# Ant Algorithms for the Exam Timetabling Problem

Michael Eley

Aschaffenburg University of Applied Science,
Logistics Laboratory (LlAb), Aschaffenburg, Germany
michael.eley@fh-aschaffenburg.de

**Abstract.** Scheduling exams at universities can be formulated as a combinatorial optimization problem. Basically one has to schedule a certain number of exams in a given number of time periods so that a predetermined objective function is minimized. In particular, the objective function penalizes schedules where students have to write exams in consecutive periods or even in the same period. Ant colony approaches have been demonstrated to be a powerful solution approach for various combinatorial optimization problems. This paper presents two ant colony approaches for the exam timetabling problem, a Max–Min and an ANTCOL approach. Using the Toronto benchmark test cases from the literature, both algorithms arc compared to other timetabling heuristics. Finally, the Max–Min and ANTCOL algorithms are compared using the same set of test cases. In spite of some shortcomings, the ANTCOL approach turned out to be a worthwhile algorithm, among the best currently in use for examination timetabling.

## 1 Introduction

The exam timetabling problem faces the problem of scheduling exams within a limited number of available periods. Setting up a conflict-free timetable is not a trivial task due to limited resources like periods, examination rooms and teacher availability. The main objective is to balance out student workload and to distribute the exams evenly within the planning horizon. In particular, it should be avoided that a student has to write two exams in the same period. Such situations will be referred to as conflicts of order 0 in the sequel. Additionally, as few students as possible have to attend $r$ exams within $y$ consecutive periods. Such conflicts can either be totally forbidden by constraints or penalized in the objective function. For example, Carter et al. proposed in [14] a cost function that imposes penalties $P_\omega$ for a conflict of order $\omega$, i.e. whenever one student has to write two exams scheduled within $\omega + 1$ consecutive periods. In the literature $\omega$ normally runs from 1 to 5 with $P_1 = 16, P_2 = 8, P_3 = 4, P_4 = 2, P_5 = 1$.

Solving practical exam timetabling problems requires that additional constraints have to be considered, e.g. some exams have to be written before other exams or some exams cannot be written within specific periods. References [8] and [13] give comprehensive lists of possible hard and soft constraints.

E.K. Burke and H. Rudová (Eds.): PATAT 2006, LNCS 3867, pp. 364–382, 2007.

The exam timetabling problem can be formulated as a graph coloring problem. Each node represents one exam. Undirected arcs connect two nodes if at least one student is enrolled in both corresponding exams. Weights on the arcs represent the number of student enrolled in both exams. The objective is to find a coloring where no adjacent nodes are marked with the same color or to minimize the weighted sum of the arcs that connect two nodes marked with the same color. The exam timetabling problem is a generalization of the graph coloring problem, as in the objective function also conflicts of higher orders are penalized.

A large number of papers presenting heuristic solution approaches to the exam timetabling problem have been published in recent years. Most of the approaches on exam timetabling are modified heuristics derived from graph coloring approaches or use local search methods. Additionally, hyper-heuristics, i.e. heuristics that choose heuristics, have been applied to the exam timetabling problem.

Carter et al. applied in [14] some well known graph coloring heuristics, i.e. saturation degree, largest degree, largest weighted degree, largest enrolment and color degree, which they combined with backtracking. These graph coloring heuristics have been integrated into various other approaches. Asmuni et al. [2] used fuzzy functions to find exams that arc difficult to schedule and those should be scheduled early when using graph coloring heuristics.

Di Gaspero aud Schearf [20] tested different variants of tabu search based techniques whose neighborhoods concerned those which contributed to the violations of hard or soft constraints. Di Gaspero [19] improved the approach by employing multiple neighborhoods. The first one considers only exams that contribute to the objective function and changes the period of a single exam. The second neighborhood exchanges the periods of two groups of exams at once. White and Xie [39] developed a tabu search approach. This approach was extended in [40] by employing long-term memory. Paquete and Stuetzle [30] developed a tabu search methodology for exam timetabling where ordered priorities were given for the constraints. The length of the tabu list was adaptively set by considering the number of violations in the solutions.

Merlot et al. [27] and Burke et al. [6] developed variants of simulated annealing approaches. While the first paper also uses simulated annealing in combination with constraint programming to generate the initial solution, the latter presents a great deluge algorithm. This approach was further studied in [10] and in [5]. Cote et al. [18] investigated a bi-objective evolutionary algorithm with the objectives of minimizing timetable length and spacing out conflicting exams.

As well as evolutionary algorithms, simulated annealing and tabu search, other local search techniques have been tested to solve exam timetabling problems. Abdullah et al. [1] developed a large neighborhood search based on the methodology of improvement graph construction. Ayob et al. [3] as well as Burke et al. [7] investigated variants of variable neighborhood search. The results of the latter approach were further improved by using a standard genetic algorithm to intelligently select subset of neighborhoods. Caramia et al. [12] developed a local search method where a greedy scheduler assigned exams into the least possible number of periods, and a penalty decreaser improved the timetable without increasing the number of periods. Finally, Casey and Thompson [15] investigated a greedy randomized adaptive search procedures approach.

The solution quality of many meta-heuristics strongly depends on how well several parameters are chosen. This problem of parameter adjustment led a number of researchers to develop new technologies aimed at operating at a higher level of generality. Kendall and Hussin [24] investigated a tabu search based hyper-heuristic where both moving strategies and constructive graph heuristics were employed as low level heuristics. Yang and Petrovic [41] employed case-based reasoning to choose graph heuristics to construct initial solutions for the Great Deluge algorithm. Burke et al. [9] investigated using tabu search to search permutations of graph heuristics to construct solutions for timetabling problems.

Comprehensive surveys on the literature on exam timetabling problems can be found in [13,37]. In particular, the latter paper lists more than 150 references from journal articles and books published since the mid-1990s.

The aim of this paper is twofold. Originally, this research was motivated by the need for a software tool to practically solving an exam timetabling problem. As ant colony approaches have been demonstrated to be a powerful tool for various combinatorial optimization problems (see the survey in [21]), it is apparent that one can adapt this solution approach to the exam timetabling problem. In the literature different variants of ant colony approaches have been suggested. A comparison of some of these strategies with respect to their suitability for the exam timetabling problem will be made.

This paper is organized as follows. In Section 2 a detailed problem formulation will be presented. Section 3 will give an introduction to ant colony systems. The following sections will present a solution approach and test results for some benchmark problems that are taken from the literature. Finally, Section 6 summarizes the results and discusses suggestions for future work.

## 2 Problem Formulation

Before stating the problem formally, some notation will be introduced.

$R$ index set of rooms
$I$ index set of exams
$T$ index set of periods
$\Omega$ index set of order of conflicts
$K_{rt}$ capacity of room $r$ in period $t$
$c_{ij}$ number of students enrolled in exam $i$ as well as in exam $j$
$E$ total number of students
$E_i$ number of students enrolled in exam $i$
$P_\omega$ penalty imposed if one student has to write two exams within $\omega + 1$ periods

$y_{it}$ binary decision variable equal to 1 if exam $i$ is scheduled in period $t$ and 0 otherwise

$p_{irt}$ decision variable indicating the number of students of exam $i$ assigned to room $r$ in period $t$.

Using this notation, the exam timetabling problem can be formulated as follows:

$$\min \frac{1}{E} \sum_{\omega \in \Omega} \sum_{i,j \in I, i \neq j} \sum_{t \in T, t > \omega} P_\omega c_{ij} y_{it} y_{j(t-\omega)} \tag{1}$$

s.t.

$$\sum_{t \in T} y_{it} = 1 \quad \forall i \in I \tag{2}$$

$$p_{irt} \leq y_{it} K_{rt} \quad \forall i \in I, \forall r \in R, \forall t \in T \tag{3}$$

$$\sum_{r \in R} \sum_{t \in T} p_{irt} = E_i \quad \forall i \in I \tag{4}$$

$$\sum_{i \in I} p_{irt} \leq K_{rt} \quad \forall r \in R, \forall t \in T \tag{5}$$

$$\sum_{t \in T} c_{ij} y_{it} y_{jt} = 0 \quad \forall i, j \in I, i \neq j \tag{6}$$

$$y_{it} \in \{0, 1\} \quad \forall i \in I, \forall t \in T \tag{7}$$

$$p_{irt} \in \mathbb{N}_0 \quad \forall i \in I, \forall r \in R, \forall t \in T. \tag{8}$$

The objective function (1) balances out students' workload by minimizing the weighted sum of all conflicts which is divided by the number of students. Constraint (2) states that each exam is assigned to exactly one period. If an exam is not assigned within a period, then no seats should be reserved for that period in any room. This is imposed by constraint (3). Constraints (4) and (5) ensure that the number of seats reserved for an exam will be equal to the number of students who are enrolled in that exam and that the room capacities are not exceeded. Finally, constraint (6) avoids conflicts of order 0, i.e. that a student has to write two exams in the same period.

The exam timetabling problem is a generalization of the graph coloring problem, which is known to be NP-hard [23]. Therefore, to solve large real-world problems within a reasonable amount of time, heuristics are used. In the following sections a solution approach will be presented. But instead of considering capacity constraints for the single rooms, only the total capacity of all available exam rooms within a period will be considered. In the IP formulation stated above this can be accomplished by replacing the set of rooms by an artificial single room.

## 3 Ant Algorithms

Ant colony optimization algorithms represent special solution approaches for combinatorial optimization problems derived from the field of swarm intelligence. They were first introduced by Colorni et al. in the early 1990s [16]. See [21] for an in-depth introduction to ant systems.

Ant algorithms were inspired by the observation of how real ant colonies find shortest paths between food sources and their nest. This observation was first implemented in algorithms for solving the traveling salesperson problem (TSP). This type of ant colony optimization algorithm is known in the literature as ant system (AS) algorithms. The basic principle of AS algorithms will be described briefly by means of the TSP. This solution approach to the TSP will then be adopted to solving the exam timetabling problem in the next section.

The solution approach consists of $n$ cycles. In each of these cycles first each of the $m$ ants constructs a feasible solution. In AS each ant builds a complete tour that visits all nodes. Obviously, this solution neither has to be optimal nor must it be even close to the (unknown) optimal value. Improved solutions can be obtained if knowledge gathered by other ants in the past on how high quality solutions can be obtained, is incorporated into the ants' decision. Assume that an ant is located in a node $i$. To choose the next node $j$ that has not yet been visited by that ant one may apply one of the following two randomized strategies:

**Strategy I:** *Constructive heuristic.* Apply one priority rule like randomized nearest neighbor. Decision values for all nodes $j$ are determined by the inverse of the distance from node $i$ to that node $j$. The next node the ant moves to is then randomly chosen according to the probabilities determined by those decision values. Consequently, if node $j_1$ is closer to $i$ than node $j_2$ it is more likely to choose node $j_1$. The decision values of the constructive heuristic will be referred to as $\eta_{ij}$.

**Strategy II:** *Pheromone trails.* This strategy is mainly inspired by the way real ants find shortest paths. While commuting between two places on different possible paths ants deposit a chemical substance called pheromone. The shorter the path is the more often the ant will use this path within a limited period of time and, consequently, the larger the amount of pheromone will be on that path. Thus, whenever an ant has to choose between different available paths it will prefer the one with higher amount of pheromone.

To adapt these observations to the TSP, the amount of pheromone is stored in a matrix $\tau$. Each cell $\tau_{ij}$ of the matrix is associated with an arc $(ij)$ and the matrix is initialized with 0 for all cells. After an ant has completed a tour, the values of the cells that belong to the arcs the ant has chosen are updated by the inverse of the obtained objective function value, i.e. the length of the tour. The amount of pheromone trail $\tau_{ij}$ associated to arc $(i, j)$ is intended to represent the learned desirability of choosing node $j$ when in node $i$. Consequently, arcs that belong to good solutions receive a high amount of pheromone.

AS algorithms combine these two strategies. The probability that an ant $\nu$ located in node $i$ chooses the next node $j$ is determined by the following formula:

$$p_{ij}^{\nu} = \begin{cases} \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{k \in N_i^{\nu}} (\tau_{ik})^{\alpha}(\eta_{ik})^{\beta}} & \text{if } j \in N_i^{\nu} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

$\alpha$ and $\beta$ are a given weighting factors and $N_i^{\nu}$ is the set of nodes that have not yet been visited by ant $\nu$ currently located in node $i$ (see [21]).

Excepting the TSP, AS algorithms have been implemented for various combinatorial optimization problems, such as the quadratic assignment problem or the sequential ordering problem. Different variants of AS algorithms have been suggested in the literature, for example ant colony systems (ACS) and Max–Min ant systems (MMAS). Both produced much better results than AS (see [21]). In particular, MMAS, which was first proposed by Stuetzle and Hoos [33], generated significantly better solutions for the TSP than AS. Socha et al. [32] compared the MMAS variant with ACS and found that MMAS outperformed the ACS approach for the considered timetabling problem.

The main modifications of MMAS are related to the way that the matrix $\tau$ is initialized and how the pheromone values are updated. Additionally, MMAS uses local search to improve the solutions found by the ants. Details are discussed in the next section.

Ant colony algorithms have been used recently to solve different types of scheduling problems, see for example Blum [4]. Socha et al. [31,32] developed algorithms for the timetabling problem for university classes, which is slightly different from the exam timetabling problem considered here. Costa and Hertz [17] used an ant colony approach to solve assignment type problems, in particular graph coloring problems. They pursued the objective of minimizing the number of periods required for a clash-free timetable. Recently, Vesel and Zerovuik [38] as well as Dowsland and Thompson modified and improved [22] this graph coloring algorithm. Whilst the first article puts the focus on graph coloring problems, the latter improves the approach of Costa and Hertz [17] with respect to the examination scheduling problem, by introducing new initialization methods, trial calculations and fitness functions.

Finally, Naji Azimi [28] implemented an ACS algorithm and compared it to simulated annealing, a genetic algorithm and tabu search. These results with randomly generated test problems indicated that the ant-based approach outperformed the other approaches. Additionally, the author tested some hybrid versions where ACS was combined with tabu search. These hybrid versions outperformed all meta-heuristics [29].

## 4 An Ant Algorithm for the Exam Timetabling Problem

### 4.1 General Modifications for the Exam Timetabling Problem

In this section a description will be given on how the AS algorithm must be modified in order to solve exam timetabling problems. The solution approach consists of $n$ cycles. Firstly, in each of these cycles each of the $m$ ants constructs a feasible solution therefore using the constructive heuristic and the pheromone trails. These exam schedules are then evaluated according to the given objective function and the experience accumulated during the cycle is used to update the pheromone trails.

Depending on the choice of a constructive heuristic and the way the pheromone values are used, there are different ways in which this basic solution approach can be adapted to the exam timetabling problem.

- In Socha et al. [32] a pre-ordered list of events is given. Each ant chooses the color for a given node probabilistically similar to the formula (9). The pheromone trail $\tau_{Ij}$ contains information on how good the solution was, when node $i$ was colored by color $t$. The constructive heuristic employed in their approach is not described.
- Socha et al. [31] suggested two variants of how the basic solution approach presented in the last section can be adapted to the (course) timetabling problem:
  - A list of periods is given. Starting with the first period in the list, each ant assigns courses to this period.
  - A list of courses is given and the ants chooses a period for each course of the list while starting with the first course from the list. The period is then chosen probabilistically according to a formula similar to (9).

  The authors preferred the second variant as it seemed to be a more natural approach, in particular when the number of events is larger than the number of periods. Additionally, the authors tested two different representations for the pheromone values in the matrix $\tau$:

  - Direct representation. A cell $\tau_{it}$ of the matrix represents the amount of pheromone if course $i$ is assigned to a period $t$.
  - Indirect representation. A cell $\tau_{ij}$ of the matrix indicates if courses $i$ and $j$ should be assigned to the same period.

  Experiments indicted that the indirect representation produced better results, in particular when an additional hill climber was used.
- In the ACS approach of Naji Azimi [29] each ant follows a list of exams and chooses a period as in [31]. The period is chosen randomly with probabilities depending on the pheromone matrix and heuristic information. The paper gives no information on how a time list of exams is sorted.
- At each stage of the construction process in the AS approach of Costa and Hertz [17] called ANTCOL the ant chooses first a node $i$ according to a probability distribution equivalent to (9) and then a feasible color. Experiments showed that also choosing the color probabilistically did not improve the solutions. As in [31] an indirect representation was employed. The matrix $\tau$ provides information on the objective function value, i.e. the number of colors required to color the graph, which was obtained when nodes $i$ and $j$ are colored with the same color.

In contrast to elite strategies where only the ant that found the best tour from the beginning of the trial deposits pheromone, all ants deposit pheromone on the paths they have chosen. According to [21] this strategy is called ant cycle strategy.

Different priority rules were tested as constructive heuristic. Among those chosen in each step, the node with the highest degree of saturation, i.e. the number of different colors already assigned to adjacent nodes, achieved the best results with respect to solution quality and computation times.

For the exam timetabling problem the way that the information in matrix $\tau$ is used in both approaches seems not to be meaningful. Due to the conflicts of higher orders the quality of a solution does not depend on how a pair of exams is scheduled, nor does it depend on the specific period an exam is assigned to. For example, assigning two exams $i$ and $j$ with $c_{ij} = 0$ to the same period can either result in a high or in a low objective function value as the quality of the solution strongly depends on when the remaining exams are scheduled. In the following a two-step approach was implemented.

**Step I:** Determine the sequence according to the exams are scheduled. As for the TSP we assume that an ant located in a node, corresponding to an exam, has to visit all other nodes, i.e. it has to construct a complete tour. The sequence according to which this ant constructs the tour corresponds to the sequence in which the exams are scheduled. Thus, $\tau_{ij}$ indicates how advantageous it is to choose exam $j$ as the $i$th exam in the sequence.

The rationale behind this idea is that the ants should learn which exams should be scheduled early, in order to avoid high penalties in the objective function in later iterations. Asmnuni et al. [2] as well as Merlot et al. [27] pursued similar concepts. Whilst in the first paper fuzzy functions were employed when ordering the exams on how difficult they were, the latter ordered the exams by the size of their domains (available periods) and scheduled them into the earliest period one by one.

**Step II:** Find the most suitable period for an exam which should be scheduled. As recommended by Costa and Hertz [17] the period is not chosen probabilistically. Instead, all admissible periods are evaluated according to the given penalty function and the exam is scheduled in the period that achieved the best evaluation. Thus, as recommended by Dowsland and Thompson [22] the construction approach is not restricted to filling one period before starting another.

If exam $j$ has a high $\tau_{ij}$ value for a small value of $i$, this exam should be scheduled early in the sequence. Assume that this exam is by chance not chosen as number $i$ in the sequence. Then values $\tau_{k,j}$ for $k = i{+}1, i{+}2, \ldots$ should also be high in order to make sure that the ant will choose $j$ soon. Therefore, Merkle and Middendorf modified (9) by the following so-called pheromone summation rule [26]:

$$p_{ij}^{\nu} = \begin{cases} \frac{\left(\sum_{h=1}^{i}[\tau_{hj}]\right)^{\alpha}(\eta_{ij})^{\beta}}{\sum_{k \in N_i^{\nu}} \left(\sum_{h=1}^{i}[\tau_{hk}]\right)^{\alpha}(\eta_{ik})^{\beta}} & \text{if } j \in N_i^{\nu} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Pheromone values $\tau_{ij}$ along the ants' paths are updated by the inverse of the objective function value. For the heuristic value $\eta_{ij}$ the following simple priority rule for graph coloring was implemented. The exam with the smallest number of available periods is selected. A period would not be available for an exam if it caused a conflict of order 0 with another exam that has already been scheduled. This priority rule corresponds to the saturation degree rule (SD) which was tested in [14]. The value $\eta_{ij}$ is chosen to be the inverse of the saturation degree.

### 4.2 MMAS Specifications

MMAS approaches mainly differ from AS algorithms in the way they use the existing information (see [33]):

- Pheromone trails are only updated by the ant that generated the best solution in a cycle. The corresponding values $\tau_{ij}$ are updated by $\rho\tau_{ij} + 1/f^{best}$ where $f^{best}$ is equal to the best objective function value found so far. For all other arcs $(i, j)$ that are not chosen by the best ant $\tau_{ij}$ is updated by $(1 - \rho)\tau_{ij}$, where $\rho \in [0, 1]$ represents the pheromone evaporation factor, i.e. the percentage of pheromone that decays within a cycle.
- Pheromone trail values are restricted to the interval $[\tau_{min}, \tau_{max}]$, i.e. whenever after a trail update $\tau_{ij} < \tau_{min}$ or $\tau_{ij} > \tau_{max}$ then $\tau_{ij}$ is set to $\tau_{min}$ or $\tau_{max}$, respectively. The rationale behind this is that if the differences between some pheromone values were too large, all ants would almost always generate the same solutions. Thus, stagnation is avoided.
- Pheromone trails are initialized to their maximum values $\tau_{max}$. This type of pheromone trail initialization increases the exploration of solutions during the first cycle.

The solution quality of ant colony algorithms can be considerably improved when it is combined with additional local search. In hybrid MMAS only the best solution within one cycle is improved by local search. For the exam timetabling problem a hill climber procedure has been implemented. Within an iteration of the hill climber two sub-procedures are carried out in succession. The hill climber is stopped if no improvement can be found within an iteration.

Within the first sub-procedure of the hill climber for all exams the most suitable period is examined. Beginning with the exam that causes the biggest contribution to the objective function value, all feasible periods are checked and the exam is assigned to its best period. The first sub-procedure is stopped if all exams have been checked without finding an improvement. Otherwise the contributions to the objective function value are recalculated and the process is repeated.

The second sub-procedure tries to decrease the objective function value by swapping all exams within two periods, i.e. all exams assigned to period $t'$ are moved to period $t''$ and the exams of that period are moved to period $t'$. Therefore all pairs of periods are examined and the first exchange that leads to an improvement is carried out. Again, the process is repeated as long as the objective function value is decreased.

Finally, the use of a so-called candidate can reduce required computational times as well as improve solution quality at the same time (see [21]). Such a list provides additional local heuristic information as it contains preferred nodes to be visited from a given node. Instead of scanning all other exams, only the exams in the candidate list are examined, and only when all exams in this list have been scheduled, are the remaining exams considered.

## 5 Computational Experiments

The proposed Max–Min algorithm was implemented in Borland Delphi 7.0. It will be referred to as MMAS-ET in the sequel. Test runs were carried out on a computer with 3.2 GHz clock under Windows XP.

### 5.1 Test Cases

To benchmark algorithms test cases of thirteen practical examination problems can be found on the site of Carter (see [34]). Table 1 summarizes some characteristics of these problems. These test problems are also known in the literature as the Toronto data set version I (see [37]).

To make a comparison meaningful all algorithms must use the same objective function. Therefore, Carter proposed weighting conflicts according to the following penalty function: $P_1 = 16, P_2 = 8, P_3 = 4, P_4 = 2, P_5 = 1$, where $P_\omega$ is the penalty for the constraint violation of order $\omega$. The cost of each conflict is multiplied by the number of students involved in both exams. The objective function value represents the costs per student. As the proposed MMAS-ET algorithm does not guarantee that no conflicts of order 0 occur, additionally, the penalty $P_0$ was imposed and set to 10 000.

### 5.2 Adjustment of the Parameters

The required parameters were specified as follows. The number of cycles was set to 50. Within each cycle 50 ants were employed to construct solutions. The candidate list contained the 20% of exams with the lowest number of available periods. Several test runs were carried out in order to determine the required parameters appropriately:

- The evaporation rate $\rho$ was set to 0.3. As in [33] it turned out that this parameter is quite robust, i.e. the parameter $\rho$ does not clearly influence the performance.
- For the restrictions of the pheromone interval values to strategies were tested. Setting $\tau_{max} = 1/\rho$ obtained slightly better results than in the case of variable $\tau_{max}$ and $\tau_{min}$ as proposed in [33].
- Different values for the weighting factors $\alpha$ and $\beta$ were tested. It turned out that the approach performed best when $\alpha$ was set to one and $\beta$ was chosen high. Best results were obtained for $\beta$ equal to 10. But the difference was on average less than one percent when $\beta$ was bigger than eight. A high $\beta$ forces

**Table 1.** Toronto data set version I from Carter et al. [14,34,35]

| Test case | Number of exams | Number of students | Number of student exams | Problem density | Number of periods |
|---|---|---|---|---|---|
| car92I | 543 | 18419 | 55522 | 13.8% | 32 |
| car91I | 682 | 16925 | 56877 | 12.8% | 35 |
| ear83I | 190 | 1125 | 8109 | 26.7% | 24 |
| hec92I | 81 | 2823 | 10632 | 42.0% | 18 |
| kfu93I | 461 | 5349 | 25113 | 5.6% | 20 |
| lse91I | 381 | 2726 | 10918 | 6.3% | 18 |
| rye92I | 486 | 11483 | 45051 | 7.5% | 23 |
| sta83I | 139 | 611 | 5751 | 14.4% | 13 |
| tre92I | 261 | 4360 | 14901 | 5.8% | 23 |
| uta92I | 622 | 21267 | 58979 | 12.6% | 35 |
| ute92I | 184 | 2750 | 11793 | 8.5% | 10 |
| yor83I | 181 | 941 | 6034 | 28.9% | 21 |
| pur93I | 2419 | 30032 | 120681 | 2.9 % | 43 |

that exams which can be scheduled, due to zero-order conflicts, only in a few remaining periods are scheduled first, as they are given a much higher probability in (9). Remember that $\eta_{ij}$ is the inverse of the saturation degree as explained in Section 4.1. Thus, a high $\beta$ value has the same effect like a candidate list. This could be a reason why the use of the candidate list did not improve the solutions. For small values of $\beta$, i.e. values lower than 5, solutions with zero-order conflicts could not always be avoided.
- As the approach is non-deterministic each test case was solved 20 times.

After determining the parameters in such a way, it turned out that less than 2% of the solutions were generated more than once. Thus, stagnation, that is caused by the fact that many ants generate almost the same solutions, could not be observed.

### 5.3 Test Results for the MMAS-ET Approach

Table 2 displays the results for different approaches. The solutions are also available on the Internet [36]. For each approach the minimal objective function value and the average result after 20 test runs are given in Table 2. Results of the proposed MMAS-ET approach are given in the second column.

In order to find out how much the hill climber contributes to the solution, the MMAS-ET approach was also tested without making use of the hill climber. Comparing the results in the second and in the third column it is obvious that the hill climber considerably improves the solutions.

Increasing the number of ants and the number of cycles to 100 in the MMAS-ET approach did not result in achieving better solutions. Neither the average value of all 20 iterations was improved nor were better solutions found during the 20 iterations.

**Table 2.** Results (objective function values) for two variants of the MMAS-ET approach for 20 test runs

| | MMAS-ET | | MMAS-ET without hill climber | |
|---|---|---|---|---|
| Test case | Best | Avg. | Best | Avg. |
| car92I | 4.7 | 4.8 | 7.7 | 7.9 |
| car91I | 5.7 | 5.8 | 9.4 | 9.6 |
| ear83I | 36.8 | 38.3 | 50.4 | 53.8 |
| hec92I | 11.2 | 11.4 | 15.0 | 15.5 |
| kfu93I | 15.0 | 15.4 | 24.0 | 24.8 |
| lse91I | 12.2 | 12.6 | 18.9 | 19.6 |
| rye93I | 10.1 | 10.3 | 17.9 | 18.5 |
| sta83I | 157.4 | 157.6 | 162.1 | 163.6 |
| tre92I | 8.9 | 9.2 | 12.2 | 12.7 |
| uta92I | 3.8 | 3.8 | 6.0 | 6.2 |
| ute92I | 27.7 | 28.4 | 32.9 | 34.4 |
| yor83I | 39.3 | 40.2 | 50.2 | 51.4 |
| pur93I | 5.5 | 5.6 | 12.2 | 12.5 |

### 5.4 Comparison with the Approach of Costa and Hertz

The results of MMAS-ET were compared with a modified version of the ANTCOL algorithm of Costa and Hertz [17], which was originally developed for solving graph coloring problems. This approach will be called ANTCOL-ET in the sequel. Within that approach the ANT_DSATUR(1) procedure was used as a constructive method as described in [17]. Also the objective function was modified in order to consider conflicts of higher order. Test runs were carried out to adjust the parameters appropriately. The parameter $\alpha$ was set to 1, $\beta$ to 30. $\rho$ was set equal to 0.3. Again, each test case was solved 20 times.

Table 3 shows the results for the 13 test cases and compares them with the MMAS-ET approach. Surprisingly, the simple AS-like approach ANTCOL-ET outperformed the MMAS-ET for some test cases. In particular, this result is contrary to other results presented in the literature where MMAS algorithms obtained better results for various combinatorial optimization problems by avoiding stagnation (see [21,33]).

Thus, ANTCOL-ET was modified by implementing additionally the hill climber already incorporated in the MMAS-ET approach. This modified version of the Costa and Hertz approach provided, on average, better solutions than the MMAS-ET approach. Obviously, the combination of an indirect representation plus a hill climber is capable of generating good solutions. Socha et al. [31] made a similar observation for the course timetabling problem.

Computing times for the MMAS-ET approach lay in the range of 10 seconds for the smallest test cases, i.e. hec-s-92, to 2.5 hours for the pur-s-93 problem. Compared to the MMAS-ET approach the computing time of the ANTCOL-ET combined with the hill climber was on the average 80% higher. Thus, one

**Table 3.** Comparison of the objective function values of different ant colony approaches

| | | MMAS-ET | ANTCOL-ET without hill climber | ANTCOL-ET with hill climber | ANTCOL-ET with hill climber |
|---|---|---|---|---|---|
| Stopping criterion | | 2500 solutions | 2500 solutions | 2500 solutions | Same running time as MMAS-ET |
| car92I | best | 4.7 | 4.5 | 4.3 | 4.3 |
| | avg. | 4.8 | 4.6 | 4.4 | 4.4 |
| car91I | best | 5.7 | 5.3 | 5.2 | 5.2 |
| | avg. | 5.8 | 5.4 | 5.2 | 5.3 |
| ear83I | best | 36.8 | 40.3 | 36.8 | 38.1 |
| | avg. | 38.3 | 41.4 | 38.3 | 38.5 |
| hec92I | best | 11.2 | 12.2 | 11.1 | 11.2 |
| | avg. | 11.4 | 12.6 | 11.4 | 11.4 |
| kfu93I | best | 15.0 | 15.4 | 14.5 | 14.6 |
| | avg. | 15.4 | 15.8 | 14.9 | 14.9 |
| lse91I | best | 12.2 | 11.9 | 11.3 | 11.4 |
| | avg. | 12.6 | 12.2 | 11.7 | 11.7 |
| rye93I | best | 10.1 | 10.2 | 9.8 | 9.8 |
| | avg. | 10.3 | 10.7 | 10.0 | 10.0 |
| sta83I | best | 157.4 | 158.2 | 157.3 | 157.3 |
| | avg. | 157.6 | 159.3 | 157.5 | 157.5 |
| tre92I | best | 8.9 | 8.8 | 8.6 | 8.6 |
| | avg. | 9.2 | 9.0 | 8.7 | 8.7 |
| uta92I | best | 3.8 | 3.6 | 3.5 | 3.5 |
| | avg. | 3.8 | 3.7 | 3.5 | 3.5 |
| ute92I | best | 27.7 | 28.9 | 26.4 | 26,7 |
| | avg. | 28.4 | 29.4 | 27.0 | 27.5 |
| yor83I | best | 39.3 | 42.2 | 39.4 | 40.1 |
| | avg. | 40.2 | 43.7 | 40.4 | 40.7 |
| pur93I | best | 5.5 | 4.8 | 4.6 | 4.6 |
| | avg. | 5.6 | 4.9 | 4.6 | 4.6 |

can conclude that ANTCOL-ET takes more time but gets a better solution quality than MMAS-ET. Note that the same stopping criterion was used for both algorithms, namely 2500 solutions.

Additionally, test runs for the ANTCOL-ET approach were carried out where the running time was limited to the time that the MMAS-ET approach required to generate 2500 solutions. The results are displayed in the last column of Table 3. Thus, comparing both algorithms with the same running time the ANTCOL-ET approach outperformed the MMAS-ET approach in ten out of the thirteen test cases.

Finally, a second variant of the ANTCOL-ET approach was implemented. As in the approach of Costa and Hertz an indirect representation was employed in this variant. But instead of using the AS framework, the pheromone trails were initialized and updated as in the MMAS approach described in Section 4.2. Also

**Table 4.** Best (b.) and average (a.) objective function value for the Toronto data set version I (best solutions printed in bold)

| Test case | | Cal96 [14] | Cal01 [12] | DGS01 [20] | DG02 [19] | PS02 [30] | BN03 [10] | Mal03 [27] | Wal04 [40] | BN04 [11] | Bal04 [6] | Aal05 [2] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| car91I | b. | 7.1 | 6.6 | 6.2 | 5.7 | - | 4.6 | 5.1 | 5.7 | 5.0 | 4.8 | 5.3 |
| | a. | 7.1 | 6.6 | 6.5 | 5.8 | | 4.7 | 5.2 | 5.8 | - | 6.1 | - |
| car92I | b. | 6.2 | 6.0 | 5.2 | - | - | 4.0 | 4.3 | 4.6 | 4.3 | 4.2 | 4.6 |
| | a. | 6.2 | 6.0 | 5.6 | - | - | 4.1 | 4.4 | 4.7 | - | 4.3 | - |
| ear83I | b. | 36.4 | **29.3** | 45.7 | 39.4 | 40.5 | 36.1 | 35.1 | 45.8 | 36.2 | 35.4 | 37.0 |
| | a. | 36.4 | **29.3** | 46.7 | 43.9 | 45.8 | 37.1 | 35.4 | 46.4 | - | 36.7 | - |
| hec92I | b. | 10.8 | **9.2** | 12.4 | 10.9 | 11.7 | 11.3 | 10.6 | 12.9 | 11.6 | 10.8 | 11.8 |
| | a. | 10.8 | **9.2** | 12.6 | 11.4 | 12.4 | 11.5 | 10.7 | 13.4 | - | 11.5 | - |
| kfu93I | b. | 14.0 | 13.8 | 18.0 | - | 16.5 | 13.7 | 13.5 | 17.1 | 15.0 | 13.7 | 15.8 |
| | a. | 14.0 | **13.8** | 19.5 | - | 18.3 | 13.9 | 14.0 | 17.8 | - | 14.4 | - |
| lse91I | b. | 10.5 | **9.6** | 15.5 | 12.6 | 13.2 | 10.6 | 10.5 | 14.7 | 11.0 | 10.4 | 12.1 |
| | a. | 10.5 | **9.6** | 15.9 | 13.0 | 15.5 | 10.8 | 11.0 | 14.8 | - | 11.0 | - |
| rye92I | b. | 7.3 | **6.8** | - | - | - | - | 8.4 | 11.6 | - | 8.9 | 10.4 |
| | a. | 7.3 | **6.8** | - | - | | - | 8.7 | 11.7 | - | 9.3 | - |
| sta83I | b. | 161.5 | 158.2 | 160.8 | 157.4 | 161.2 | 168.3 | 157.3 | 158.0 | 161.9 | 159.1 | 160.4 |
| | a. | 161.5 | 158.2 | 167.0 | 157.7 | 168.7 | 168.7 | 157.4 | 158.0 | - | 159.4 | - |
| tre92I | b. | 9.6 | 9.4 | 10.0 | - | 9.3 | 8.2 | 8.4 | 8.9 | 8.4 | 8.3 | 8.7 |
| | a. | 9.6 | 9.4 | 10.5 | - | 10.2 | 8.4 | 8.6 | 9.2 | - | 8.4 | - |
| uta92I | b. | 3.5 | 3.5 | 4.2 | 4.1 | - | 3.2 | 3.5 | 4.4 | 3.4 | 3.4 | 3.6 |
| | a. | 3.5 | 3.5 | 4.5 | 4.3 | - | 3.2 | 3.6 | 4.5 | - | 3.5 | - |
| ute92I | b. | 25.8 | **24.4** | 27.8 | - | 28.7 | 25.5 | 25.1 | 29.0 | 27.4 | 25.7 | 27.8 |
| | a. | 25.8 | **24.4** | 31.3 | - | 30.5 | 25.8 | 25.2 | 29.1 | - | 26.2 | - |
| yor83I | b. | 41.7 | 36.2 | 41.0 | 39.7 | 38.9 | 36.8 | 37.4 | 42.3 | 40.8 | 36.7 | 40.7 |
| | a. | 41.7 | **36.2** | 42.1 | 40.6 | 41.7 | 37.3 | 37.9 | 42.5 | - | 37.2 | - |
| ≥MMAS | | 5 | 4 | 11 | 6 | 7 | 2 | 0 | 11 | 4 | 1 | 7 |
| ≥ANTCOL | | 6 | 5 | 11 | 6 | 7 | 2 | 3 | 12 | 6 | 1 | 12 |

Cal96: Carter et al.; Cal01: Caramia et al.; DGS01: Di Gaspero and Schaerf; DG02: Di Gaspero; PS02: Paquete and Stuetzle (Lex-seq approach); BN03: Burke and Newall; Mal03: Merlot et al.; Wal04: White et al.; BN04: Burke and Newall; Bal04: Burke et al.; Aal05: Asmuni et al.

the trail values were restricted and the hill climber was used. This MMAS variant of the approach of Costa and Hertz generated solutions comparable to the MMAS-ET approach. Only for the sta83I test case was the objective function value of 157.3 for the best solution better than the best solution generated by the MMAS-ET approach. And for the yor83I test case both other approaches were outperformed with an objective function value of 39.2. Thus, the test results indicate that, irrespective of the representation, the ANTCOL approach outperforms the MMAS approach.

**Table 5.** Best (b.) and average (a.) objective function value for the Toronto data set version I (best solutions printed in bold)

| Test case | | Cal05 [18] | KH05 [24] | YP05 [41] | Aal06 [1] | Bal06a [9] | Bal06b [7] | ABK [3] | BB06 [5] | MMAS -ET | ANTCOL -ET |
|---|---|---|---|---|---|---|---|---|---|---|---|
| car91I | b. | 5.4 | 5.4 | 4.5 | 5.2 | 5.4 | 4.6 | 4.5 | **4.4** | 5.7 | 5.2 |
| | a. | 5.5 | - | **4.5** | - | - | - | - | - | 5.8 | 5.2 |
| car92I | b. | 4.2 | 4.7 | 3.9 | 4.4 | 4.5 | 4.0 | 4.9 | **3.7** | 4.7 | 4.3 |
| | a. | 4.3 | - | **4.0** | - | - | - | - | - | 4.8 | 4.4 |
| ear83I | b. | 34.2 | 40.2 | 33.7 | 34.9 | 37.9 | 32.8 | 36.3 | 32.8 | 36.8 | 36.8 |
| | a. | 35.6 | - | 34.9 | - | - | - | - | - | 38.3 | 38.3 |
| hec92I | b. | 10.4 | 11.9 | 10.8 | 10.3 | 12.3 | 10.0 | 11.1 | 10.2 | 11.2 | 11.1 |
| | a. | 10.5 | - | 11.4 | - | - | - | - | - | 11.4 | 11.4 |
| kfu93I | b. | 14.3 | 15.8 | 13.8 | 13.5 | 15.2 | **13.0** | 14.7 | **13.0** | 15.0 | 14.5 |
| | a. | 14.4 | - | 14.4 | - | - | - | - | - | 15.4 | 14.9 |
| lse91I | b. | 11.3 | - | 10.4 | 10.2 | 11.3 | 10.0 | 12.1 | 9.8 | 12.2 | 11.3 |
| | a. | 11.5 | - | 10.8 | - | - | - | - | - | 12.6 | 11.7 |
| rye92I | b. | 8.8 | - | 8.5 | 8.7 | - | - | 10.7 | - | 10.1 | 9.8 |
| | a. | 9.1 | - | 8.8 | - | - | - | - | - | 10.3 | 10.0 |
| sta83I | b. | **157.0** | 157.4 | 158.4 | 159.2 | 158.2 | 159.9 | 157.3 | **157.0** | 157.4 | 157.3 |
| | a. | **157.1** | - | - | - | - | - | - | - | 157.6 | 157.5 |
| tre92I | b. | 8.6 | 8.4 | 7.9 | 8.4 | 8.9 | 7.9 | 8.9 | **7.8** | 8.9 | 8.6 |
| | a. | 8.8 | - | **8.1** | - | - | - | - | - | 9.2 | 8.7 |
| uta92I | b. | 3.5 | - | **3.1** | 3.6 | 3.9 | 3.2 | 3.6 | **3.1** | 3.8 | 3.5 |
| | a. | 3.6 | - | **3.2** | - | - | - | - | - | 3.8 | 3.5 |
| ute92I | b. | 25.3 | 27.6 | 25.4 | 26.0 | 28.0 | 24.8 | 26.4 | 24.8 | 27.7 | 26.4 |
| | a. | 25.5 | - | 26.1 | - | - | - | - | - | 28.4 | 27.0 |
| yor83I | b. | 36.4 | - | 36.5 | 36.2 | 41.4 | 37.3 | 39.0 | **34.8** | 39.3 | 39.4 |
| | a. | 37.6 | - | 36.9 | - | - | - | - | - | 40.2 | 40.4 |
| ≥MMAS-ET | | 0 | 3 | 1 | 1 | 8 | 1 | 3 | 0 | | |
| ≥ANTCOL | | 4 | 7 | 1 | 4 | 11 | 1 | 7 | 0 | | |

Cal05: Cote et al.; KH05: Kendall and Hissan; YP05: Yang and Petrovic;
Aal06: Abdullah et al.; Bal06a: Burke et al.; Bal06b: Burke et al.;
ABK06: Ayob et al.; BB06: Burke and Bykov .

### 5.5 Comparison with Other Exam Timetabling Approaches

The MMAS-ET as well as the ANTCOL-ET approach were compared with approaches from the literature. These benchmark approaches minimize the objective function of Carter et al. presented in Section 5.1. The approaches have been tested using the Toronto data set version I. Results of the benchmarks are taken from the literature [37,40] and from the internet (see the timetabling database at the University of Melbourne [35]).

Tables 4 and 5 display the best solution and the average solution achieved. The results can be summarized as follows. Although neither the MMAS-ET nor the ANTCOL-ET approach can improve the best solution for any of the 12 test cases, their performance is comparable with most of the 19 benchmark algorithms. It is striking that no approach outperforms all other approaches for all test cases. Therefore, the last two lines in Tables 4 and 5 indicate how often the MMAS-ET approach and the ANTCOL-ET approach, respectively, obtained solutions that were not worse than the benchmark approach corresponding to the column in the table.

The ANTCOL-ET approach is capable of finding better solutions for at least some test cases compared to almost all benchmark approaches. For example, the highly competitive Bal04 approach of Burke et al. was outperformed for one test case, namely sta83I. Good results were obtained in particular for the test cases sta83I, car92I, uta92I and car91I. Only the recently published algorithm of Burke and Bykov [5] clearly dominates both ant algorithms for all test cases. But, for this flex-deluge approach, running times between five and ten hours were reported which are considerably higher than for ANTCOL-ET.

There are also some test cases where MMAS-ET outperforms the approaches. For example, MMAS-ET generates better solutions than the Cal01 approach of Caramia et al. (that holds the best results in five out of the twelve test cases) in four out of the twelve test cases, i.e. for the test cases car91I, car92I, sta83I and tre92I. White et al. argued in [40] that these test cases seem to be in a way easier than the other test cases.

## 6 Conclusion

In this paper different strategies for solving exam timetabling problems by ant algorithms have been implemented. In ant colony optimization the search for good solutions is incorporated in the learning process controlled by the definition of the pheromone trail and the constructive heuristic. Different representations for the pheromone values have been tested as well as different strategies for updating the pheromone, i.e. ant systems (AS) and Max–Min ant systems (MMAS).

The most effective ant algorithm turned out to be a modified version of the AS approach of Costa and Hertz [17]. In particular, a hill climber was added to this approach that improves the solutions and has a strong impact on the solution quality. Unlike other combinatorial optimization problems, for example the TSP or the QAP, the exam timetabling problem using the MMAS approach did not outperform the simpler AS strategy.

Of course, it goes without saying that proper adjusting parameters can improve the performance of an algorithm considerably. In particular the values of $\alpha$ and $\beta$ have a strong impact on the solution quality.

The implemented algorithms have been compared with the existing literature on the problem. Unfortunately, the experimental analysis shows that the results of our algorithms are not satisfactory on all benchmark instances. Nevertheless, we consider these results quite encouraging, and they provide a good basis for future improvements. One promising direction could be the parallelization of the ANTCOL-ET or the MMAS-ET approach. Recently, Manfrin et al. [25] tested different interconnection topologies for a Max–Min approach to the TSP.

They showed that the parallel models outperformed the equivalent sequential algorithms.

Another self-evident extension of the approach would be to incorporate additional constraints and requirements, like for example scarce room resources or precedence constraints between exams.

## References


1. Abdullah, S., Ahmadi, S., Burke, E.K., Dror, M.: Investigating Ahuja–Orlins large neighbourhood search for examination timetabling. OR Spectrum 29, 331–372 (2007)
2. Asmuni, H., Burke, E.K., Garibaldi, J., McCollum, B.: Fuzzy multiple ordering criteria for examination timetabling. In: Burke, E.K., Trick, M.A. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 334–353. Springer, Heidelberg (2005)
3. Ayob, M., Burke, E.K., Kendall, G.: An iterative re-start variable neighbourhood search for the examination timetabling problem. In: Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling, Brno, pp. 336–344 (August 2006)
4. Blum, C.: Theoretical and Practical Aspects of Ant Colony Optimization. Akademische Verlagsgesellschaft, Berlin (2004)
5. Burke, E.K., Bykov, Y.: Solving exam timetabling problems with the flex-deluge algorithm. In: Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling, Brno, pp. 370–372 (August 2006)
6. Burke, E.K., Bykov, Y., Newall, J.P., Petrovic, S.: A time-predefined local search approach to exam timetabling problems. IIE Transactions 36, 509–528 (2004)
7. Burke, E.K., Eckersley, A.J., McCollum, B., Petrovic, S., Qu, R.: Hybrid variable neighbourhood approaches to university exam timetabling. Technical Report NOTTCS-TR-2006-2, University of Nottingham (2006)
8. Burke, E.K., Elliman, D.G., Ford, P.H., Weare, R.F.: Examination timetabling in British universities: A survey. In: Burke, E.K., Ross, P. (eds.) Practice and Theory of Automated Timetabling. LNCS, vol. 1153, pp. 76–92. Springer, Heidelberg (1996)
9. Burke, E.K., McCollum, B., Meisels, A., Petrovic, S., Qu, R.: A graph based hyper-heuristic for exam timetabling problems. European Journal of Operational Research 176, 177–192 (2007)
10. Burke, E.K., Newall, J.P.: Enhancing timetable solutions with local search methods. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, pp. 195–206. Springer, Heidelberg (2003)
11. Burke, E.K., Newall, J.P.: Solving examination timetabling problems through adaptation of heuristic orderings. Annals of Operational Research 129, 107–134 (2004)
12. Caramia, M., Dell'Olmo, P., Italiano, G.F.: New algorithms for examination timetabling. In: Swierstra, S.D. (ed.) PLILP 1995. LNCS, vol. 982, pp. 230–241. Springer, Heidelberg (1995)
13. Carter, M.W., Laporte, G.: Recent developments in practical examination timetabling. In: Burke, E.K., Ross, P. (eds.) Practice and Theory of Automated Timetabling. LNCS, vol. 1153, pp. 3–21. Springer, Heidelberg (1996)
14. Carter, M.W., Laporte, G., Lee, S.Y.: Examination timetabling algorithmic strategies and applications. Journal of the Operational Research Society 47, 373–383 (1996)

15. Casey, S., Thompson, J.: Grasping the examination scheduling problem. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, pp. 233–244. Springer, Heidelberg (2003)
16. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed optimization by ant colonies. In: Proceedings of the 1st European Conference on Artificial Life, pp. 134–142. Elsevier, Amsterdam (1992)
17. Costa, D., Hertz, A.: Ants can color graphs. Journal of the Operational Research Society 48, 295–305 (1997)
18. Cote, P., Wong, T., Sabouri, R.: Application of a hybrid multi-objective evolutionary algorithm to the uncapacitated exam proximity problem. In: Burke, E.K., Trick, M.A. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 151–168. Springer, Heidelberg (2005)
19. Di Gaspero, L.: Recolour, shake and kick: A recipe for the examination timetabling problem. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, Springer, Heidelberg (2003)
20. Di Gaspero, L., Schaerf, A.: Tabu search techniques for examination timetabling. In: Burke, E., Erben, W. (eds.) PATAT 2000. LNCS, vol. 2079, pp. 104–117. Springer, Heidelberg (2001)
21. Dorigo, M., Di Caro, G., Gambarella, L.M.: Ant algorithms for discrete optimization. Artificial Life 5, 137–172 (1999)
22. Dowsland, K., Thompson, J.: Ant colony optimisation for the examination scheduling problem. Journal of the Operational Research Society 56, 426–439 (2005)
23. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New York (1979)
24. Kendall, G., Hussin, N.M.: A tabu search hyper-heuristic approach to the examination timetabling problem at the Mara University of Technology. In: Burke, E.K., Trick, M.A. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 199–218. Springer, Heidelberg (2005)
25. Manfrin, M., Birattari, M., Stuetzle, T., Dorigo, M.: Parallel ant colony optimization for the traveling salesman problem. In: Dorigo, M., Gambardella, L.M., Birattari, M., Martinoli, A., Poli, R., Stützle, T. (eds.) ANTS 2006. LNCS, vol. 4150, pp. 224–234. Springer, Heidelberg (2006)
26. Merkle, M., Middendorf, M.: An ant algorithm with a new pheromone evaluation rule for total tardiness problems. In: Oates, M.J., Lanzi, P.L., Li, Y., Cagnoni, S., Corne, D.W., Fogarty, T.C., Poli, R., Smith, G.D. (eds.) EvoIASP 2000, EvoWorkshops 2000, EvoFlight 2000, EvoSCONDI 2000, EvoSTIM 2000, EvoTEL 2000, and EvoROB/EvoRobot 2000. LNCS, vol. 1803, pp. 287–296. Springer, Heidelberg (2000)
27. Merlot, L.T.G., Boland, N., Hughes, P.J., Stuckey, B.D.: A hybrid algorithm for the examination timetabling problem. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, pp. 207–231. Springer, Heidelberg (2003)
28. Naji Azimi, Z.: Comparison of metaheuristic algorithms for examination timetabling problem. Applied Mathematics and Computation 16, 337–354 (2004)
29. Naji Azimi, Z.: Hybrid heuristics for examination timetabling problem. Applied Mathematics and Computation 163, 705–733 (2005)
30. Paquete, L., Stuetzle, T.: Empirical analysis of tabu search for the lexicographic optimization of the examination timetabling problem. In: Burke, E.K., De Causmaecker, P. (eds.) PATAT 2002. LNCS, vol. 2740, pp. 413–420. Springer, Heidelberg (2003)

31. Socha, K., Knowles, J., Samples, M.: A max–min ant system for the university course timetabling problem. In: Dorigo, M., Di Caro, G.A., Sampels, M. (eds.) Ant Algorithms. LNCS, vol. 2463, pp. 1–13. Springer, Heidelberg (2002)
32. Socha, K., Sampels, M., Manfrin, M.: Ant algorithms for the university course timetabling problem with regard to state-of-the-art. In: Raidl, G.R., Cagnoni, S., Cardalda, J.J.R., Corne, D.W., Gottlieb, J., Guillot, A., Hart, E., Johnson, C.G., Marchiori, E., Meyer, J.-A., Middendorf, M. (eds.) EvoCOP 2003. LNCS, vol. 2611, pp. 334–345. Springer, Heidelberg (2003)
33. Stuetzle, T., Hoos, H.H.: Max–min ant systems. Future Generation Computer Systems 16, 889–914 (2000)
34. http://www.cs.nott.ac.uk/~rxq/data.htm
35. http://www.or.ms.unimelb.edu.au/timetabling
36. http://www.fh-aschaffenburg.de/index.php?idlogdown
37. Qu, R., Burke, E.K., McCollum, B., Merlot, L.T.G., Lee, S.Y.: A survey of search methodologies and automated approaches for examination timetabling. Computer Science Technical Report No. NOTTCS-TR-2006-4, University of Nottingham (2006)
38. Vesel, A., Zerovni, J.: How well can ants color graphs? Journal of Computing and Information Technology (CIT) 8, 131–136 (2000)
39. White, G.M., Xie, B.S.: Examination timetabling and tabu search with longer-term memory. In: Burke, E., Erben, W. (eds.) PATAT 2000. LNCS, vol. 2079, pp. 85–103. Springer, Heidelberg (2001)
40. White, G.M., Xie, B.S., Zonjic, S.: Using tabu search with long-term memeory and relaxation to create examination timetables. European Journal of Operational Research 153, 80–91 (2004)
41. Yang, Y., Petrovic, S.: A novel similarity measure for heuristic selection in examination timetabling. In: Burke, E.K., Trick, M.A. (eds.) PATAT 2004. LNCS, vol. 3616, pp. 377–396. Springer, Heidelberg (2005)