Int. J. Production Economics ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at SciVerse ScienceDirect

Int. J. Production Economics

journal homepage: www.elsevier.com/locate/ijpe

An ant colony based timetabling tool

Thatchai Thepphakorn [a], Pupong Pongcharoen [a,*], Chris Hicks [b]

[a] Centre of Operations Research and Industrial Applications (CORIA), Industrial Engineering Department, Faculty of Engineering, Naresuan University, Pitsanulok 65000, Thailand
[b] Business School, University of Newcastle upon Tyne, Newcastle upon Tyne, NE1 7RU, UK

ARTICLE INFO

Article history:
Received 11 April 2012
Accepted 16 April 2013

Keywords:
Best-Worst ant
Ant system
Ant colony
Local search
Course timetabling

ABSTRACT

The timetabling of lecturers, seminars, practical sessions and examinations is a core business process for academic institutions. A feasible timetable must satisfy hard constraints. An optimum timetable will additionally satisfy soft constraints, which are not absolutely essential. An Ant Colony based Timetabling (ANCOT) tool has been developed for solving timetabling problems. New variants of Ant Colony Optimisation (ACO) called the Best-Worst Ant System (BWAS) and the Best-Worst Ant Colony System (BWACS) were embedded in the ANCOT program. Local Search (LS) strategies were developed and embedded into BWAS and BWACS to enhance their efficiency and to help find the best timetable with the lowest number of soft constraint violations. Statistical tools for experimental design and analysis were adopted to investigate the factors affecting the BWAS performance. Eight benchmark problems were used for evaluating the performance. For large problems, the BWACS produced the best timetable and was better than the other ACO variants. The best proposed local search strategy enhanced the performance of both the BWAS and the BWACS by up to 74.5%, but this was at the expense of longer execution time.

1. Introduction

Effective timetabling is critical for educational institutions as it affects resource utilisation as well as staff and student satisfaction. Solving large course timetabling problems is extremely difficult and may require a group of people to work for several days (Burke and Petrovic, 2002; MirHassani, 2006). A common approach is to modify previous timetables to meet new requirements (Azimi, 2005; Daskalaki et al., 2004). However, this approach often does not work because the numbers of students, lecturers and student preferences are uncertain and vary from year to year (Azimi, 2005). In recent years, with better computing technology, automated tools based on mathematical models and algorithms are becoming increasingly effective at constructing timetables to the desired specification (Daskalaki et al., 2004; Lee and Chen, 2009).

Timetabling is a combinatorial optimisation (CO) problem. It is a non-deterministic polynomial (NP) hard problem (Daskalaki et al., 2004; Socha et al., 2003), which means that the amount of computation required increases exponentially with problem size. Enumerative search algorithms can guarantee optimal solutions (Blum, 2005), but those algorithms are often infeasible in practice because it takes too long to find an optimal solution (Blum and Roli, 2003; Dorigo et al., 2006). Approximation algorithms, such as metaheuristics, have been widely used for solving large-scale CO problems (Blum, 2005). These algorithms can produce near optimal solutions, in reduced computational time, but they do not guarantee optimum solutions (Blum and Roli, 2003; Lewis, 2008). Blum and Roli (2003) categorised metaheuristic search techniques as (i) single point, such as Tabu Search (TS) (Glover, 1989), Simulated Annealing (SA) (Kirkpatrick et al., 1983) and Iterated Local Search (ILS) (Lourenço et al., 2002) and (ii) population-based, including Genetic Algorithms (GA) (Goldberg, 1989), Particle Swarm Optimisation (PSO) (Kennedy and Eberhart, 2001), Artificial Bee Colony (ABC) optimisation (Pansuwan et al., 2010) and Ant Colony Optimisation (ACO) (Dorigo and Blum, 2005; Dorigo and Stützle, 2004).

In the last decade, ACO has been successfully used to solve various NP-hard problems such as machine layout problems (Leechai et al., 2009), bin packing problems (Thapatsuwan et al., 2008), and scheduling problems (Chainual et al., 2007; Neto and Filho, 2011). Other ACO variants called the Best-Worst Ant System (BWAS) and Best-Worst Ant Colony System (BWACS) have produced high quality solutions for the travelling salesman (Cordón et al., 2000) and quadratic assignment problems (Cordón et al., 2002a). The use of the ACO method for course timetabling has been reported in the literature. For example, the Max-Min Ant System (MMAS) has been used to solve course timetabling problems. It produces good solutions, even for large problems (Eley, 2006; Socha et al., 2003) and it has been shown to perform better than the benchmarking GRASP approach (Dino Matijaš

* Corresponding author. Tel.: +66 55964201; fax: +66 55964003.
E-mail addresses: pupongp@nu.ac.th, pupongp@gmail.com (P. Pongcharoen).

0925-5273/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ijpe.2013.04.026

et al., 2010). Another variant of ACO called the Elitist Ant System (EAS) has been reported to be superior to the Ant System (AS) (Jaradat and Ayob, 2010).

Research has been conducted that has aimed to improve the ACO processes. The Rank-based Ant System (AS-rank) performance has been improved by embedding new multiple ordering heuristics in the AS-rank initialisation process to improve the efficiency of constructing feasible timetables (Thepphakorn and Pongcharoen, 2012). An improved AS-rank method has been shown to perform better than the conventional AS-rank approach with a single ordering heuristic. Djamarus and Ku-Mahamud (2009) improved the AS performance by introducing four heuristic factors and negative pheromone updating strategies in the AS initialisation and pheromone updating processes. These guide an ant so that they avoid infeasible tours. AS with these strategies help to construct better timetables than that without those strategies.

Ant Colony System (ACS) have also been hybridised with other heuristics (e.g. GA, SA and TS) and applied to solve both course and examination timetabling problems. The hybrid ACS-TS outperforms the other hybridisations for examination timetabling problems (Azimi, 2005), whilst the hybrid ACS-SA outperforms other ACO variants for solving course timetabling problems (Ayob and Jaradat, 2009). Double compact pheromone matrices (Nothegger et al., 2012) have been introduced to exploit information in the AS solution construction process for solving post enrolment course timetabling problems. The AS has produced high quality solutions even without using SA, but better solutions could be obtained when including it. The Die-Hard Co-Operative Ant Behaviour approach proposed by Ejaz and Javed (2007) was been introduced to find feasible course timetables in the first phase before getting into the optimisation phase. Five out of eleven cases related to medium-large problem sizes which produced new globally best solutions within limited time using this approach. The Hypercube framework (Johnson et al., 2006) was implemented with the MMAS (called MTH-MMAS) for solving university course timetabling problems and achieved good results for small and medium problems.

There have been a number of research articles that have focused upon improving metaheuristics by adopting optimal parameter settings (Figlali et al., 2009; Naderi et al., 2010; Pongcharoen et al., 2007; Thapatsuwan et al., 2012) or hybridisation approaches (Azimi, 2005; Pongcharoen et al., 2008a; Shelokar et al., 2007). Due to the nature and complexity of the problem domains, some of these algorithms are problem specific. The performance of the algorithms usually depends on the parameter settings (Li et al., 2010; Pongcharoen et al., 2008b; Zandieh et al., 2009). There are several ways to select parameter settings: ad hoc selection (Aytug et al., 2003); adopting recommendations of the previous work; a best-guess approach (Montgomery, 2012) or systematically identifying optimum settings through designed experiments. Due to the problem specific nature of the algorithms there is no generic optimal parameter set that can be efficiently applied to every problem domain (Figlali et al., 2009). Thus, the settings recommended by previous studies will only be applicable in similar domains. Trial-and-error experiments can be used to identify good parameter settings, but this approach is based upon experience and intuition. It can be costly and time-consuming and sometimes impossible to verify that the best values have been identified (Chen et al., 2009). The one-factor-at-a-time experimental strategy has been adopted by some researchers, but this approach is inefficient and fails to consider any possible interaction between the factors (Figlali et al., 2009). When there is interaction between factors the effect of one factor will vary according to the levels of other factors. Montgomery (2012) suggested that the correct approach for dealing with several factors is to conduct a factorial experiment, in which factors are systematically varied together, instead of one at a time. Relatively few researchers have investigated optimal parameter settings for metaheuristics by using proper experimental designs.

The objectives of this paper are to: (i) describe the development of the BWAS and BWACS for solving university course timetabling problems; (ii) demonstrate the use of experimental design and analysis for investigating the appropriate BWAS parameter settings; (iii) verify the performance of the algorithms with appropriate parameter settings; (iv) compare the performance of BWAS and BWACS with various ACO variants (including AS, ACS, MMAS, EAS and AS-rank) in terms of average results and convergence speeds; (v) improve the performance of both the BWAS and BWACS methods by combining the approaches with new local search (LS) strategies; and (vi) compare the performance of the combined approaches with the original BWAS and BWACS algorithms in terms of the quality of the results obtained, solution convergence speed, and the computational time required.

The next section describes course timetabling problems. Section 3 briefly explains the concepts of the BWAS and the BWACS. Section 4 considers the application of those methods and proposes local search strategies which are embedded in the Ant Colony based Timetabling (ANCOT) tool. Section 5 presents the experimental design, analysis and results followed by conclusions.

2. Course timetabling problem

There are many types of general timetabling problem such as nurse rostering, sports timetabling, transportation timetabling, and educational timetabling (Burke et al., 2007). In educational institutions, timetabling courses and examinations is a crucial activity, which assigns appropriate timeslots for students, lecturers, and classrooms. The general constraints in course timetabling can be classified into two types: hard constraints (*HC*) and soft constraints (*SC*) (Burke et al., 2007; Lewis, 2008). Hard constraints are the most important and must be satisfied to have a feasible timetable (Burke and Newall, 2004). For example, it is necessary to avoid the double booking of lecturers, students or classrooms. Soft constraints are more relaxed as some violations are acceptable; however, algorithms should aim to minimise the number of violations. Eighteen soft constraints have been reported in literature (Pongcharoen et al., 2008b), but they do not apply in all institutions (e.g. compulsory lunch times). A commercial version of the timetabling tool may require some customisation to cover special constraints such as those related to cultural or religious issues that may apply to universities in other countries.

The second international timetabling competition (the third tracks) described hard and soft constraints (Di Gaspero et al., 2007). The hard constraints considered in this work were: (i) all lectures within a course must be scheduled and assigned to distinct periods (HC_1); (ii) only one lecture can take place in the same classroom during the same period (HC_2); (iii) lectures within different modules or taught by the same lecturers must be scheduled in other periods (HC_3); and (iv) if a teacher for a course is not available to give a lecture during a given period, it cannot be scheduled during that period (HC_4). The soft constraints considered in this research were: (i) for each module, the number of students attending the course must be less or equal to the number of seats for all the classrooms hosting the lectures (SC_1); (ii) the lectures for each module must be spread into a minimum number of days (SC_2); (iii) lectures belonging to a programme should be adjacent to each other (i.e., in consecutive periods) (SC_3); and (iv) each lecture for a module should take place in the same classroom (SC_4).