Lecture Notes in Management Science (2012) Vol. **4**: 87–96
4th International Conference on Applied Operational Research, Proceedings

ISSN 2008-0050 (Print), ISSN 1927-0097 (Online)

Heuristic ordering for ant colony based timetabling tool

Thatchai Thepphakorn and Pupong Pongcharoen

Centre of Operations Research & Industrial Applications, Department of Industrial Engineering, Faculty of Engineering, Naresuan University, Pitsanulok, Thailand
pupongp@nu.ac.th

Abstract. Course timetabling is one of the core operations faced by educational institution. Timetabling problems periodically arise every academic term and are usually solved by academic staff with/without course timetabling tool. The Ant Colony based Timetabling Tool (ANCOTT) has been developed for the university course timetabling. One of ant colony optimisation variants called Rank-based Ant System (AS-rank) has been applied and embedded into the ANCOTT program to seek the feasible timetables with the lowest number of soft constraint violations. New multiple heuristic orderings between the Largest Unpermitted Period Degree first (LUPD) and the Largest Enrollment first (LE) were combined and proposed for reducing the number of infeasible timetables faced by AS-rank. Advanced statistical tools were applied to investigate the optimal setting of AS-rank's parameters. A sequential experiment was designed and carried out using three course timetabling datasets adopted from the third track of the International Timetabling Competition (ITC2007). Serial LUPD+LE and Parallel LUPD&LE outperformed all single heuristic orderings and also produced almost 100% feasible timetables with quicker computational time than that of conventional orderings.

Keywords: heuristic ordering; rank-based ant system; metaheuristics; course timetabling

Introduction

One of the classical annual problems faced by educational institutions is course timetabling. University course timetabling is classified (Socha et al., 2003) as Non-deterministic Polynomial (NP) hard problem, which means that the amount of computational time required increases exponentially with problem size. Solving large course timetabling problems with manual approach is extremely difficult and

may require a group of academic staff to work for several days (MirHassani, 2006). Nowadays with better computing technology, automated timetabling tools based on mathematical models and algorithms are becoming increasingly effective for constructing timetables to the desired specification (Lee and Chen, 2009). Exact algorithms can guarantee optimal solutions, but those algorithms are often infeasible in practice due to unacceptable computational time to find a solution (Blum, 2005). Metaheuristics or approximation optimisation algorithms have been widely accepted for solving course timetabling problem in the last few decades such as Genetic Algorithms (GA) (Pongcharoen et al., 2008), Tabu Search (TS) (Burke et al., 2007), Simulated Annealing (SA) (Chainate et al., 2008), Ant Colony Optimisation (ACO) (Socha et al., 2003), and etc. These algorithms can produce high-quality solutions but do not guarantee optimality (Lewis, 2008).

ACO has been successfully applied to solve various NP-hard problems because ant colonies have a built-in optimisation capacity including path selection based on probabilistic rule and pheromone mechanisms guide ants to find high quality solutions (Dorigo et al., 2006). The use of the ACO method and hybrids to produce educational timetabling has been reported in the literature. For example, Ant Colony System (ACS) has been compared with Max-Min Ant System (MMAS) to construct the best course timetable, in which MMAS outperforms ACS (Socha et al., 2003). Good solutions obtained from MMAS have been found in large problems (Eley, 2006). Elitist Ant System (EAS) has been reported to solve course timetabling problems, its result being superior to the Ant System (AS) (Jaradat and Ayob, 2010). ACO algorithms have also been hybridised with other heuristics (GA, SA and TS) and applied to timetabling problems (Azimi, 2005). Another variant of ACO called the Rank-based Ant System (AS-rank) has been introduced to solve travelling salesman problems (Dorigo and Stützle, 2004), scheduling problems (Chainual et al., 2007), machine layout problems (Leechai et al., 2009), and others. However, the AS-rank algorithm and the optimisation of the associated parameters setting have been neglected in the timetabling literature.

Infeasible timetables are often found during the timetable construction especially with high numbers of courses, students and teachers but less numbers of teaching periods and classrooms. Finding feasible timetables is even more difficult when several hard constraints are imposed (Lewis, 2008). There are three approaches to deal with infeasible solutions: (i) discard them; (ii) apply a high penalty so that they are unlikely to survive to the next iteration; or (iii) repair them (Blum and Roli, 2003). Those approaches directly affect to computational time required to rectify the infeasible timetables (Asmuni et al., 2009).

Heuristic orderings based upon graph coloring approach have been widely accepted for dealing with this problem (Asmuni et al., 2009). The event or course having high priority should be scheduled first in order to avoid generating infeasible solutions (Burke and Newall, 2004). There are two strategies of heuristic ordering for course timetabling: (i) single ordering, for example, Largest Degree first (LD), Largest Enrollment first (LE), Least Saturation Degree first (SD), Largest Coloured Degree first (LCD), Largest Weighted Degree first (LWD), and Random Ordering (RO) (Burke et al., 2007); and (ii) multiple ordering such as Adaptive heuristic orderings (Burke and Newall, 2004), Fuzzy multiple heuristic orderings (Asmuni et al., 2009),

and Graph-based hyper-heuristic (Burke et al., 2007). Those ordering approaches have focused on avoidance of double booking of lecturers, students or classrooms. However, there has been no report on the heuristic ordering directly focusing on other local hard constraints such as the unavailable periods of courses. The objectives of this paper were to: (i) investigate the appropriate setting of AS-rank parameters; (ii) propose new heuristic ordering called Largest Unpermitted Period Degree first (LUPD) and investigating its performances.

The next section describes course timetabling problems followed by a brief explanation on the concepts of heuristic orderings and AS-rank method embedded in the ANCOTT. Then, the experimental design and analysis of computational results are presented before drawing the conclusions.

Course timetabling problems

Timetabling in educational institutions is a crucial activity to schedule courses or examinations, which must be completely assigned into appropriate timeslots for students, lecturers, and classrooms subject to constraints. The general constraints for course timetabling can be classified into two types: hard constraints and soft constraints (Burke et al., 2007; Lewis, 2008). A practical timetable must satisfy all hard constraints, while soft constraints are not essential but the amount of violations associated with soft constraints should be minimised. Hard and soft constraints considered in this work were adopted from the third track of the ITC2007 (Di Gaspero et al., 2007).

Hard constraints (*HC*) are: all lecturing periods of a course must be scheduled into distinct periods (HC_1); two or more lectures cannot be taken in the same room at the same time (HC_2); lectures of courses in the same curriculum or taught by the same teacher must all be scheduled in different periods (HC_3); and a course cannot be scheduled on the unpermitted period (HC_4). Whilst soft constraints (*SC*) are: a number of students attending the course should not exceed the number of seats available in the classroom (SC_1); lectures of a course should at least be spread according to the number of teaching days specified for each course (SC_2); lectures belonging to a curriculum should be adjacent to each other (SC_3); and all lectures of a course should be assigned into single classrooms (SC_4).

The design task was to construct feasible timetables for students, lecturers, and classrooms that satisfy all of the hard constraints and minimise the number of soft constraint violations (Kostuch, 2005). The total violation index (*Z*) was used to assess the quality of timetables by using equation (1).

$$Minimise\ Z = W_1SC_1 + W_2SC_2 + W_3SC_3 + W_4SC_4 \quad (1)$$

Where W_1-W_4 are the weight parameters associated with four soft constraints. Generally, the weight setting is not certainly restricted. Higher weight values indicate higher priority of the associated soft constraints. In this work, the weights (W_1-W_4) were set at 1, 5, 2 and 1, respectively. The number of soft constraint violations were minimised using the AS-rank embedded in the ANCOTT program.

Ant Colony based timetabling tool (ANCOTT)

Generally, the performance of automated timetabling program employing some heuristics or algorithms is more effective in constructing a good timetable. In this present research, Ant Colony based Timetabling Tool (ANCOTT), using new heuristic orderings and Rank-based Ant System (AS-rank), was developed and coded in modular style using the TCL/TK programming language. Heuristic ordering was used to sort the prior courses at the initialisation processes before constructing timetables in the AS-rank processes. The brief pseudo code is shown in Figure 1.

```
upload problem data and assign parameters setting
/*Initialisation processes*/
create and sort events list using RO/LE/LUPD/Series /Parallel
heuristic orderings
create pheromone matrix and candidate timeslots list
while iteration ≤ maximum_iteration do
/*AS-rank processes*/
while ant k < max_ants do
set empty tour (Tk) of ant k
for course = 1 to max_courses do
check feasible timeslots in candidate timeslots list
if available do
choose timeslot into Tk using random proportional rule
else reschedule Tk (infeasible solution)and set IFT = IFT+1
if all courses of Tk were scheduled (feasible solution)do
calculate Z of Tk and set k = k + 1
record the Tbs and sort the ranks of the Tk tours depended on
its Z
update pheromone trail of the Tbs  and the Ti tours(i = 1,
2,3,…,r)
end while
show the best feasible timetable (Tbs)
```

Fig. 1. Pseudo code of AS-rank with the proposed heuristic orderings

Heuristic orderings

Due to the International Timetabling Competition (ITC2007), unavailable periods of courses were imposed as the forth hard constraint (HC_4). High number of unpermitted periods of courses indicated that course had low available teaching periods per week and should be scheduled first in order to avoid infeasible timetable situation. Although this concept is similar to the Least Saturation Degree first (SD), SD is dynamic heuristic ordering that needs reordering unscheduled courses every time (Asmuni et al., 2009). In this work, new single ordering called Largest Unpermitted Period Degree first (LUPD) was proposed. The proposed

ordering was also combined in series and parallel with another well-performed ordering called Largest Enrollment first (LE) (Asmuni et al., 2009).

I. LE: courses were decreasingly sorted by using a number of students enrolled in each course. The course with the highest number of students received the highest priority.
II. LUPD: courses were decreasingly sorted by using a number of unpermitted periods given in the courses. The course with the highest number of unavailable periods received the highest priority.
III. Serial LUPD+LE and LE+LUPD: two different types of single ordering, LE and LUPD, were conducted in series operation. The Serial LUPD+LE means that the LUPD is firstly sorted and then followed by the LE and vice versa for the Serial LE+LUPD.
IV. Parallel LUPD&LE: a number of unpermitted periods and students enrolled in each course were multiplied with their given weights (both weights were fixed at 1) before summation. The course with the highest value of summations received the highest priority.

The heuristic ordering performance was investigated by using two criteria including the percentage of the feasible timetables generated (*%f*) shown in equation (2) and their computational times.

$$\%f = \left(\frac{FT}{FT + IFT} \right) \times 100 \tag{2}$$

FT is the amount of feasible timetables generated by AS-rank. *IFT* is the total number of infeasible timetables found; the *%f* value is distributed from 0 to 100. If there is no infeasible timetable (*IFT* = 0), all ants are feasible timetable (*%f* = 100). After courses were sorted by heuristic ordering, ant constructed a timetable based on Rank-based Ant System, which is described in the next section.

Rank-based Ant System (AS-rank)

AS-rank is variant of Ant Colony Optimisation (ACO) that was proposed by Bullnheimer in 1999 (Dorigo and Stützle, 2004). The general procedures of AS-rank are similar to Ant System (AS) as follows (Dorigo and Stützle, 2004): (i) use random proportional rule to determine the probability of feasible timeslot that ant *k* moving from node (event) *i* to *j* depending on the amount of the pheromone trail (τ_{ij}) and heuristic information from neighbourhood arcs; (ii) evaporate some pheromone from all arcs every generation. A difference procedure of them is pheromone increasing, the AS add some pheromone on the iteration best tour (timetable) (T^{ib}) only while the AS-rank add some pheromone on the best so far timetable (T^{bs}) and on the specified ranks (r) of tour or timetable as shown in equation (3) (Dorigo and Stützle, 2004).

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{r=1}^{w\text{-}1} (w\text{-}r) \Delta\tau_{ij}^{r} + w \Delta\tau_{ij}^{bs} \tag{3}$$

Where w is weight parameter that is varied by the r^{th} ranks except for the T^{bs}. The rank of each timetable is considered by using its total violation index (Z). A number of ranks (r) selected for update are calculated from w-1. For timetable with the highest rank (r = 1) and the lowest Z in current iteration, more pheromone value is added than in other ranks. In this work, the weight (w) parameter was adopted from the Bullnheimer's experiment recommended at 6 (Dorigo and Stützle, 2004).

Experimental design and analysis

In this work, the computational experiments were designed into two steps: (i) investigate the appropriate AS-rank parameter setting via design of experiment and analysis of experiment; and (ii) investigate the performance of the new heuristic orderings proposed in this work. Due to the limitation of computational time and resources required for computational experiments using AS-rank with six types of heuristic orderings, three of twenty-one instants selected from the third track of ITC2007 (Di Gaspero et al., 2007) were adopted for conducting computational experiments. The selected instant problems ranged from small to large sizes (more details in Table 1). All computational runs were based on personal computers with Core 2 Quad 2.66 GHz CPU with 4 GB DDR3 RAM.

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

Problems	*Characteristics of course timetabling problems*						
	Courses	*Classrooms*	*Days/week*	*Periods/day*	*Teachers*	*Curricula*	*Unavailable constraints*
1	30	6	5	6	24	14	53
2	85	17	5	5	68	60	486
3	115	18	5	5	88	67	694

AS-rank's screening experiment

The first experiment was aimed to demonstrate the use of advanced statistical design and analysis to investigate the influence of factors within the AS-rank. The factors and levels are summarised in Table 2.

Table 2. Experimental factors and its levels.

Factors	*Levels*	*Values*		
		Low (-1)	*Medium (0)*	*High (+1)*
AI	3	20*45	30*30	45*20
α	3	0.01	0.5	0.99

Factors	*Levels*	*Values*		
		Low (-1)	*Medium (0)*	*High (+1)*
β	3	0	2.5	5
ρ	3	0.01	0.5	0.99

The considered factors were a combination of the number of ants multiplied by the number of iterations (*AI*), pheromone weight (α), heuristic information weight (β), and pheromone evaporation rate (ρ). The values of the parameters selected were based on previous research (Chainual, 2007). Generally, the combination factor (*AI*) determines the amount of search (feasible timetables generated) in the solution space conducted by AS-rank. This factor is directly related with the size of the problem considered. The high value of this combination usually increases the probability of getting the best solution but requires longer computational time and resources. In this work, the computational limitations were practically imposed; this combination (*AI*) was therefore fixed at 900 in order to accommodate the computational search within the time limit.

Due to the number of parameters and their levels, applying a full factorial design would lead to excessive computation. To overcome this difficulty, the one-third fraction of the 3^{k-1} experimental design (Montgomery, 2012) was adopted for the screening experiment, which reduced the number of computational runs by 66.67% per replication. The second instant problem was considered in this experiment and was repeated five times by using different random seed numbers. The computational results obtained from 135 (3^{4-1}*5) runs were analysed by using a general linear model form of analysis of variance (ANOVA). Table 3 shows an ANOVA table consisting of Source of Variation (Source), Degrees of Freedom (*DF*), Sum of Square (*SS*), Mean Square (*MS*), and *F* and *P* values. A factor with value of $P \leq 0.05$ was considered statistically significant with a 95% confidence interval.

Table 3. ANOVA on the AS-rank's parameters.

Source	*DF*	*SS*	*MS*	*F*	*P*
AI	2	46077	23038	0.220	0.805
α	2	2690490	1345245	12.680	0.000
β	2	179974622	89987311	848.060	0.000
ρ	2	2688160	1344080	12.670	0.000
Seeds	4	27206	6801	0.060	0.992
Error	122	12945439	106110		
Total	134	198371993			

From Table 3, it can be seen that the AS-rank's parameters composed of α, β, and ρ were statistically significant with a 95% confidence interval. The most significant factor was β followed by α and ρ, respectively. The main effect plots suggested that the main factors including *AI*, α, β, and ρ should be defined at 30*30, 0.99, 5, and 0.5, respectively.

Performances of heuristic orderings

The aim for this experiment was to investigate the proposed heuristic orderings using the AS-rank with the optimal parameter setting from earlier experiment. The Random Ordering (RO) was used for benchmarking comparison with others. The experiment was repeated five times using different random seed numbers. The percentage of feasible timetables generated (*%f*) described in equation (2) was statistically analysed in terms of their average (*Avg*) and standard deviation (*SD*), while the average computational times (*T*: hour unit) and the total violation index (Z) were also considered as shown in Table 4.

Table 4. Feasible timetables generated by heuristic orderings on three problem sizes.

Ordering types	*Percentage of feasible timetables generated (%f)*											
	Problem 1			*Avg Z*	*Problem 2*			*Avg Z*	*Problem 3*			*Avg Z*
	Avg	*SD*	*T*		*Avg*	*SD*	*T*		*Avg*	*SD*	*T*	
RO	27.19	1.92	3.61	55.80	27.48	2.26	19.28	419.80	0	0	> 48	-
LE	50.66	5.59	2.31	113.40	94.11	1.05	6.73	420.80	74.67	8.31	9.82	501.80
LUPD	92.94	1.70	1.11	67.00	66.58	1.95	7.80	414.60	20.42	2.15	28.20	511.20
Serial LE+LUPD	99.76	0.20	1.16	70.00	99.80	0.14	5.70	402.40	26.82	6.27	20.65	486.60
Serial LUPD+LE	99.65	0.18	1.23	95.60	99.85	0.17	6.36	411.20	95.45	1.22	8.51	464.60
Parallel LUPD&LE	99.76	0.20	1.13	70.00	99.96	0.06	5.56	423.80	91.66	2.31	7.91	498.60

Remark: the average of *%f* = 0 means that all five computational runs took more than 48 hours and therefore dismissed.

From Table 4, it can be seen that multiple heuristic orderings both Serial LUPD+LE and Parallel LUPD&LE had better performance than all single heuristic orderings for all problems due to the high average %*f* value but low *SD* and consuming time. For example, in medium sized problem, the Parallel LUPD&LE obviously increased %*f* up to 72% and also saved computational times about 71% when compared with the RO type. Although the %*f* obtained from LE was better than LUPD in medium and large sized problems, LUPD was more beneficial for multiple orderings except the Serial LE+LUPD. However, the *Avg* Z obtained from different ordering types slightly varied for medium and large sized problems.

Conclusions

Ant Colony based Timetabling Tool has been developed to solve university course timetabling problems. Experimental design and analysis tools were used to investigate the appropriate parameters setting of the proposed Rank-based Ant System (AS-rank) before sequentially conducting a comparative study on the performance of the proposed heuristic orderings. The statistical analysis on experimental results indicated that some AS-rank's parameters were statistically significant with a 95% confidence interval. The appropriate settings of *AI*, α, β, and ρ parameters were

found best at 30*30, 0.99, 5, and 0.5, respectively. Subsequent experiment also indicated that new multiple heuristic orderings including Serial LUPD+LE and Parallel LUPD&LE increased the percentage of feasible timetable and also saved computational times up to 70%.

Acknowledgments—This work was one part of the research project supported by Naresuan Univer-sity Research Fund under the grant number R2555C035.

References

Asmuni, H., Burke, E.K., Garibaldi, J.M., McCollum, B.and Parkes, A.J., (2009). An investigation of fuzzy multiple heuristic orderings in the construction of university examination timetables. Computers & Operations Research 36: 981-1001.

Azimi, Z.N., (2005). Hybrid heuristics for examination timetabling problem. Applied Mathematics and Computation 163: 705-733.

Blum, C., (2005). Ant colony optimization: Introduction and recent trends. Physics of Life Reviews 2: 353-373.

Blum, C.and Roli, A., (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys 35: 268-308.

Burke, E.K., McCollum, B., Meisels, A., Petrovic, S.and Qu, R., (2007). A graph-based hyper-heuristic for educational timetabling problems. European Journal of Operational Research 176: 177-192.

Burke, E.K.and Newall, J.P., (2004). Solving Examination Timetabling Problems through Adaption of Heuristic Orderings. Annals of Operations Research 129: 107-134.

Chainate, W., Thapatsuwan, P.and Pongcharoen, P., (2008). Investigation on cooling schemes and parameters of simulated annealing for timetabling university courses, Proceedings of the International Conference on Advanced Computer Theory and Engineering, Phuket, Thailand, pp. 200-204.

Chainual, A., (2007). Ant colony optimisation for production scheduling in capital goods industries, Industrial Engineering, Faculty of Engineering. Naresuan University, Pitsanulok, Thailand.

Chainual, A., Lutuksin, T.and Pongcharoen, P., (2007). Computer based scheduling tool for multi-product scheduling problems. International Journal of the Computer, the Internet and Management 15: 26.1-26.6.

Di Gaspero, L., McCollum, B.and Schaerf, A., (2007). The second international timetabling competition (ITC-2007): Curriculum-based course timetabling track, in: Gavanelli, M., Mancini, T. (Eds.), Proceedings of the 14th RCRA workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion, Rome, Italy.

Dorigo, M., Birattari, M.and Stutzle, T., (2006). Ant colony optimization - Artificial ants as a computational intelligence technique. IEEE Comput. Intell. Mag. 1: 28-39.

Dorigo, M.and Stützle, T., (2004). Ant Colony Optimization. MIT Press, Massachusetts.

Eley, M., (2006). Some Experiments with Ant Colony Algorithms for the Exam Timetabling Problem. Lecture Notes in Computer Science 4150: 492-499.

Jaradat, G.M.and Ayob, M., (2010). An Elitist-Ant System for Solving the Post-Enrolment Course Timetabling Problem. Communications in Computer and Information Science 118: 167-176.

Kostuch, P., (2005). The university course timetabling problem with a three-phase approach. Lecture Notes in Computer Science 3616: 109-125.

Lee, Y.and Chen, C.Y., (2009). A heuristic for the train pathing and timetabling problem. Transp. Res. Pt. B-Methodol. 43: 837-851.

Leechai, N., Iamtan, T.and Pongcharoen, P., (2009). Comparison on Rank-based Ant System and Shuffled Frog Leaping for designing multiple row machine layout. SWU Engineeing Journal 4: 102-115.

Lewis, R., (2008). A survey of metaheuristic-based techniques for university timetabling problems. OR Spectrum 30: 167-190.

MirHassani, S.A., (2006). A computational approach to enhancing course timetabling with integer programming. Applied Mathematics and Computation 175: 814-822.

Montgomery, D.C., (2012). Design and Analysis of Experiments, 8 ed. John Wiley & Sons, New York.

Pongcharoen, P., Promtet, W., Yenradee, P.and Hicks, C., (2008). Stochastic Optimisation Timetabling Tool for university course scheduling. International Journal of Production Economics 112: 903-918.

Socha, K., Sampels, M.and Manfrin, M., (2003). Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. Lecture Notes in Computer Science 2611: 334-345.