Experimental Design and Analysis on Parameter Investigation and Performance Comparison of Ant Algorithms for Course Timetabling Problem

T. Lutuksin, A. Chainual and P. Pongcharoen*

Department of Industrial Engineering, Faculty of Engineering, Naresuan University, Phitsanulok, 65000
**Email: pupongp@yahoo.com, pupongp@nu.ac.th*

Abstract : Course timetabling usually arises every academic year and is solved by academic staff with/without course timetabling tool. The desirable timetable must be satisfied by hard constraints whilst soft constraints are not absolutely essential. Course timetabling is known to be NP-hard problem, which means that the computational time required to find the solution increases exponentially with problem size. Automated timetabling tool has been developed for university courses scheduling. In this work, two variants of Ant Colony Optimisation algorithms called Max-Min Ant System and Ant Colony System were applied to solve university course timetabling problem. A two-step sequential experiment was sensibly designed and carried out using six benchmarking course timetabling problems. The analysis of the obtained results suggested that each method performed best on one another based on its parameter configuration.

Keywords : Experimental Design and Analysis, Course Timetabling, Ant Colony Optimisation.

1. INTRODUCTION

There have been comprehensive literature reviews on automated procedures for constructing efficient and desired timetables. The approaches adopted can be classified into four categories [1]: i) sequential methods, such as graph colouring [2]; ii) clustering methods [3]; iii) constraint-based methods e.g. integer programming [4]; and iv) metaheuristics methods [5] such as Genetic Algorithms (GA) [6], Simulated Annealing (SA) [7], Taboo Search (TS) [8], Neural Network (NN) [9], Artificial Immune System (AIS) [10, 11] and Ant Colony Optimisation (ACO) [12, 13]. In addition, new approaches including case-based reasoning approach [14], fuzzy methodology [15] and hyper-heuristics approach [16] have also been used to solve timetabling problems.

Ant Colony Optimisation has several advantages and has therefore received high attention in the last few decades. It performs multiple directional searches using a set of candidate solutions (ants) while most conventional optimisation methods conduct single directional search. ACO uses heuristic information related to the problem considered as part of the route construction process. Some ACO variants e.g. Ant Colony System (ACS) uses candidate list during the search process for composing its tour. Furthermore, stochastic transition rule based on the pheromone trial depositing by ants has been used to guide the ant exploring the solution space. Due to the approximation process for searching the optimal solution, the method usually performs faster than conventional optimisation methods when solving a huge problem but the optimal solution can not be guaranteed.

Although the advantages of ACO have previously been described, the applications of ACO to solve university course timetabling were rarely reported. For example, Socha et al. [12] have applied Max-Min Ant System (MMAS) to solve course timetabling problems, in which good solutions were obtained even for the large problem. Azimi [13] has also applied Ant Colony System (ACS) for solving examination timetabling problems. The ACS performance has been compared with other metaheuristics including GA, SA and TS. The research has shown that the solutions obtained from ACS were better than those using other techniques. An another example, Lutuksin et al. [17] have developed the automated timetabling tool called ANCOTT by applying Ant Colony Optimisation (ACO) to solve a course timetabling problem. However, there have been only three hard and soft constraints considered. The research has concluded that the timetables obtained from ACS satisfied all hard constraints and lowest violation of soft constraints both in terms of the best so far and its average. However, the parameters' values used in the previous work have been assigned in an ad hoc fashion without statistical investigation.

Statistical tools called experimental design and analysis have been widely used for determining the influenced factors and investigating the appropriate parameter setting for the process or system [18]. The tools have therefore been applied for investigating the optimised parameter configuration of the metaheuristics such as GA [19-21], AIS [22], SA [23] and Ant System

(AS) [24].

The objectives of this paper were to demonstrate the use of advance statistical design and analysis for investigating the appropriate parameters' setting of Ant Colony Optimisation including Max-Min Ant System (MMAS) and Ant Colony System (ACS) and to compare the performance of the MMAS and ACS both in terms of the quality of the results obtained and the computational time required. The next section in this paper presents the definition of the course timetabling problems. Section 3 explains the Ant Colony Optimisation followed by the architectural design of the Ant Colony based Timetabling Tool (ANCOTT). Section 5 presents the experimental design and analysis followed by conclusions.

2. COURSE TIMETABLING PROBLEM

There are many types of timetabling problems such as employee timetabling [25], sports timetabling [26], transportation timetabling [27] and educational timetabling problem [12, 28]. For educational institutions e.g. high school, college or university, timetabling problem periodically arises every term and is solved manually by academic staff or automatically generated using software packages. Timetabling problem is known to be NP-hard problem [4], which means that the amount of computational time required to find the solution increases exponentially with problem size [29]. Many factors such as the number of academic staff, students, curriculum and institutional facilities are related to the complexity and the size of timetabling problem.

Generally, educational timetabling problem is a set of events (courses or exams) that must be appropriately assigned into a certain number of timeslots (time periods) and subject to hard and soft constraints [28]. A feasible timetable must satisfy all hard constraints in order to prevent clashes while soft constraints are not absolutely essential and should be minimised. Educational timetabling problem can be classified as examination and course timetabling including lectures, seminars, tutorials and laboratories. The similarity of both timetabling is that, for example, two events (courses or exams) cannot be scheduled in the same classroom at the same periods [30]. The difference is that compact course timetable is usually preferred, whereas spread examination timetable is generally required by students.

Course timetabling may be defined as the process of assigning courses and its corresponding lecturers to specific time periods throughout the working days and to specific classrooms suitable for the number of students registered and the needs of each course. In most educational institutions, the number of timeslots is equally split. Each room has a capacity of available seats. Finally, a curriculum is a group of courses based on curricula and is set according to the curricula published by the university. A feasible timetable is a set of all lectures of each course, properly assigned within a given number of rooms and time periods (timeslots) and satisfied by all hard constraints. In this work, the following hard and soft constraints were considered.

Hard constraints

1. Lectures (Hc_1): All lectures of a course must be scheduled and assigned to distinct periods.
2. Room occupancy (Hc_2): Two or more lectures cannot take place in the same room at the same period.
3. Conflicts (Hc_3): Lectures of courses in the same curriculum or taught by the same teacher must be all scheduled in different periods.
4. Unavailable constraints (Hc_4): If a teacher of a course is not available to give lecture of the course at a given period, then no lecture of the course can be scheduled on the period.

Soft constraints

1. Room capacity (Sc_1): For each lecture, the number of students attending the course must be less or equal than the number of seats of all the rooms hosting the lectures.
2. Minimum working days (Sc_2): The lectures of each course must be spread into a minimum number of days.
3. Curriculum compactness (Sc_3): Lectures belonging to a curriculum should be adjacent to each other (i.e., in consecutive periods).
4. Room stability (Sc_4): Each lecture of a course should be given in the same room.

The design task was to generate timetables for lecturers, students and classrooms that satisfy all hard constraints and avoid the violations of soft constraints [31]. A good solution must not violate hard constraints while soft constraints violation should be zero or closed to zero if possible. Total violation index (TVI) of a timetable can be determined by the equation (1).

$$\text{Minimise } TVI = w_1Sc_1 + w_2Sc_2 + w_3Sc_3 + w_4Sc_4 \qquad (1)$$

Where w_1 - w_4 are the weights corresponding to the amount of violations of four soft constraints. The weight setting usually guides the search process to avoid a violation of soft constraint. If there is no priority between soft constraints, the weight setting can be equally specified. Otherwise, the high priority constraint may be more weighted. In this work, the weights (w_1 - w_4) are specified to 1, 5, 2 and 1, respectively.

3. ANT COLONY OPTIMISATION (ACO)

Max-Min Ant System (MMAS) [32] and Ant Colony System (ACS) [33] are variants of Ant Colony Optimisation (ACO), which was inspired by the foraging behaviour of real ants to find the shortest path between the source foods to their nest [34-36]. The methods have been successfully applied for solving various combinatorial optimisation problems e.g. travelling salesman [37], scheduling [38] and assignment problems [39]. The basic concept of ACO is the use of a probabilistic solution construction mechanism based on stigmergy. Two main phases of the ACO metaheuristic feature are the solution construction (where the travelling routes or tours are constructed by the ants) and the pheromone update, which includes pheromone evaporation and pheromone deposition when the ants completed their tours. The heuristic information and the pheromone values are uniquely used to determine the probabilities of moving decisions.

The behaviour of real ants to find the shortest path can be shown in Figure 1. At time = 0, ants move from node A to node C via node B or D equally based on the probability of selection, in which the heuristic information (distance) is considered more than pheromone information. During periods of the journey (time = 5), ants deposit some pheromone on their trail when they complete their tours (back from node C to A). At the same time, pheromones on the trails are also evaporated automatically. Finally, at time = 10, ants obviously move on the shorter A-B-C path having more amount of pheromones than A-D-C path. In this stage, pheromone information is considered more than heuristic information.

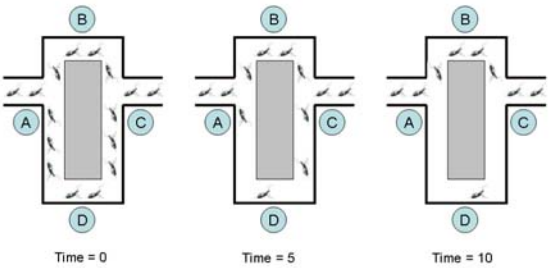

Fig. 1. The moving behaviour of ants [33].

There are some differences between MMAS with ACS. ACS performs the accumulated search experience called pseudorandom proportional rule which is used to determine the probability of ants moving while MMAS uses random proportional rule. ACS conducts both local and global pheromone update procedure but MMAS only uses global pheromone update procedure. Finally, only MMAS specifies boundary of pheromone level (max-min) on the trails [34]. The pseudo code of the ACO procedures is shown in Figure 2.

```
ACO procedures
input problem data
    parameters' setting and initialisation
    sort courses (C) according to the priority of events and student sizes
    while iteration ≤ max iteration do
        for k = 1 to m do
            create ant k (where k = 1, 2, …, m)
            set timetable k is empty
            for c = 1 to C do
                choose timeslot (t) using pseudorandom/random proportional rules
                if ACS then update local pheromone
            end for
            withdraw ants
        end for
        record best solution
        update global pheromone based on the type of ACO methods
        if MMAS then update pheromone level (max-min) on the trails
    end while
output the best solution
```

Fig. 2. Pseudo code of the Ant Colony Optimisation.

4. ANT COLONY BASED TIMETABLING TOOL

The Ant Colony based Timetabling Tool (ANCOTT) program has been coded in modular style using a general purpose programming language called TCL/TK [40]. The development of the ANCOTT was based on three principles; program structure (see Figure 2), model design (see Figure 3) and user interface (see Figure 4). Figure 3 shows the architectural design of the ANCOTT system, which is mainly categorised into three phases; input, timetabling and output phases. In the first phase, the input data including classrooms, students, courses, teachers and timetabling constraints must be specified by scheduler via graphical user interface (GUI) provided. All data is then encoded in the second phase where course scheduling is performed using Ant Colony Optimisation methods, in which its parameters must also be assigned via the GUI. The evaluation process of the timetables generated is based on the desired hard and soft constraints. In the final phase, the feasible timetables for each teacher, student and room are reported.

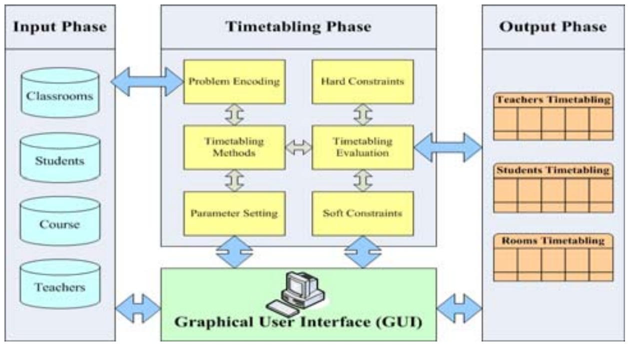

Fig. 3. Structural architecture of the system [17].

From Figure 4, it can be seen that the GUI provided within the ANCOTT helps user to easily assign the values of each parameter, random seed number, enable and disable hard and soft constraints considered, weight of the soft constraints and priorities of timetabling process.

5. EXPERIMENTAL DESIGN AND ANALYSIS

This section is aimed to present computational experiments conducted using MMAS and ACS algorithms to solve benchmarking instant problems (shown in Table 1) provided in the international timetabling competition [30] organised by the timetabling research groups from various universities. The objective of the competition was to share the knowledge between researchers and practitioners on how to use the techniques or methodology for solving timetabling problems. In this work, the timetables generated by ANCOTT program were measured by counting the number of violations on the soft constraints mentioned in section 2. Personal computers with Core2 Quad 2.66 GHz CPU and 4 GB RAM was used to determine the simulation time required to execute a computational run.

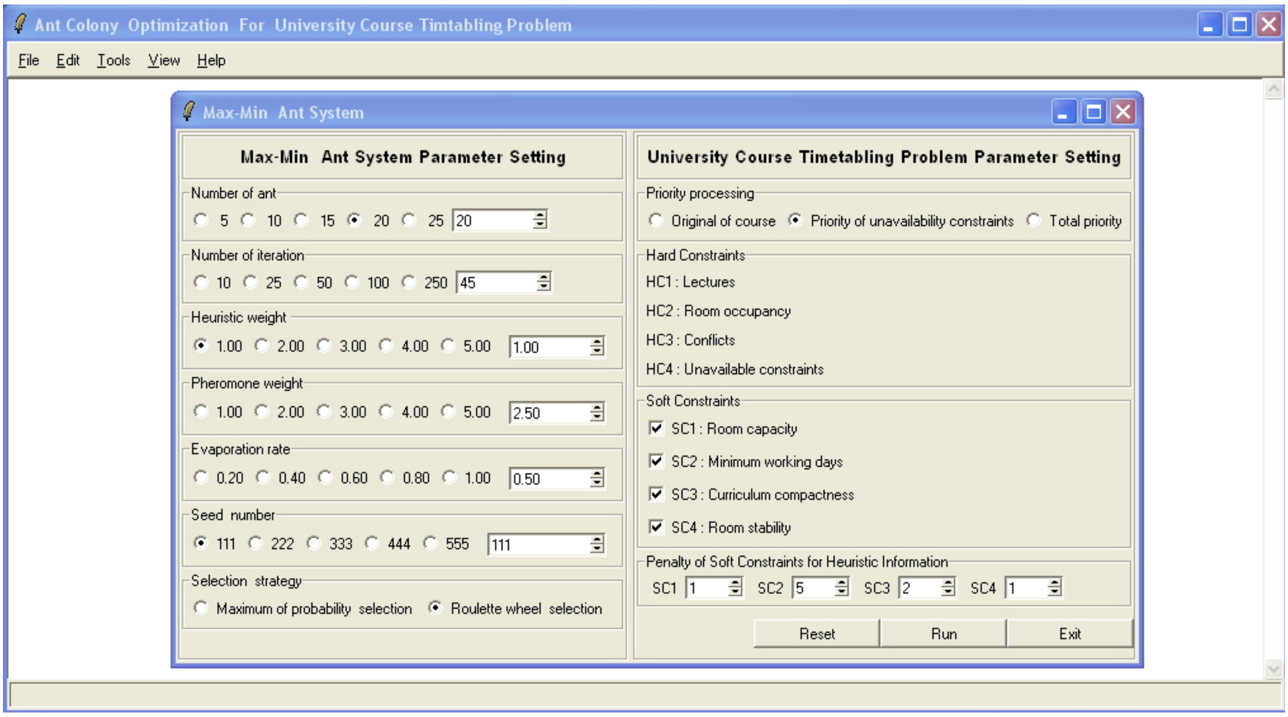

Fig. 4. Snapshots of the graphical user interface provided within the ANCOTT.

Table 1 Characteristics of benchmarking instant problems considered in this work.

Problem number	Characteristics of course timetabling problems						
	Courses	Classrooms	Days/week	Periods/day	Teachers	Curricula	Unavailable constrains
1	30	6	5	6	24	14	53
2	30	5	5	9	24	13	94
3	54	9	6	6	47	139	771
4	72	16	5	5	61	68	382
5	79	18	5	5	70	57	396
6	85	17	5	5	68	60	486

5.1 Experiment 1

The first experiment was aimed to demonstrate the use of advance statistical design and analysis for investigating the appropriate parameter setting of both MMAS and ACS algorithms for solving the benchmarking problems. Four ACO's parameters including number of ants multiply number of iterations ($A*I$), pheromone weight (α), heuristic information weight (β) and pheromone evaporation rate (ρ) and its level considered are shown in Table 2. The values of ACO's parameters were adopted from previous research [41]. Due to the number of parameters and its level, there will be a difficulty on the amount of experimental work when adopting conventional statistical design.

One-third fractional factorial (3^{4-1}_{IV}) experimental design [18] was adopted in this experiment for decreasing the number of computational runs by 66.67%. The computational experiment was based on the first instant problem and repeated five times using different random seed numbers. The computational results obtained from 135 (3^3*5) runs were analysed using a general linear model form of analysis of variance (ANOVA). Table 3 shows ANOVA table consisting of Source of Variation, Degrees of Freedom (*DF*), *F* and *P* values. A factor with value of $P \leq 0.05$ was considered statistically significant with 95% confidence interval.

From Table 3, it can be seen that all main factors

Table 2 Experimental factors and its levels.

Factors	Levels	Coded Values		
		Low (-1)	Medium (0)	High (+1)
$A*I$	3	20*45	30*30	45*20
α	3	0.01	0.50	0.99
β	3	0.00	2.50	5.00
ρ	3	0.01	0.50	0.99

of both MMAS and ACS algorithms except the combination of ants and iterations ($A*I$) were statistically significant with 95% confidence interval. Besides, the most influence factor for this experiment was heuristic information (β) factor because of the extremely F value and the next influence factors were pheromone weight (α), pheromone evaporation rate (ρ), and ants and iterations ($A*I$), respectively. The main effect plots of MMAS shown in Figure 5 suggested that those factors including $A*I$, α, β, and ρ should be at 20*45 or 30*30, 0.99, 2.50 or 5.00 and 0.50 or 0.99, respectively. Whilst the main effect plots of the ACS parameters shown in Figure 6 suggested that those factors should be at 30*30 or 45*20, 0.01, 2.50 or 5.00 and 0.01, respectively. It should be noted that two-interaction term cannot be considered in the ANOVA table due to a lack of degrees of freedom obtained from the proposed design.

Table 3 ANOVA on the ACO's parameters.

Source of Variation	DF	MMAS		ACS	
		F	***P***	***F***	***P***
$A*I$	2	2.86	0.061	2.09	0.128
α	2	35.89	0.000	8.13	0.000
β	2	4027.1	0.000	1371.6	0.000
ρ	2	9.49	0.000	18.86	0.000
Error	126				
Total	134				

A sequential sub-experiment was conducted to verify the appropriate parameter setting identified previously for both algorithms by comparing the average of the best so far solutions obtained with those using randomly combined parameter setting based on five replications as show in Table 4.

It can be seen that the averages of the total violations associated with the timetables obtained from the appropriate parameter setting identified previously were dramatically lower than those results with random parameter settings. This demonstration supported that the performance of the metaheuristics depends on its mechanism and parameter setting. The performance can be improved up to 25 percents using the appropriate parameter configuration.

Table 4 Results obtained from different settings.

Methods	Parameters' settings		Improve (%)
	Identified from experiment	Random combination	
MMAS	72	97	25.77
ACS	110.8	147.5	24.88

5.2 Experiment 2

This experiment was designed to compare the performance of MMAS and ACS for solving six benchmarking problems detailed in Table 1. The parameter settings for both methods were adopted from the previous experiment. The experimental results were statistically analysed in terms of the minimum, maximum, average, standard deviation and the computational time required of the best so far solutions (timetables) obtained from the problems, each of which was conducted with five replications using different random seed numbers. The analysis on the experimental results was summarised in Table 5. According to the minimum, maximum and mean values, it can be seen that the MMAS produced the timetables with lower constraint violations than those using the ACS for the problem number 1 and 2, which are relatively smaller than the remaining problems. Nevertheless, the ACS performed better than the MMAS for the larger problems. The average computational times required by both algorithms were marginally different on all benchmarking problems.

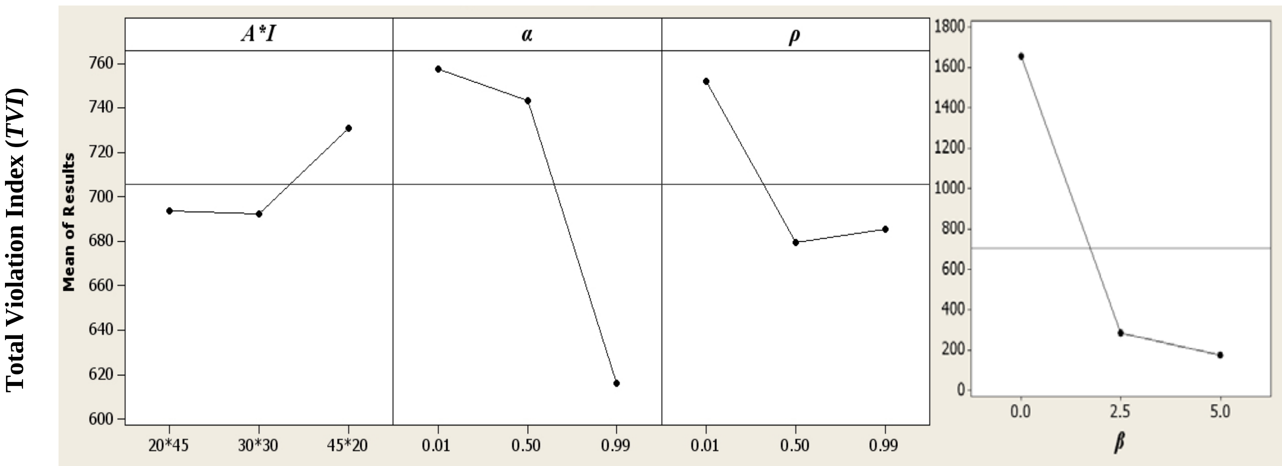

Fig. 5. Main effect plots of MMAS algorithm.

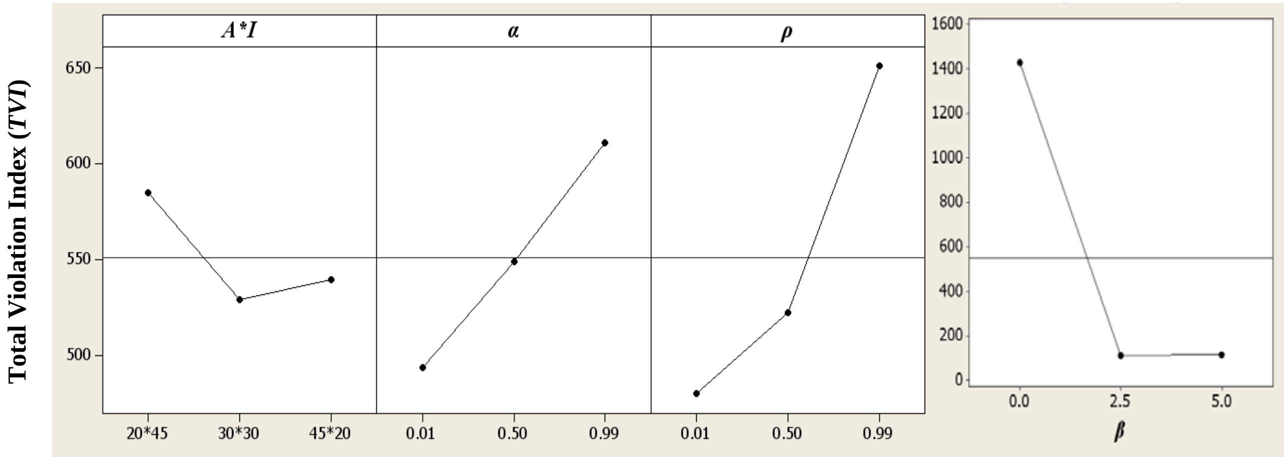

Fig. 6. Main effect plots of ACS algorithm.

6. CONCLUSIONS

Ant Colony based Timetabling Tool has been developed to solve university course timetabling problem. This paper demonstrated the use of the experimental design and analysis tools for investigating the appropriate parameters setting before sequentially study the performance of the MMAS and ACS algorithms. The analysis on experimental results indicated that all parameters except the combination of ants and iterations were statistically significant with 95% confidence interval. The most influence factor was the heuristic information (β), which should be set between 2.5-5, followed by the pheromone weight (α) and the pheromone evaporation rate (ρ). However, the appropriate parameter setting could be varied between datasets due to the nature of the problem domains and its complexity. It was also found that the results can be improved up to 25 percents having use of appropriate parameter setting. The sequential experiment indicated that the quality of timetables produced by MMAS is better than those obtained from ACS for relatively small problems whilst ACS performed better than MMAS for larger problems. The average execution times required by both algorithms slightly altered on all benchmarking problems.

Table 5 The computational results obtained from the MMAS and ACS method.

Problem number	**Methods**	**Best so far solutions**				
		Minimum	Maximum	Average (Av.)	Standard deviation	Av. execution time (Hours)
1	MMAS	62	78	72	6.52	1.08
	ACS	105	117	110.8	5.36	1.05
2	MMAS	22	31	26	3.24	2.25
	ACS	52	59	56.2	3.03	2.30
3	MMAS	520	566	539.4	18.08	3.66
	ACS	515	541	522	10.98	3.86
4	MMAS	406	458	426.4	23.69	4.09
	ACS	349	385	359	15.54	4.33
5	MMAS	339	369	355	11.11	5.91
	ACS	263	286	274	8.89	5.92
6	MMAS	405	441	424	15.33	5.54
	ACS	331	341	336.6	4.39	5.57

7. ACKNOWLEDGEMENTS

The corresponding author would like to acknowledge the Naresuan University since this work was part of the research project supported by the Naresuan University Research Fund under the grant number EN-AR-053/2552.

8. REFERENCES

[1] A. Schaerf, "A survey of automated timetabling," *Artificial Intelligence Review*, vol. 13, pp. 87-127, 1999.

[2] A. S. Asratian and D. de Werra, "A generalized class-teacher model for some timetabling problems," *European Journal of Operational Research*, vol. 143, pp. 531-542, 2002.

[3] N. Balakrishnan, A. Lucena, and R. T. Wong, "Scheduling examinations to reduce second-order conflicts," *Computers & Operations Research*, vol. 19, pp. 353-361, 1992.
[4] S. Daskalaki, T. Birbas, and E. Housos, "An integer programming formulation for a case study in university timetabling," *European Journal of Operational Research*, vol. 153, pp. 117-135, 2004.
[5] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, pp. 268-308, 2003.
[6] P. Pongcharoen, W. Promtet, P. Yenradee, and C. Hicks, "Stochastic Optimisation Timetabling Tool for university course scheduling," *International Journal of Production Economics*, vol. 112, pp. 903-918, 2008.
[7] W. Chainate, P. Thapatsuwan, and P. Pongcharoen, "Investigation on cooling schemes and parameters of simulated annealing for timetabling university course," in *Proceedings of the International Conference on Advanced Computer Theory and Engineering*, Phuket, Thailand, 2008.
[8] R. Alvarez-Valdes, E. Crespo, and J. M. Tamarit, "Design and implementation of a course scheduling system using Tabu Search," *European Journal of Operational Research*, vol. 137, pp. 512-523, 2002.
[9] M. P. Carrasco and M. V. Pato, "A comparison of discrete and continuous neural network approaches to solve the class/teacher timetabling problem," *European Journal of Operational Research*, vol. 153, pp. 65-79, 2004.
[10] D. Dasgupta, *Artificial Immune Systems and Their Applications*. Heidelberg: Springer, 1998.
[11] Y. L. He, S. C. Hui, and E. M. K. Lai, "Automatic timetabling using artificial immune system," *Algorithmic Applications in Management, Proceedings*, vol. 3521, pp. 55-65, 2005.
[12] K. Socha, M. Sampels, and M. Manfrin, "Ant algorithms for the university course timetabling problem with regard to the state-of-the-art," *Applications of Evolutionary Computing*, vol. 2611, pp. 334-345, 2003.
[13] Z. Naji Azimi, "Hybrid heuristics for Examination Timetabling problem," *Applied Mathematics and Computation*, vol. 163, pp. 705-733, 2005.
[14] S. Petrovic, Y. Yang, and M. Dror, "Case-based selection of initialisation heuristics for metaheuristic examination timetabling," *Expert Systems with Applications*, vol. 33, pp. 772-785, 2007.
[15] H. Asmuni, E. K. Burke, J. M. Garibaldi, and B. McCollum, "Fuzzy multiple heuristic orderings for examination timetabling," *Practice and Theory of Automated Timetabling V*, vol. 3616, pp. 334-353, 2005.
[16] E. Burke, M. Dror, S. Petrovic, and R. Qu, "Hybrid Graph Heuristics within a Hyper-Heuristic Approach to Exam Timetabling Problems," *The Next Wave in Computing, Optimization, and Decision Technologies*, pp. 79-91, 2005.
[17] T. Lutuksin, A. Chainual, and P. Pongcharoen, "Development of course timetabling tool: an ant colony approach," in *Proceedings of the 5th International Conference on Developing Real-Life Planning Experiences: Education Reform through Educational Standards*, KMITL, Bangkok, 2007.
[18] D. C. Montgomery, *Design and Analysis of Experiments*, 5 ed. New York: John Wiley & Sons, 2001.
[19] P. Pongcharoen, C. Hicks, and P. M. Braiden, "The development of genetic algorithms for the finite capacity scheduling of complex products, with multiple levels of product structure," *European Journal of Operational Research*, vol. 152, pp. 215-225, 2004.
[20] P. Pongcharoen, C. Hicks, P. M. Braiden, and D. J. Stewardson, "Determining optimum Genetic Algorithm parameters for scheduling the manufacturing and assembly of complex products," *International Journal of Production Economics*, vol. 78, pp. 311-322, 2002.
[21] P. Pongcharoen, D. J. Stewardson, C. Hicks, and P. M. Braiden, "Applying designed experiments to optimize the performance of genetic algorithms used for scheduling complex products in the capital goods industry," *Journal of Applied Statistics*, vol. 28, pp. 441-455, 2001.
[22] P. Pongcharoen, W. Chainate, and S. Pongcharoen, "Improving artificial immune system performance: inductive bias and alternative mutations," *Lecture Notes in Computer Science*, vol. 5132, pp. 220-231, 2008.
[23] W. Chainate, P. Thapatsuwan, and P. Pongcharoen, "Investigation on cooling schemes and parameters of simulated annealing for timetabling university courses," in *Proceedings of the International Conference on Advanced Computer Theory and Engineering*, Phuket, Thailand, 2008.
[24] N. Figlali, C. Ozkale, O. Engin, and A. Figlali, "Investigation of Ant System parameter interactions by using design of experiments for job-shop scheduling problems," *Computers & Industrial Engineering*, vol. 56, pp. 538-559, 2009.
[25] W. J. Gutjahr and M. S. Rauner, "An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria," *Computers & Operations Research*, vol. 34, pp. 642-666, 2007.
[26] T. Bartsch, A. Drexl, and S. Kroger, "Scheduling the professional soccer leagues of Austria and Germany," *Computers & Operations Research*, vol. 33, pp. 1907-1937, 2006.
[27] Z. Liu, J. Shen, H. Wang, and W. Yang, "Regional Bus Timetabling Model with Synchronization," *Journal of Transportation Systems Engineering and Information Technology*, vol. 7, pp. 109-112, 2007.
[28] E. K. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu, "A graph-based hyper-heuristic for

educational timetabling problems," *European Journal of Operational Research*, vol. 176, pp. 177-192, 2007.

[29] A. Chainual, T. Lutuksin, and P. Pongcharoen, "Computer based scheduling tool for multi-product scheduling problems," *International Journal of the Computer, the Internet and Management*, vol. 15, pp. 26.1-6, 2007.

[30] L. Di Gaspero, B. McCollum, and A. Schaerf, "The Second International Timetabling Competition (ITC-2007): Curriculum-based Course Timetabling Track," in *Proceedings of the 14th RCRA workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion*, Rome, Italy, 2007.

[31] P. Kostuch, "The University Course Timetabling Problem with a Three-Phase Approach," *Practice and Theory of Automated Timetabling V*, pp. 109-125, 2005.

[32] T. Stützle and H. Hoos, "Improvements on the ant system: Introducing MAX-MIN ant system," in *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*, 1997.

[33] M. Dorigo and L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 53-66, 1997.

[34] M. Dorigo and T. Stützle, *Ant colony optimization*. Cambridge, Massachusetts: MIT Press, 2004.

[35] M. Dorigo, M. Birattari, and T. Stützle, "Ant colony optimization - Artificial ants as a computational intelligence technique," *Ieee Computational Intelligence Magazine*, vol. 1, pp. 28-39, 2006.

[36] M. Dorigo, V. Maniezzo, and A. Colorni, "Positive feedback as a search strategy," Dipartimento di Elettronica, Politecnico di Milano, Italy, Technical Report 91-016, 1991.

[37] T. Stützle and M. Dorigo, "ACO algorithms for the traveling salesman problem," *Evolutionary Algorithms in Engineering and Computer Science*, pp. 163-183, 1999.

[38] A. Chainual, T. Lutuksin, and P. Pongcharoen, "Forward and backward ant walking strategies for production scheduling with multiple products," in *Proceedings of the 12th Annual Symposium on Computational Science and Engineering*, Ubon Ratchathani, 2008.

[39] N. Leechai, T. Iamtan, and P. Pongcharoen, "Designing machine layout using Rank-based Ant System and Shuffled Frog Leaping," in *Proceedings of the Annual National Operations Research Conference*, Bangkok, 2009.

[40] J. K. Ousterhout, *Tcl and the tk toolkit*. Massachusetts: Addison-Wesley, 1994.

[41] A. Chainual, "Ant colony optimisation for production scheduling in capital goods industries," M.Eng. dissertation, department of Industrial Engineering, Naresuan University, Thailand, 2008.