

Ant Colony Optimization for a Plan Guide Course Registration Sequence

Wael Waheed Al-Qassas¹,

Mohammad Said El-Bashir², Rabah Al-Shboul³

Faculty of Information Technology
Al Al-Bayt University, Mafraq, Jordan

Anwar Ali Yahya⁴

Science and Information Systems
Najran University
Najran – 61441, Saudi Arabia

Abstract—Students in universities do not follow the prescribed course plan guide, which affects the registration process. In this research, we present an approach to tackle the problem of guide for plan of course sequence (GPCS) since that sequence may not be suitable for all students due to various conditions. The Ant Colony Optimization (ACO) algorithm is anticipated to be a suitable approach to solving such problems. Data on sequence of the courses registered by students of the Computer Science Department at Al Al-Bayt University over four years were collected for this study. The fundamental task was to find the suitable pheromone evaporation rate in ACO that generates the optimal GPCS by conducting an Adaptive Ant Colony Optimization (AACO) on the model that used the collected data. We found that 17 courses out of 31 were placed in semesters differing from the semesters preset in the course plan.

Keywords—Ant colony; optimization; guide plan; university course registration

I. INTRODUCTION

The aim of this paper is to propose an algorithm that extracts a guide plan for course sequence from the registration behaviors of university students in order to solve the problem of disobedience of the students of the sequence of courses set in the course plan guide.

The course plan guide is a sequence of courses ordered by semester that the students should follow, semester by semester, during their study. Problems in following the sequence of courses to register according to the guide plan periodically arise in universities. This sort of problem is known to be NP-hard.

In this paper, we focus on students of the computer science (CS) department in an effort to find the optimal plan guide for course sequence as a case study. The proposed algorithm was evaluated using real data taken from the registration department of Al Al-Bayt University related to the courses registered for by students who graduated from the CS department. These courses were studied by students during their four-year study period. In each study year, the student registers for two or three semesters; the first, second, and summer semesters. Quite often, the students register for four to six courses in the first and second semesters and up to three courses in the summer semester. This means that actually the students need to register for up to 11 semesters to be able to graduate within four years, based on that the last study year consists of only two semesters with no summer semester.

There exists a plan guide, but the students do not follow it. They usually prefer to register courses in a sequence that differs from that set in the course plan following the behavior of previous students and based on the difficulties they sometimes face in registering the due courses. The prerequisite courses also affect the decision taken by the student on the courses to register. In consequence, this research was intended to find the best sequence of courses (plan guide) to register based on the courses which the previous students actually registered. The best sequence (optimal path) will be found by implementing ACO model on data taken from the registration records of previous CS students who already graduated.

To conduct ant colony optimization, a software module specially-tailored for the data and need of the present study was built. This module generates paths (plan guide) depending on the course sequence patterns of the courses registered actually by students. Using the Ant Colony methodology, an optimal path for course sequence will be found. This path represents the actual guide for course registration that depends on student student's actual course registration behavior [1], [2].

II. LITERATURE REVIEW

Ant Colony Optimization (ACO) is used to solve combinational problems such as vehicle routing [3], [4], salesman's travel problem [5], [6], graph coloring [7], the quadratic assignment problem [8], and process planning and scheduling [9].

Zuo et al. [10] investigated task-scheduling problems in cloud computing and proposed a resource cost model that defines the demand of tasks on resources that depend on a multi-objective optimization method. Simulation experiments were designed to evaluate performance of their proposed method [10].

In 2003, Dr. Dorigo won the Marie Curie excellence award for inventing the Ant Colony algorithm. This algorithm is considered as a population-based algorithm and classified within the field of swarm intelligence. The course time tabling problem was researched by Socha et al. [11]. In [12], the Ant Colony algorithm was applied on 11 courses to produce optimal timetable sets. Behavior of the real ant corresponds to communication via sharing and distributing information according to a biological method known as the pheromone trails. Many combinatorial optimization problems have been solved successfully by implementing ACO [13]-[15]. Real

ants search for food. When the first ant finds food it returns back to the nest (colony), leaving a pheromone trail. Other ants will follow this pheromone trail to reach the food source. However, the pheromone evaporates with time. Therefore, the long paths that are used by few ants will have less pheromone concentration due to evaporation of the pheromone. On the other hand, more ants use the short paths, thus resulting in increment of the pheromone levels of these paths [16]. Normally, most of the ants follow the path that has the higher pheromone concentration. Only few ants do not usually follow that path. These ants are the ones which often find other possible good paths. In the case of appearance of any obstacle or interruption in the main path that is followed by most ants, the other paths that were detected by the fewer ants become a useful solution. This frequently happens if the path is interrupted by an obstacle or if the food runs out.

Use of the pheromone by ants represents a way of indirect coordination and communication between the colony members. This method of communication is separated in time. Hence, one member of the community modifies the same parameters in the environment and the others see this modification later and use this information to take suitable decision, which is a form of self-organization in a large community that produces intelligent complex structure without need for direct communication or centralized decision and control.

In 2009, Jardat proposed a hybrid algorithm using an Ant Colony system with simulated annealing to solve the problem of university course timetable, which is regarded as a complex organization problem for which it is hard to find an optimal solution [17].

Socha et al. (2002) developed a min-max ant system algorithm for the university course time tabling problem. They used data sets for 11 course time tabling tests and compared their results with those of a random restart local search algorithm. The comparison results uncovered that the proposed algorithm was able to give better results than the random restart local search algorithm and to produce viable solutions [12].

III. METHODOLOGY

In this study, data were collected and cleaned. Then, an ACO algorithm was applied to the cleaned data in order to extract an optimal path for a course plan guide.

A. Data Collection

Data were collected from the Registration and Admission Department in Al Al-Bayt University. The collected registration data pertain to 200 students formerly admitted to the CS department in the academic years 2009/2010 and 2010/2011 and who already graduated from the university. These data represent the sequence of the courses registered for by each student. This sequence consists of the registered courses and the semester and year in which each course was registered. A sample of the collected data is show in Table I.

TABLE I. SAMPLE OF THE RAW DATA

Student number	Academic Year	Semester	Course Code	Course Name
1000901074	2009	2	901210	Object Oriented Programming
1000901074	2009	2	901131	Computer Skills
1000901074	2009	3	901200	Discrete Math
1050901044	2010	1	901220	Logic Design
1050901044	2010	1	901099	Computer 1
1050901044	2010	1	901200	Discrete math
1050901044	2010	1	902230	Info. Systems
1050901044	2010	1	901131	Computer Skills
1050901031	2012	2	901325	Networks
1050901031	2012	2	901240	Data Structures
1050901031	2012	2	901211	Java Lab
1050901031	2012	2	901300	Computation
1050901031	2012	2	901320	Architecture

B. Data Cleaning

In order to prepare the data for processing, there was a need for these data to be cleaned by removing the compulsory courses and grouping the different elective courses under a unique course code (i.e., 'elective') so as to treat the various elective courses in the same way how the compulsory courses are treated [18]. Furthermore, if a student re-registered a course due to inability to pass it from the first time, then we only considered the first registration for the course and removed the repetitive registrations, if any.

C. Ant Colony Algorithm

A software module was designed for performing the ACO in order to generate the optimal path (sequence plan guide). This module has n nodes, each representing a course in the guide plan. Theoretically, this module should be able to find a path that goes through all nodes, which is due to be generated based on information taken from the students' course registration sequences. However, since there are 200 students, then the maximum number of possible paths is only 200, but practically this will not be the case since there will be some impossible paths for certain sequences due to the constraint of the prerequisite course requirement. According to the ACO algorithm, the particular path generated will depend on different paths.

The ACO uses the concept of pheromone evaporation with time. Thus, the less visited nodes or edges will contain lower amount of pheromone than the nodes and edges which are more visited. On this account, the student represents the ant agent and the sequence of courses taken by a student will draw the path from beginning to end and, in consequence, increment the amount of pheromone in that path.

The path consists of a set of edges that connect the nodes (courses). Each edge has a parameter representing the amount of pheromone at that edge. The pheromone evaporates at a constant rate. Thus, if an edge is not visited for a long time, then the pheromone level at it will be zero. If the evaporation rate is set too high, then the pheromone rate at all edges will rapidly reach to zero, which will not help in finding a path. On the other hand, if the evaporation rate is too low, then the pheromone will be saturated at the edges, which too will not help in finding a path. Experimentally, we will find the suitable pheromone evaporation rate in order to identify an optimal path. The ant system algorithm that was designed in the present study to optimize the course plan guide problem is presented next.

Owing to that the number of possible routes is large because of the large number of courses (31 course), which will lead to 31x31 possible edges, and as an improvement over the algorithm, this study did not use a connection matrix to represent this model. Instead, an ad-hoc connection was constructed so as to reduce the time complexity. This was achieved by preparing a dynamic list that contains the visited edges. By so doing, the model only modifies the visited edges.

Algorithm: An Ant System for Optimizing the Course Plan Guide Problem

initialization:

initialize : Q : amount of pheromone increased when an edge is visited

initialize : p : evaporation rate.

initialize : m = 100 students.

initialize : c = 27 number of courses.

initialize : S = 11 number of semesters.

initialize : MSC matrix, that consists the pheromone value for every course in each semesters.

for k=1 to m

for g=1 to c

 get course from the input data for each student

 update pheromone trails on the specified semester-course in the MCS

for i=1 to S

 for j=1 to c

 apply evaporation on all edges

 find the path with the highest pheromone rate

D. Results and Discussion

The raw data were drawn from the Admission and Registration Department of Al Al-Bayt University. They pertain to students who already graduated. Those students were admitted to the university in the academic years 2009/2010 and 2010/2011. The data consist of records that include the student number, course number, course name, and the semester and year in which the student registered each course (Table I).

The collected data first passed through a preprocessing stage comprising several steps of data cleaning. These steps included elimination of some courses that were unrelated, and were, therefore, out of the study scope such as the university required courses. Moreover, cleaning included elimination of the courses that were re-registered by the students who failed to pass them from the first time.

Afterwards, the ACO model was applied by running a module that was specially tailored and customized to fit the research data. Structurally, this module consists of a set of nodes, each representing one course. Overall, the number of nodes in this study was 31, corresponding to 31 courses. Every student must register for the 31 courses during her/his study period, which is normally four academic years, each consisting of a maximum of three semesters. This means that each student can register a maximum of 11 semesters until graduation, assuming that only two semesters are registered for in the last year. The 11 semesters represent stages in our model. Hence, the final results will correspond to the 31 courses distributed among 11 stages (semesters) and the optimal extracted path shown in Fig. 1. In this figure, S_i ($i = 1, 2, 3, \dots, 11$) represents the semester number.

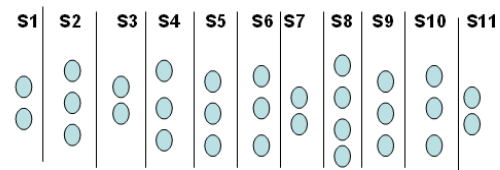


Fig. 1. Sample plan guide.

The experiment was run 10 times for different pheromone evaporation rates, ranging from 0.1 to 1 at a step of 0.1. A table of distribution of the courses among semesters was constructed for each run. Then, the information in those tables was normalized in order to compare the results between the tables and validate them. The paths were then extracted from these tables on the basis of the pheromone rate as shown in Table II.

It was found that some courses were located in the same semesters regardless of the pheromone evaporation rate. On the other hand, allocation of some other courses to semesters varied from semester to another, depending on the pheromone evaporation rate (Table II).

As Table II reveals, fourteen courses were not affected by change in the pheromone rate. These courses are the courses listed in Table II with the numbers 1, 2, 5, 6, 11, 12, 13, 17, 19, 23, 25, 27, 28, and 29.

On the other hand, sixteen courses moved between semesters with the change in the pheromone evaporation rate. These courses are the courses appearing in Table II with the numbers 3, 4, 7, 8, 9, 10, 14, 15, 16, 18, 21, 22, 24, 26, 30, and 31.

TABLE II. DISTRIBUTION OF COURSES AMONG SEMESTERS FOR DIFFERENT PHEROMONE EVAPORATION RATES

		Pheromone Evaporation Rate									
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Course Number	1	2	2	2	2	2	2	2	2	2	2
	2	4	4	4	4	4	4	4	4	4	4
	3	5	5	5	7	7	7	7	7	7	7
	4	5	5	5	7	7	7	7	7	7	7
	5	4	4	4	4	4	4	4	4	4	4
	6	1	1	1	1	1	1	1	1	1	1
	7	7	7	7	8	8	8	8	8	8	8
	8	7	7	7	8	8	8	8	8	8	8
	9	8	8	8	3	3	3	3	3	3	3
	10	5	7	7	7	7	7	7	7	7	7
	11	4	4	4	4	4	4	4	4	4	4
	12	7	7	7	7	7	7	7	7	7	7
	13	4	4	4	4	4	4	4	4	4	4
	14	8	8	8	8	10	10	10	10	10	10
	15	5	5	5	8	8	8	8	8	8	8
	16	7	5	5	5	5	5	5	5	5	5
	17	7	7	7	7	7	7	7	7	7	7
	18	6	8	8	8	8	8	8	8	8	8
	19	10	10	10	10	10	10	10	10	10	10
	20	9	10	10	9	9	9	9	9	9	9
	21	8	8	8	8	11	11	11	11	11	11
	22	8	8	8	8	8	9	9	9	9	9
	23	9	9	9	9	9	9	9	9	9	9
	24	7	10	10	10	10	10	10	10	10	10
	25	10	10	10	10	10	10	10	10	10	10
	26	10	10	10	10	8	8	8	8	8	8
	27	10	10	10	10	10	10	10	10	10	10
	28	11	11	11	11	11	11	11	11	11	11
	29	3	3	3	3	3	3	3	3	3	3
	30	11	11	11	8	8	8	8	8	8	8
	31	10	10	10	10	11	11	11	11	11	11

Course number 20 moved twice from semester 9 to semester 10 at the pheromone evaporation rate of 0.2. It then returned back to semester 9 at the pheromone evaporation rate of 0.4. The numbers and names of all courses are shown in Table III.

After analyzing the data generated by the built model, it was found that there is a stable guide plan for the pheromone evaporation rates ≤ 0.3 (Table IV). On the other hand, when increasing the pheromone evaporation rate above 0.3 the results become unstable and unaccepted logically based on the researchers' knowledge of the sequence of registration of courses.

TABLE III. NUMBERS AND NAMES OF COURSES

Index	Course Number	Course Name	Index	Course Number	Course Name
1	901131	Computer Skills IT	17	901327	Wireless Networks
2	901210	OOP	18	901331	DBMS
3	901211	Java	19	901332	Operating Systems
4	901213	Java Lab	20	901340	Algorithm
5	901214	Object Oriented Lab	21	901341	Software Engineering
6	902230	Information Systems	22	902410	Fourth Generation Languages
7	902350	Web Design	23	901430	Distributed Systems
8	902354	Web Design Lab	24	902444	System Analysis
9	902450	E-Commerce	25	901470	Artificial Inelegance
10	902201	Numerical Analysis	26	901480	Data Security
11	901220	Digital Logic Design	27	901499	Graduation Project
12	901240	Data Structures	28	901500	Training
13	901300	Computation Theory	29	901200	Discrete Math
14	901310	Visual Programming	30	901351	Human Computer Interaction
15	901320	Computer Architecture	31	902481	Special Topics
16	901325	Computer Networks			

TABLE IV. DISTRIBUTION OF COURSES AMONG SEMESTERS FOR PHEROMONE RATES ≤ 0.3 AND > 0.3

Course number	Semester Number with pheromone evaporation ≤ 0.3	Semester Number with pheromone evaporation > 0.3	Course number	Semester Number with pheromone evaporation ≤ 0.3	Semester Number with pheromone evaporation > 0.3
1	2	2	17	7	7
2	4	4	18	8	8
3	5	7	19	10	10
4	5	7	20	10	9
5	4	4	21	8	11
6	1	1	22	8	8
7	7	8	23	9	9
8	7	8	24	10	10
9	8	3	25	10	10
10	7	7	26	10	8
11	4	4	27	10	10
12	7	7	28	11	11
13	4	4	29	3	3
14	8	10	30	11	8
15	5	8	31	11	11
16	5	5			

As can be seen in Table IV, for 11 courses the registration semester differed by increasing the evaporation rate beyond 0.3. Meantime, for the remaining 20 courses the registration semester was not affected by increasing the evaporation rate. In view of these findings, the pheromone evaporation rate of 0.3 was selected as the optimum pheromone evaporation rate for generation of the course plan guide.

TABLE V. DISTRIBUTION OF COURSES AMONG SEMESTERS AS GENERATED BY THE PROPOSED ALGORITHM

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
6	1	29	2	3		7	9	23	19	28
			5	4		8	14		20	30
			11	15		10	18		24	31
			13	16		12	21		25	
						17	22		26	
									27	

TABLE VI. ACTUAL DISTRIBUTION OF COURSES IN SEMESTERS

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
1	2		6	3		7	9		22	23
	5		10	4		8	17		25	26
	29		11	13		14	19		27	28
			12	15		16	20		31	30
						18	21			
						24				

A comparison of the results generated by this study (Table V) and the existing plan guide (Table VI) indicates that for 14 courses the registration semesters matched while 17 courses were registered in different semesters.

By Taking the ‘Computer Skills for IT Students’ course as an example to explain the observed differences between the plan guides in Tables V and VI, it can be said that this course is a programming course that is usually registered in the second semester since most of the students prefer to register fundamental courses (e.g., Information Systems) in the first semester in addition to university requirement courses, even though the ‘Computer Skills for IT Students’ course is supposed to be registered in the first semester as shown in Table VI.

E. Conclusion

This study developed a course plan guide based on real data on courses registered by CS students in Al Al-Bayt University over their four-year study period. An Ant Colony algorithm was used to monitor the actual sequence of the courses registered by the students. The developed plan guide explains why some course sections were cancelled; the numbers of students registering in them were not high enough. Comparison in the course registration sequence between the present and the proposed plan guides uncovered match between the two plans in the order of registration for 14 courses and differences for 17 courses. In future work, the size of data can be increased and the proposed algorithm can be applied to the course schedule so as to monitor its performance.

ACKNOWLEDGMENTS

The authors express their sincere thanks to Al Al-Bayt University – Jordan for cooperation in providing them with the necessary data and facilities.

REFERENCES

- [1] Clemens Nothegger, Alfred Mayer, Andreas Chwatal and Günther R. Raidl, “Solving the post enrolment course timetabling problem by ant colony optimization.” *Annals of Operations Research* April 2012, Volume 194, Issue 1, pp 325-339.
- [2] C.W. Leung, T.N. Wong, K.L. Mak and R.Y.K. Fung. “Integrated process planning and scheduling by an agent-based ant colony optimization”. *Computers & Industrial Engineering* 59 (2010) 166–180.
- [3] John E. Bella and Patrick R. McMullenb. Ant colony optimization techniques for the vehicle routing problem.” *Advanced Engineering Informatics* 18 (2004) pp.41–48.
- [4] Dongming Zhao, Liang Luo and Kai Zhang. “An improved ant colony optimization for the communication network routing problem.” *Mathematical and Computer Modelling* 52 (2010) pp. 1976–1981.
- [5] Marco Dorigo and Luca Maria Gambardella. “Ant colony system: a cooperative learning approach to the traveling salesman problem.” *IEEE Transactions on Evolutionary Computation*, VOL. 1, NO. 1, APRIL 1997, pp 53 – 66.
- [6] Marco Dorigo, Luca Maria Gambardella. Ant colonies for the travelling salesman problem. *Biosystems* Volume 43, Issue 2, July 1997, Pages 73–81.
- [7] Thang N. Bui, ThanhVu H. Nguyen, Chirag M. Patel and Kim-Anh T. Phan. An ant-based algorithm for coloring graphs. *Discrete Applied Mathematics*, Volume 156, Issue 2, 15 January 2008, Pages 190–200.
- [8] L. M. Gambardella; E. D. Taillard and M. Dorigo. Ant Colonies for the Quadratic Assignment Problem. *The Journal of the Operational Research Society*, Vol. 50, No. 2 (Feb., 1999), pp. 167-176.
- [9] S. Zhang, and T. N. Wong, Integrated process planning and scheduling: an enhanced ant colony optimization heuristic with parameter tuning. *Journal of Intelligent Manufacturing* December 2014.
- [10] L. Zuo, L. Shu, S. Dong, C. Zhu, T. Hara, “A Multi-Objective Optimization Scheduling Method Based on the Ant Colony Algorithm in Cloud Computing,” *IEEE Access*, vol. 3, pp. 2687-2699, 2015.
- [11] Krzysztof Socha, Michael Sampels and Max Manfrin. Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art. *Proceedings of the 2003 international conference on Applications of evolutionary computing*, Pages 334-345 Springer-Verlag Berlin, Heidelberg.
- [12] Krzysztof Socha, Joshua Knowles and Michael Sampels. A MAX-MIN Ant System for the University Course Timetabling Problem, *Third International Workshop, ANTS 2002 Brussels, Belgium, September 12–14, 2002 Proceedings*, Pages pp 1-13.
- [13] E. Bonabeau, M. Dorigo, and G. Theraulaz, “From Natural to Artificial Swarm Intelligence,” Oxford University Press, Oxford, 1999.
- [14] M. Dorigo, V. Maniezzo, and A. Coloni, “The Ant system: optimization by a colony of cooperating Agents.” *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(1):29-41, 1996.
- [15] M. Dorigo and B. Christian. *Ant Colony Optimization: A Survey*. Elsevier B.V., *Theoretical Computer Science* 344 (2005) 243 – 278. 2005.
- [16] Anuj K. Gupta. Computation of Pheromone Values in AntNet Algorithm. *International Journal of Computer Network and Information Security*, 2012, 9, pp. 47-54.
- [17] Masri Ayob, Ghaith Jaradat. Hybrid Ant Colony Systems for Course Timetabling Problems. *2nd Conference on Data Mining and Optimization*, Selangor, Malaysia, IEEE, pp. 120-126 ,2009.
- [18] Ajith Abraham and Vitorino Ramos. Web Usage Mining Using Artificial Ant Colony Clustering and Genetic Programming. *The 2003 Congress on Evolutionary Computation*, 2003.