Room> as a candidate of timetable element. The element that represented by sequence of edges in the path between nest and food, will be selected based on the number of pheromone deposited by the ant that moves from nest to food. Therefore, before the path selection is performed, there should be a process to construct the pheromone trail.

It is assumed that the rooms are of the same size and can accommodate any class size and of unlimited number. Since the rooms can be used for any time slots, there are as many as |T|*|R| edges connecting time slots and rooms. T and R represent the number time slots and number of rooms respectively. However, one timetable element usually uses a group of contiguous time slots, so there will be as much as |I|*|R| potential time slots - room pairs to be chosen, where I is a set of potential valid time interval that consists of contiguous time slots.

Natural hard constraints that prevent a lecturer to do more than one job in a certain time slots and the multiple used of a room in a certain time are represented by weight of the relation that is limited to 1. Hard constraints are handled by decreasing the weight of edges by 1 every time the edge is selected, so that there is no negative weight in the graph. The soft constraints that are preferable to be conformed are managed by an approach of the pheromone trail construction.

III. HEURISTIC ANALYSIS OF THE GRAPH MODEL

The graph model should be equipped with four heuristic factors that can be used to direct the ant colony movements in constructing a tuple <Course, Lecturer, Time Slots, Room> as a candidate of the timetable element.

The first factor is to give high priority in choosing a destination vertex that has the minimum number of edges leading to it from the source vertex. This is to avoid unused destination vertex as sometimes there is only one edge leading to it from a source vertex that has other edges leading to other destination vertices.

Decision on distributing the load of the source vertex to destination vertices is another heuristic factor to consider. The load of the source vertex has to be distributed evenly among the destination vertices that have edges leading to them. This hopefully will lead to a schedule that has courses with several sections being assigned to different lectures and also a lecturer will be assigned to several time slots for several courses thus avoiding overlapping time slot.

The third factor to consider is to give high priority to courses that require more time to deliver. These courses will be dealt first as compared to courses that require less time to deliver. This will ensure bigger time slots for these courses, thus avoiding the big time slots to be divided to several courses that require less time.

The last factor is to give higher priority to edges that represent the lecturer's expertise in selecting courses and also preferable time slots. This consideration will lead to better quality schedule.

IV. PROPOSED ANT SYSTEM ALGORITHM

The algorithm to construct the course timetabling is based on Ant System algorithm with modification on the pheromone update strategy. The amount of pheromone on the edges is limit to between minimum and maximum value as in [13]. Negative pheromone will be laid on edges that lead to the vertex that is fully utilized. This is to discourage the colony from traveling along that particular edge as this edge is considered as a dead end. Greater amount of pheromone will be placed to edges that have a higher priority to be selected as compared to edges with lower priority.

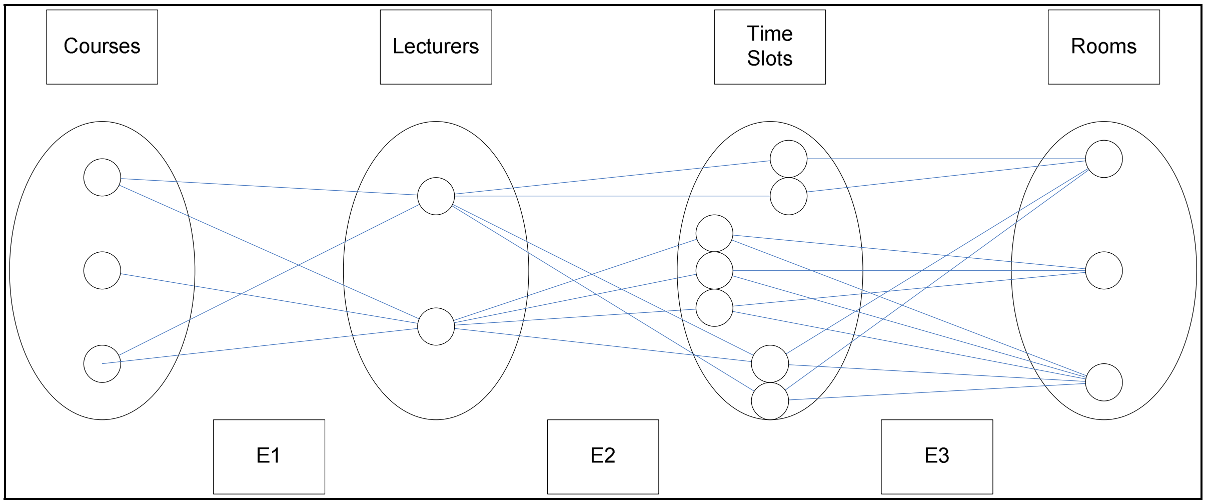

Figure 1. Graph Representation for Course Timetabling Problem

The construction and the selection phases are performed repeatedly until all sections are served or a time limit is reached. The algorithm consists of seven steps as follows:

Step 1: Develop graph model with vertices (Courses, Lecturers, Time slots and Rooms) and edges connecting the vertices, including their attributes.
Step 2: Initialize edges with pheromone.
Step 3: Initialize Schedule as empty set.
Step 4: For each ant, record all paths followed by ants, leave some (positive/negative) pheromone based on their (success/failure) journey from Courses to Rooms vertices.
Step 5: Select the best path based on the pheromone laid on the edges.
Step 6: Include in Schedule if selected path is valid.
Step 7: Repeat Step 4 until Schedule is complete or maximum iteration reached.

Completed Schedule will give a satisfying solution. However, a non satisfying solution is obtained if a maximum iteration is reached.

V. EXPERIMENT AND RESULTS

The data used for this study consist of data on courses, lecturers, time slots and rooms that are generated randomly. In this case study, 59 courses, 148 lecturers, 60 time-slots and 40 rooms are used to construct a course timetable that consists of 694 sections per week. A sample of the input data is presented in Tables 1 and 2 respectively. The number of section required to be opened for a certain course is indicated by 'Section' in Table 1. For example, course C00203 requires 11 sections to be opened and each section would require 2 contiguous time slots. The number of time slot needed for each course is represented by 'Credit Unit'. The teaching load for each lecturer is indicated by the number of maximum sections the lecturer could fulfilled, while maximum total lecturing time for each lecturer is indicated by 'Duration of Lecture' as depicted in Table 2.

Data for time slots are presented as T101, T102, …., T505, T506, T507. For example, T101 represents the first time slot on Monday while T507 represents the seventh time slots on Friday. There are only 5 working days in a week and 7 time slots in a day. Rooms are numbered from 1 to 4 since there are only four available rooms in this case. A sample of the relationship between Courses and Lecturers is depicted in Table 3 while Table 4 presents a sample of the relationship between Lecturers and Time slots.

The amount of pheromone on the edges is limited in the range of 10 (minimum) up to 1000 (maximum). The pheromone on each edges of success path will be increased by 10, and some preference factors that depend on heuristic factor of the edge. The pheromone on edges of failure path will be decreased by -10. Evaporation rate is assumed constant at a rate of 0.1%.

TABLE III. COURSE ENTITY

Course	Lecturer	Max Section	Expert Level
C00203	L1003	1	2
C00203	L1022	1	1
C00205	L1008	1	1
C00233	L1010	2	1
C00252	L1045	2	3
C00309	L1001	1	1
C00312	L1094	2	1
C00313	L1127	2	1
C00316	L1104	2	2
C00346	L1146	2	1

TABLE IV. COURSE ENTITY

Lecturer	Time slot	Unit	Priority
L1001	T101	3	1
L1001	T201	3	1
L1003	T101	3	1
L1003	T501	3	5
L1010	T404	2	4
L1045	T204	3	2
L1045	T503	3	5
L1094	T401	2	3
L1104	T106	3	2
L1127	T401	3	4

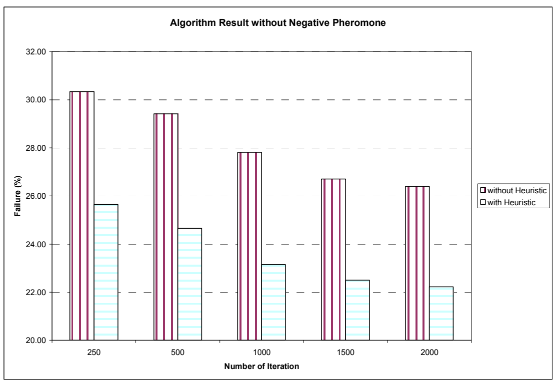

Figure 2. Algorithm Comparison without Negative Pheromone.

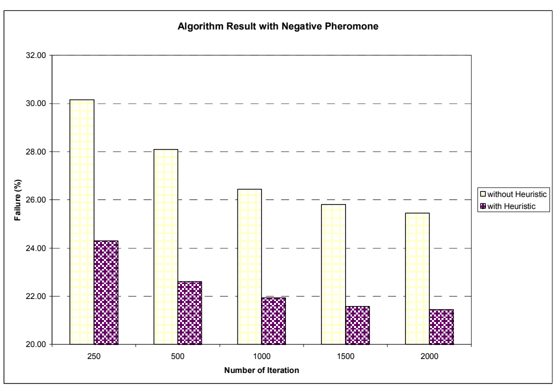

Figure 3. Algorithm Comparison with Negative Pheromone.

Performance of the proposed algorithms is compared to the original ant system algorithm. Number of iteration employed in the experiments is set to 250, 500, 1000, 1500 and 2000. Figure 2 shows the comparison of performances where the negative pheromone concept is not applied, while Figure 3 exhibits the case where the negative pheromone concept is applied.

Both results show that the proposed algorithm produces better performance as compared to the original ant system algorithm with or without the negative pheromone concept being applied. The inclusion of the four heuristic factors in developing the course timetable has significant contribution to the obtained results. Figure 4 displays the comparison of performance that applied the negative pheromone concept and inclusion of heuristic factors.

On average, the negative pheromone contributes 3.55% in the algorithm that does not employ heuristic factors and 5.61% in the algorithm that employ the heuristics. On the other hand, the heuristic factors contribute 19.04% in the algorithm without the negative pheromone and 21.42% in the algorithm with the negative pheromone.

VI. CONCLUSION

The inclusion of the negative pheromone concept and the consideration of four heuristic factors in the ant system algorithm for the course timetabling problem were able to improve the algorithm performance in term of scheduled sections as the course element. Bigger contribution to better performance was observed from the inclusion of the heuristic factors as compared to the negative pheromone concept.

Future work in improving the performance of the algorithm can consider the memorization the chosen element concept as applied in taboo search or swapping the chosen element like mutation operator in the genetic algorithm.

REFERENCES

[1] Bardadym, V. A. (1995). *Computer-Aided School and University Timetabling: The New Wave.* Paper presented at the First International Conference on the Practice and Theory of Automated Timetabling (ICPTAT '95).

[2] Blum, C., & Roli, S. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys, 35*(3), 268 - 308.

[3] Burke, E., & Petrovic, S. (2002). Recent research direction in automated timetabling. *European Journal of Operational Research – EJOR, 140*(2), 266 - 280.

[4] Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1994). *Introduction to algorithms. USA: The MIT Press.* USA: The MIT Press.

[5] Csima, J., & Gotlieb, C. C. (1964). Test on computer method for constructing school timetables. *Communications of the ACM, 7*(3), 160 – 163.

[6] Digalakis, J., & Margaritis, K. (2001, July 16-20, 2001). *A parallel memetic algorithm for solving optimization problems.* Paper presented at the 4th Metaheuristics International Conference (MIC'2001), Porto, Portugal.

[7] Djamarus, D., & Ku-Mahamud, K. (2008). *Ant System Algorithm with Negative Pheromone for Course Scheduling Problem.* Paper presented at the Eighth International Conference on Intelligent Systems Design and Applications 2008, Kaohsiung, Taiwan.

[8] Dorigo, M., & Stutzle, T. (2004). *An colony optimization*. USA: The MIT Press.

[9] Duong, T., & Lam, K. (2004, February 2004). *Combining constraint programming and simulated*

annealing on university exam timetabling. Paper presented at the International Conference RIVF'04, Hanoi.
[10] Lakhani, A. V. (2006). *Detours in Foraging Strategies.* The University of Sheffield.
[11] Marte, M. (2002). *Models and algorithms for school timetabling – A constraint-programming approach.* Fakultat fur Mathematik, Informatik und Statistik-Ludwig-Maximilians-Universitat, Munchen.
[12] Robinson, E. J. H., Jackson, D. E., Holcombe, M., & Ratnieks, F. L. W. (2005). Insect Communication: 'No entry' signal in ant foraging. *Nature, 438*, 442.
[13] Socha, K., Knowles, J., & Sampels, M. (2002). *A max-min ant system for the university course timetabling problem.* Paper presented at the ANTS 2002 – Third International Workshop on Ant Algorithms.
[14] Tripathy, A. (1984). School timetabling – A case in large binary integer linear programming. *Management Science, 30*(12), 1473 - 1489.
[15] Willemen, R. (2002). *School timetable construction – Algorithms and complexity.* Eindhoven University of Technology.
[16] Wyatt, T. D. (2003). *Pheromones and animal behaviour: communication by smell and taste*. UK: Cambridge University Press.