Artificial Bee Colony Algorithm for Solving Educational Timetabling Problems

Article · August 2014

DOI: 10.4018/jncr.2012040101

4 authors:

Asaju La'aro Bolaji
Federal University Wukari

Ahamad Tajudin Khader
Universiti Sains Malaysia

Mohammed Azmi Al-Betar
Al-Balqa' Applied University

Mohammed a. Awadallah
Al-Aqsa University

Artificial Bee Colony Algorithm for Solving Educational Timetabling Problems

Asaju La'aro Bolaji, Universiti Sains Malaysia, Malaysia, & University of Ilorin, Nigeria

Ahamad Tajudin Khader, Universiti Sains Malaysia, Malaysia

Mohammed Azmi Al-Betar, Universiti Sains Malaysia, Malaysia, & Jadara University, Jordan

Mohammed A. Awadallah, Universiti Sains Malaysia, Malaysia

ABSTRACT

This paper presents an artificial bee colony algorithm (ABC) for Education Timetabling Problem (ETP). It is aimed at developing a good-quality solution for the problem. The initial population of solutions was generated using Saturation Degree (SD) and Backtracking Algorithm (BA) to ensure the feasibility of the solutions. At the improvement stage in the solution method, ABC uses neighbourhood structures iteratively within the employed and onlooker bee operators, in order to rigorously navigate the UTP search space. The technique was evaluated using curriculum-based course timetabling (CB-CTT) and Uncapacitated Examination Timetabling Problem (UETP) problem instances. The experimental results on UETP showed that the technique is comparable with other state-of-the-art techniques and provides encouraging results on CB-CTT.

Keywords: *Artificial Bee Colony (ABC), Course Timetabling, Examination Timetabling, Nature-Inspired Computing, Swarm Intelligence, Timetabling*

INTRODUCTION

Bio-inspired algorithms are general-purpose stochastic search techniques mimicking natural evolution phenomena. One important reason for the success of bio-inspired algorithms is the ability to navigate the problem search space using different operators which prevents them from getting stuck in the local optima and therefore increases the possibility of finding global optima. As a result, bio-inspired algorithms could be viewed as global search techniques which are successfully applied by researchers to solve a wide range of timetabling problems. Some examples of commonly used bio-inspired algorithms include: Artificial Immune System (AIS) (Malim, Khader, & Mustafa, 2006), Ant Colony Optimisation (ACO) (Socha, Sampels, & Manfrin, 2003), Firefly Algorithm (Yang, 2009), Genetic Algorithm (GA) (Abdullah, Turabieh, McCollum, & Burke, 2010), Harmony Search Algorithm (Al-Betar & Khader, 2008)

DOI: 10.4018/jncr.2012040101

and Particle Swarm Optimisation (Sheau Fen Ho, Safaai, & Hashim, 2009). Artificial Bee Colony Algorithm (ABC) is a search method motivated by the foraging behavior of honeybee swarm, and target complex optimization problems. The ABC algorithm that was developed by Karaboga in 2005 is a population-based heuristic algorithm.

One of the most common example of scheduling problems is Educational Timetabling Problem (ETP), which could be described as the scheduling of resources for assignments under predefined constraints, such that it enhances the possibility of allocation or reduces the violation of such constraints (Thanh, 2007). ETP is a difficult administrative process, which needs to be repeated every semester in academic institutions such as universities and colleges. ETP has also been reported to be a hard combinatorial optimisation problem (Johnson & Garey, 1979), and was tackled using different approaches by workers of operational research and artificial intelligence. Generally, operation of ETP could be defined as the assignment of a set of events (courses or exams) to given timeslots and rooms, based on some given constraints. Two types of constraints peculiar to timetabling problems are hard and soft constraints; the hard constraints must be strictly satisfied in the solution to be feasible yet, the quality of the solution is determined by satisfying the soft constraints satisfaction. Furthermore, ETP can be classified into three categories: course timetabling problem, examination timetabling problem and school timetabling problem. Therefore, this study considered both course and examination timetabling problems. In order to tackle the problems, two neighbourhood structure mechanisms are incorporated within the operators of ABC for CB-CTT while three neighbourhood structures are used for UETP. For the purpose of evaluation, a dataset established by ITC-2007 is used for the course timetabling problem whereas the Carter Dataset is used for the examination timetabling problem.

The last three decades have witnessed tremendous application of techniques to educational timetabling problem. These techniques range from constraints and mathematical programming techniques (Cambazard et al., 2012; De Causmaecker et al., 2009; Goltz & Matzke, 1998); local search based techniques such as simulated annealing (Abdullah et al., 2010; Burke et al., 2004; McCollum, 2010; Thompson & Dowsland, 1998), great deluge (Burke & Bykov, 2006; Landa-Silva & Obit, 2009), Tabu search (White & Xie, 2001; White et al., 2004), hybrid techniques such as (Abdullah et al., 2007; Burke et al., 2005; Turabieh & Abdullah, 2011), population and nature-inspired based techniques (Eley, 2006; Nothegger et al., 2012; Al-Betar et al., 2010; Al-Betar & Khader, 2008), and so on. The comprehensive survey of methodologies used in educational timetabling problem can be found in Burke and Petrovic (2002), Lewis (2008), Qu et al. (2009), and Schaerf (1999).

Recently, ABC was tailored for Course Timetabling (CB-CTT) by Bolaji et al. (2011) and later adapted and hybridized with simulated annealing by Alzaqebah and Abdullah (2011a, 2011b) for examination timetabling, where series of neighbourhood structures and selection strategies were introduced. However, the exploitation capability of ABC methods still needs further investigation in order to enhance its performance. Therefore, the main aim of this paper is fold: (i) to investigate and enhance the performance of ABC on different educational timetabling problem, and (ii) to study a suitable neighbourhood structures combined with process for each problem considered.

The rest of this paper is organised as follows; the next two sections presents detailed description and mathematical formulations of CB-CTT and UETP; while the fundamentals of ABC and its adoptions are discussed in the fourth section; details of the computational results and comparison with the average of results are discussed in the fifth section. The final section presents the conclusion and possible future directions.

COURSE TIMETABLING PROBLEM

The Course Timetabling Problem is the assignment of a set of courses to given periods and rooms, based on some given (hard and soft) constraints. Course Timetabling Problem exists in two forms; post enrolment course timetabling (PE-CTT) and curriculum-based course timetabling problems (CB-CTT), the focus of the current paper. Formulation of the first CB-CTT problem was initially proposed by Di Gaspero and Schaerf (2006) and a later version was presented in 2007 as one of the tracks by the organisers of international timetabling competition (ITC, 2007)[1]. The purpose of the competition was to close the gap between existing research and real practice in the education timetabling domain (McCollum et al., 2010). Several optimisation techniques and algorithms have been applied to solve CB-CTT; some of these developed include *mathematical programming* such as integer programming (Burke, Mareček, Parkes, & Rudová, 2008; Lach & Lübbecke, 2010) and *metaheuristics* (Abdullah, Turabieh, et al., 2010; Burke et al., 2008; Geiger, 2009, 2010; Lü & Hao, 2010; Lü, Hao, & Glover, 2011; Shaker & Abdullah, 2009).

Mathematical techniques such as the integer programming approach was proposed by Lach and Lübbecke (2010), where the method used was broken into two steps: the initial step was used to guarantee the feasibility, whereas the later step was used to enhance the feasible solutions to the state-of-the-art results. Burke et al. (2008) presented the branch and cut technique for CB-CTT problem which was divided into two phases; an alternative integer programming formulation was employed at the first phase, where it was used to lower the number of variables, resulting into a slight increase in the number of soft constraints. In the second phase, the branch and cut procedure was used for further improvement. At this phase, the constraints necessary to attain optimality from enumeration of events/free-period patterns were added. The results generated showed that it was possible to obtain optimal solutions for two instances within a reasonable time period by using the method. CB-CTT was also studied by Lü and Hao (2010) using hybrid algorithm called Adaptive Tabu Search (ATS). ATS technique was integrated to the neighborhood structures and combined with perturbation from iterated local search. The results showed that the method enhanced the previously reported results. The ITC-2007 problem was investigated by Müller (2009) and was evaluated as the best in the CB-CTT competition track. The method used one-stage solution approach which combined great deluge and simulated annealing incorporated into the constraint solver library. The iterative local search was used at the initial stage and the solution was further enhanced with the aid of constraint-based statistics (CBS). In another development, the application of local search based approach for CB-CTT was presented by Geiger (2010), where, heuristics similar to squeaky wheel optimization, was used to obtain a feasible timetabling solution. The solution was further enhanced with the aid of threshold accepting criteria from simulated annealing, where the computation results showed that the algorithm was good for the problem. In another research, the CB-CTT was considered as a multi-objective optimisation problem by Geiger (2009). The author presented a solution framework based on local search heuristics, using two different aggregation techniques (i.e., a weighted sum aggregation and a reference point-based method). The experimental results showed that the approach was able to obtain good solutions. Application of a dynamic tabu search to curriculum-based course timetabling was considered by De Cesco et al. (2008). In their work, the adjustment of hard and soft constraints was carried out using a short-term tabu exclusion with variable size tabu length and dynamic weight. The computation results showed a better performance of the technique for the problem. Clark et al. (2008) used quick repair-based search heuristic on Track 3 datasets in the ITC2007 timetabling competition and the method was able to produce comparable results. The performance of ABC with two neighbourhood structures was investigated on CB-CTT by Bolaji et al. (2011), where the ABC was run through two dependable phases; an initial phase using saturation degree (SD) and back-tracking algorithm to ensure feasible timetabling solution and the ABC was used in the second phase for the enhancement of the solution. However, the achieved results were not comparable with those available in the literature for the CB-CTT.

The CB-CTT problem formulations and descriptions have been discussed by McCollum et al. (2010), where pertinent issues and detailed overview of CB-CTT were presented with respect to ITC-2007.

CB-CTT Description

The CB-CTT is the scheduling of a set of lectures of courses to a set of rooms and periods on a weekly basis, in accordance with a given set of constraints. Once all lectures of courses have been assigned to periods and rooms with respect to the hard constraints (H1 - H4), then the timetable solution is said to be feasible. In addition, a feasible timetabling solution satisfying all hard constraints gives a penalty cost for the violations of the four soft constraints (S1 - S4). The main objective is to minimise the penalty of soft constraint violations in a feasible solution. The four hard and four soft constraints are outlined below:

Hard Constraints

H1. **Lectures:** All lectures of a course must be assigned to a distinct period and a room.
H2. **Room Occupancy:** Two lectures cannot be scheduled to the same room during the same period.
H3. **Conflicts:** Lectures of courses in the same curriculum taught by the same teacher must be scheduled to different periods.
H4. **Availability:** If the teacher of a course is not available at a given period, then no lectures of the course can be allocated to that period.

Soft Constraints

S1. **Room Capacity:** The number of students attending the course for each lecture must be less than or equal to the capacity of the rooms hosting the lectures
S2. **Room Stability:** All lectures of a particular course should be assigned to the same room; otherwise, the number of occupied rooms should be less.
S3. **Curriculum Compactness:** Lectures of courses belonging to the same curriculum should be in consecutive periods (i.e., adjacent to each other).
S4. **Minimum Working Days:** The lectures of each course should be spread across a given number of days.

CB-CTT Formulation

The formulation of CB-CTT (Table 1) is presented here in terms of assigning a set of courses to a set of periods and rooms subject to satisfying set constraints. Let $C = \{c_1, c_2, ..., c_k\}$ be a set of k courses, $R = \{r_1, r_2, ..., r_m\}$ be a set of m rooms, and $P = \{p_1, p_2, ..., p_w\}$ be a set of w periods, where period is the composition of minimum working days *(u)* and timeslots per working days *(v)* $(i.e.\ w = u \times v)$. Each course c_i consists of a set of lectures l_i to be assigned to different periods. The problem consists of a set of z curricula $Q = \{q_1, q_2, ..., q_z\}$ where each curriculum q_i is a group of courses having common students and finally, the CB-CTT consists of a set of students std_i and each student is assigned with a set of courses within the same curriculum.

In order to choose a suitable timetable representation, all courses in CB-CTT solutions are mapped into particular periods and rooms.

Table 1. The symbols used for CB-CTT problem

Symbols	Definition
K	The total number of courses.
M	The total number of rooms.
U	The total number of minimum working days per week.
V	The total number of timeslots per days.
W	The total number of periods, $w = u \times v$.
Z	The total number of curricula.
C	Set of the courses, $C = \left\{c_1, c_2, \ldots, c_k\right\}$.
R	Set of the rooms, $R = \left\{r_1, r_2, \ldots, r_m\right\}$.
P	Set of the periods, $P = \left\{p_1, p_2, \ldots, p_w\right\}$
Q	Set of the curricula, $Q = \left\{q_1, q_2, \ldots, q_z\right\}$
q_j	The curriculum j including a set of courses
l_i	The total number of lectures c_i .
nl	The total number of all lectures $\sum_{i=1}^{k} l_i$.
std_i	The number of students attending course c_i.
tc_i	is the teacher taking course c_i.
mw_i	The number of minimum working days of course c_i.
cpr_k	is the capacity of room r_k
$a_{i,j}$	Unavailable constraints matrix: whether course c_i available at period j. $\begin{cases} 0; & \text{if course } i \text{ is available} \\ 1; & \text{otherwise} \end{cases}$

continued on the following page

Table 1. Continued

Symbols	Definition
$b_{i,j}$	The Conflict matrix element: whether course c_i and c_j are in conflict. $\begin{cases} 0; & \text{if } c_i \text{ and } c_j \text{ are not in conflict} \\ 1; & \text{otherwise} \end{cases}$
$d_{i,j}$	Teacher Course Matrix: whether teacher tc_i teaches course c_i . $\begin{cases} 0; & \text{if teacher } tc_i \text{ is not teaches course } c_i \\ 1; & \text{otherwise} \end{cases}$
$e_{i,j}$	Course Room Capacity Matrix: whether the capacity of course c_i is less than the capacity of room r_j . $\begin{cases} 0; & \text{if } r_j \geq c_i \\ 1; & \text{otherwise} \end{cases}$
$tnr_i(X)$	The total number of rooms utilised by course c_i for timetable solution X
$nd_i(X)$	The total number of days used by course c_i in timetable solution X
$q_{s,i}(X)$	$\begin{cases} 0; & \text{if a lecture of any course in } q_s \\ & \text{takes place at } w_i \text{ for timetable solution } X \\ 1; & \text{otherwise} \end{cases}$
$x_{j,i}$	The lecture of course that takes place in the period *j* and room *i*
$cq_{a,s}$	Course Curricula Matrix: whether course c_m belong to the curricula q_s $\begin{cases} 1; & \text{if } c_a \in q_s \\ 0; & \text{otherwise} \end{cases}$

The representation is in the form of a matrix **X** with *n* rows and *h* columns, where *n* and *h* refers to the number of periods and the number of rooms, respectively. The value $x_{j,i}$ is the lecture that takes place in the period j and room i. The $x_{j,i}$ takes the value -1 if it is empty i.e., no lecture is placed in the position of the timetable corresponding to period *j* and room *i*. see Figure 1.

H_1: **Lectures** $\forall c_k \in C \quad \sum_{j \in k} \sum_{i \in m} \Im(x_{j,i} = l_d)$

The Boolean function $\Im$ return 1 if the l_d is in a position and 0 otherwise appear

H_2: **Room Occupancy**: This constraint is satisfied automatically by the solution representation.

H_3: **Conflicts**

$$\forall x_{j,i}, x_{j,i} \in X, x_{j,i} = c_a, x_{j,l} = c_g, \quad b_{a,g} = 0$$

H_4: **Availability**: $\forall c_a = x_{j,i} \in X, a_{k,i} = 0$

S_1: **Room Capacity**:

$$\forall x_{j,i} \in X, x_{j,i} = c_a$$

$$\mathrm{S}_1\left(\mathrm{x}_{\mathrm{i,j}}\right) = \begin{cases} \beta_1 & \cdot(\mathrm{std}_\mathrm{m} - \mathrm{cpr}_\mathrm{j}), \\ & \text{if } \mathrm{cpr}_\mathrm{j} < \mathrm{std}_\mathrm{m}; \\ 0, & \text{otherwise} \end{cases}$$

Figure 1. Timetabling solution representation

$$X_{n,h} = \begin{bmatrix} x_1(1) & x_1(2) & \cdots & x_1(n) \\ x_2(1) & x_2(2) & \cdots & x_2(n) \\ \cdots & \cdots & \cdots & \cdots \\ x_h(1) & x_h(2) & \cdots & x_h(n) \end{bmatrix} \downarrow Room$$
$$\rightarrow Period$$

S_2: **Room Stability**:
$\forall c_i \in C \quad S_2(c_i) = \beta_2 \cdot (tnr_i(X) - 1)$

S_3: **Curriculum Compactness**
$\forall x_{j,i} \in X, x_{j,i} = c_a,\ S_3(x_{j,i}) = \beta_3 \cdot \sum_{q_s \in Q} cq_{a,s} \cdot \theta_{s,j}(X)$

Where

$$\theta_{s,j}(X) = \begin{cases} 1, & if(j\%v = 1 \vee q_{s-1}(X) = 0) \wedge \\ & (j\%v = 0 \vee q_{s+1}(X) = 0) \\ 0, & otherwise \end{cases}$$

Note that in S_3 the computation is restricted within the same day. $\theta_{s,j}(X) = 1$ indicate that there is no assigned course in the curriculum q_s is adjacent (before or after) to the timeslot $j\ \%v$ in the $[j/v]^{th}$ day. Particularly, curriculum q_s does not appear before (after) period p_j means that p_j is the first (last) timeslot of a working day or q_s does not appear at $p_{j-1}(p_{j+1})$

S_4: **Minimum Working Days**:

$$\forall c_i \in C \ S_4(c_i) = \begin{cases} \beta_4 \cdot (mwd_i - nd_i(X)), \\ \quad if \ mwd_i < nd_i(X); \\ 0, \ otherwise \end{cases}$$

This formulation is adapted from Lü and Hao (2010) with minor modifications. Note that $\beta_1, \beta_2, \beta_3, \beta_4$ are the penalty values for each soft constraint which are fixed in the problem formulation. The objective function $f(x)$ is represented as the summation of penalty values for the violation of four soft constraints, given as

$$f(x) = \begin{array}{l} \sum_{x_{i,j} \in X} S_1(x_{j,i}) + \sum_{c_i \in C} S_2(c_i) + \\ \sum_{x_{i,j} \in X} S_3(x_{j,i}) + \sum_{c_i \in C} S_4(c_i) \end{array} \tag{1}$$

The details of the CB-CTT problem could be found in their website.[1] The main characteristics of the 21 datasets comprising courses (C), total lectures (NL), rooms (R), period per day (P), days (D), curricula (Q) and minimum and maximum lecture per day per curriculum (MW) are shown in Table 2.

EXAMINATION TIMETABLING PROBLEM

The university examination timetabling problem could be defined as the assignment of events (i.e., a set of examinations) into limited number of time periods and rooms, subject to a set of constraints (Qu et al., 2009). The Examination timetabling problem could be either capacitated or uncapacitated and the major difference is that the capacitated examination timetabling considers hard constraints in relation to the rooms (i.e., the room capacity must be respected), while the uncapacitated examination timetabling considers the assignment of exams to timeslots, but neglects the room assignments. This paper focuses on uncapacitated examination timetabling problem (UETP), whose dataset was initially proposed by Carter et al. (1996). This dataset was defined with 13 problem instances reflecting real-world examination

Table 2. Characteristics of CB-CTT datasets

Instance	C	NL	R	P	D	Q	MML
Comp01	30	160	6	6	5	14	2-5
Comp02	82	283	16	5	5	70	2-4
Comp03	72	251	16	5	5	68	2-4
Comp04	79	286	18	5	5	57	2-4
Comp05	54	152	9	6	5	139	2-4
Comp06	108	361	18	5	5	70	2-4
Comp07	131	434	20	5	5	77	2-4
Comp08	86	324	18	5	5	61	2-4
Comp09	76	279	18	5	5	75	2-4
Comp10	115	370	18	5	5	67	2-4
Comp11	30	162	5	9	5	13	2-6
Comp12	88	218	11	6	6	150	2-4
Comp13	82	308	19	5	5	66	2-4
Comp14	85	275	17	5	5	60	2-4
Comp15	72	251	16	5	5	68	2-4
Comp16	108	366	20	5	5	71	2-4
Comp17	99	339	17	5	5	70	2-4
Comp18	47	138	9	6	6	52	2-3
Comp19	74	277	16	5	5	66	2-4
Comp20	121	390	19	5	5	78	2-4
Comp21	94	327	18	5	5	78	2-4

timetabling problems. For the purposes of the present study, we refer to the 'Carter dataset.' This problem has been studied using a variety of techniques by the researchers in the domain of operation research and artificial intelligence. Here, we shall review the population and hybrid based techniques that we used as comparative methods in this paper. A multi-objective evolutionary algorithm for UETP was presented by Côté et al. (2005) with the aim of reducing the length of timetable and to space out conflicting exams as much as possible. The authors used local search based methods (tabu search and variable neighbourhood descent) to replace the recombination operators of EA (i.e., crossover and mutation operators). The result obtained by the method is competitive when compared with the other techniques in the literature. The application of max-min ant system (MMAS) to the uncapacitated version of the examination timetabling problem was proposed by Eley (2006). In their study, ants are used to generate feasible timetables and the hill-climbing optimizer is applied to the best timetable generated during each cycle in order to reduce the cost of the timetable. The use of this list was found to reduce the computational time and improve the quality of timetables. The hybridization of variable neighbourhood search with genetic algorithm for solving the exam timetabling problems was presented by Burke et al. (2010). They intelligently used GA to choose a subset of neighbourhoods during the search process. The hybrid method was able to generate good results on some of the Carter benchmark datasets. Al-Betar et al. (2010) investigated the performance of three selection techniques in memory consideration operator of harmony search algorithm (HSA) for examination timetabling problem. The three selection strategies used in their study are random, global best and roulette wheel. HSA technique was evaluated on Carter dataset and compared with other existing techniques; it showed the competitiveness of HSA techniques. Turabieh and Abdullah (2011) proposed a hybrid approach that incorporated electromagnetic-like mechanism with a great deluge algorithm for the examination timetabling problem. The objective of their study is to move sample points towards a high quality solution and avoidance of local minima by using a calculated force value. The hybrid technique is effective when compared with others from the literature.

Uncapacitated Examination Timetabling Problem (UETP) Description

The problem is tackled by scheduling a set of exams, each taken by a set of students, to a set of time periods (timeslots) subject to hard and soft constraints. The main objective is to obtain a timetable which satisfies the hard constraint (H1) with the least violation in the penalty of soft constraint (S1) where,

H_1: no student can sit for two exams simultaneously.

S_1: the exams taken by the same student should be spread out evenly across a timetable.

The detailed description of the problem were summarized by Qu et al. (2009). The notation for the UETP formulation is given in Table 3. A timetabling solution is represented by the vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_M)$ of exams, where x_i is the timeslot, $t \in T$, for exam, $i \in E$. The proximity cost function is used in the evaluation of the timetable and refers to the ratio of the penalty assigned to the total number of soft constraint violations and the total number of students. The formulation for the proximity cost function is given in (2)

$$\min \ f(x) = \frac{1}{N} \times \sum_{i-1}^{M-1} \sum_{j=1+1}^{M} c_{i,j} \times a_{i,j} \qquad (2)$$

H1: No student can sit for two exams simultaneously.

$$x_i \neq x_j \quad \forall x_i, x_j \in x \wedge c_{i,j} \geq 1$$

Note that: the value of the proximity cost function $f(x)$ is referred to as the object cost of a feasible timetable (Carter et al., 1996).

The Carter dataset consists of 13 datasets, which reflect the real-world examination timetabling problems. In the present study, 12 datasets circulated in the literature were used. The characteristics of Carter datasets vary in size and complexity are shown in Table 4. The conflict matrix in the last column illustrates density which is the ratio between the number of elements of values $ci,j > 0$ and the total number of elements in the conflict matrix (Qu et al., 2009).

ARTIFICIAL BEE COLONY ALGORITHM

This section describes the proposed ABC algorithm. Firstly, a brief description of the ABC is provided, while the next part explains the process of optimizing the proposed ABC for the timetabling problem.

Fundamentals of Artificial Bee Colony

The Artificial Bee Colony (ABC) algorithm is a nature inspired meta-heuristic algorithm which was proposed by Karaboga (2005) and Karaboga and Basturk (2007) for optimizing numerical problems. It was motivated by the intelligent foraging behavior of honey bees. The algorithm is particularly based on the model proposed in Teodorović and Dell'Orco (2005) for the foraging manners of honey bee colonies.

Table 3. The symbols used for UETP problem

Symbols	Definition
M	The total number of exams.
N	The total number of students.
P	The total number of time periods.
E	Set of exams $E = \{1,2,...,M\}$.
S	Set of students $S = \{1,2,...,N\}$.
T	Set of time periods $T = \{1,2,...,P\}$.
x	A timetable solution is given by $x_1, x_2, ..., x_M)$.
x_i	The timeslot of exam *i*.
$a_{i,j}$	Proximity coefficient matrix element: whether the timetable ***x*** is penalized based on the distance between time period of exam *i* and time period of exam *j*. $a_{i,j} = \begin{cases} 2^{5-\lvert x_i - x_j \rvert} & if 1 \le \lvert x_i - x_j \rvert \le 5 \\ 0 & otherwise \end{cases}$
$u_{i,j}$	Student-exam matrix element: whether student s_i is sitting for exam *j*. $u_{i,j} = \begin{cases} 1 & if\ student\ i\ is\ sitting\ for\ exam\ j \\ 0 & otherwise \end{cases}$
$c_{i,j}$	Conflict matrix element: total number of students sharing exam *i* and exam *j* $c_{i,j} = \sum_{k=1}^{N} u_{k,i} \times u_{k,j} \quad \forall i, j \in E.$

The model consists of three vital components: employed, and unemployed foraging bees, and food sources. The first two components i.e., employed and unemployed forager search for rich food sources. The two principal modes of behaviour which are necessary for self-organization and collective intelligence are also described by the model. In practice such mode includes the recruitment of foragers to the rich food sources resulting in positive feedback and abandonment of poor food sources by foragers causing negative feedback (Karaboga, 2005).

In the colony of ABC, there are three groups of bees: employed, onlooker and scout bees. Associated with particular food source is the employed bee whose behaviour is studied by the onlookers to select the desired food source while the scout bees searches for new food sources randomly once it is abandoned. Both onlookers and scouts are considered as unemployed foragers. The position of a food source in ABC corresponds to the possible solution of the problem to be optimized and the nectar amount of a food source represents the fitness (quality) of the associated solution. The number of employed bees is equal to the number of food sources (solutions), since each employed bee is associated with one and only one food source (Karaboga, 2005). The algorithm has the ability to employ fewer control parameters

Table 4. Characteristics of UETP datasets

Problem Instance	Institution	Time-Periods	Exams	Students	Density
CAR-S-91-I	Carleton University, Ottawa	35	682	16,925	0.13
CAR-F-92-I	Carleton University, Ottawa	32	543	18,419	0.14
EAR-F-83-I	Earl Haig Collegiate Institute, Toronto	24	190	1125	0.27
HEC-S-92-I	Ecole des Hautes Etudes Commercial	18	81	2823	0.42
KFU-S-93	King Fahd University, Dharan	20	461	5349	0.06
LSE-F-91	London School of Economics	18	381	2726	0.06
RYE-S-93	Ryeson University, Toronto	23	481	11,483	0.07
STA-F-83-I	St. Andrew's Junior High Sch. Toronto	13	139	611	0.14
TRE-S-92	Trent University Peterborough, Ontario	23	261	4360	0.18
UTA-S-92-I	Faculty of Arts and Sciences, University of Toronto.	35	622	21,266	0.13
UTE-S-92	Faculty of Engineering, University of Toronto	10	184	2750	0.08
YOR-F-83-I	York Mills Collegiate Institute, Toronto	21	181	941	0.29

such as population size (SN), limit, maximum cycle number (MCN) and can compete well with other population-based algorithms (Karaboga & Basturk, 2007).

The Key Phases of the ABC Algorithm

The key phases of the algorithm as proposed by Karaboga and Basturk (2007) are as follows:

Generate the initial population of the food sources randomly.

REPEAT

- Send the employed bees onto the food sources and calculate the fitness cost
- Evaluate the probability values for the food sources
- Send the onlooker bees onto the food sources depending on probability and calculate the fitness cost.
- Abandon the exploitation process if the sources are exhausted by the bees
- Send the scouts into the search area for discovering new food sources randomly
- Memorize the best food source found so far

UNTIL (requirements are met)

NEIGHBOURHOOD SEARCH

The neighbourhood search begins with initial feasible and complete solution generated using a saturation degree and backtracking algorithm. The components of ABC, such as employed and onlooker bees, use neighbourhood structures to explore different solution space thoroughly to enhance the quality of the solution. The idea of using three different neighbourhood structures is to reduce the redundancy or ineffectiveness of using a particular type alone. For example, the search space which can easily be reached by swap may be difficult for move to be accessed. Two neighbourhood structures are employed for CB-CTT while three are used for UETP. The three neighbourhood searches incorporated to the component of ABC to tackle timetabling problems are: move, swap, and kempe chain; denoted by NL-move, NL-swap, NL-kempe respectively as used by Thompson and Dowsland (1998) and Al-Betar, Khader, and Nadi (2010).

1. NL-Move: moves selected events to a feasible period and room randomly i.e., replace the time period $x_i^{'}$ of event i by another feasible timeslot
2. NL-Swap: swap two selected courses at random i.e., select event i and event j randomly, swap their time periods ($x_i^{'}, x_j^{'}$).
3. NL-Kempe: Firstly, select the timeslot $x_i^{'}$ of event i and randomly select another timeslot p'. Secondly, All exams that have the same timeslot $x_i^{'}$ and conflict with one or more exams timetabled in p' are entered to chain α where
   $\alpha = \left\{ j \middle| x_j^{'} = x_i^{'} \wedge t_{i,p'} = 0 \wedge \forall j \in E \right\}$,
   Thirdly, All exams that have the same timeslot p' and conflict with one or more exams timetabled in $x_i^{'}$ are entered to a chain α' where
   $\alpha' = \left\{ k \middle| x_k^{'} = p' \wedge t_{k,x_i^{'}} = 0 \wedge \forall k \in E \right\}$
   and Lastly, simply, assign the exams in α to p' and the exams in α' to $x_i^{'}$.

Artificial Bee Colony Procedure for ETP

Step 1. Initialization of the ABC and timetabling problem parameters: The timetabling (i.e., CB-CTT and UETP) parameters such as the set of events, the set of rooms, the set of periods and the set of constraints are extracted from the problem instances. The main decision variable of timetabling is the events which can be assigned to a feasible location in the timetable solution. The objective functions described in Eq. (1) and (2) are used to evaluate each solution during the ABC process which includes the functions of the four soft constraints for CB-CTT and one soft constraint for UETP. In addition, the three control parameters of ABC that are required to solve the problems are also initialized. These parameters are:

a. Population size (SN): the number of food sources (or solutions) in the population. SN is equal to the number of the employed bees or the onlooker bees.
b. Maximum Cycle Number (MCN): refers to the maximum number of generations.
c. Limit: it is used to diversify the search which determines the number of allowable generations for each non-improved food source to be abandoned.

Step 2. Initialize the Food Source Memory (FSM): The Food Source Memory (FSM) is an augmented matrix of size $SN \times N$, comprising in each row a vector representing a timetable solution as in (2). Note that the vectors in FSM are generated with a method that combines the saturation degree (SD) and backtracking algorithms. SD starts with an empty timetable, where the event (course or exam) with the least number of valid periods in the timetable is scheduled first. Then, the next selected event to be scheduled is based on the number of available periods. Wherever SD fails to schedule all events into the periods and rooms because of earlier assignments, then the backtracking algorithm (BA) is applied to re-assign unscheduled events. This technique used to guarantees that all the solutions are feasible. In addition, they are sorted in ascending order according to their objective function values.

$$\textbf{FSM}=\begin{bmatrix} x_1(1) & x_1(2) & \cdots & x_1(N) \\ x_2(1) & x_2(2) & \cdots & x_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ x_{SN}(1) & x_{SN}(2) & \cdots & x_{SN}(N) \end{bmatrix} \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{SN}) \end{bmatrix} \tag{3}$$

Step 3. Send the employed bees to the food sources: Here, the employed bees operator selects a timetable solution from the population one by one and applies the neighbourhood structures randomly to generate other solutions from the neighbourhoods. The fitness of the new solution is calculated and if it is better than that of previous solutions, then the best solution with higher fitness is memorised. This process is repeated iteratively until all solutions have been examined (see Figure 2 for details).

Step 4. Send the onlooker bees: The onlooker bee has the same number of food sources (timetable solution) as the employed bee. She initially calculates the selection probability of each food sources generated by the employed bee in the previous step. The fittest food source is selected by the onlooker using the roulette wheel selection mechanism. The process of selection in the onlooker phase works as follows:

i. assign for each employee bee a selection probability p_j as follows:

$$p_j = \frac{f(x_j)}{\sum_{k=1}^{SN} f(x_k)}$$

Note that the $\sum_{i=1}^{SN} p_i$ is unity

ii. the fitness of the food source of each employed bee is sampled by the onlooker bee and selects the one with the highest fitness based on its selection probability. She adjusted the selected food source to its neighbourhood using the same strategy as the employed bee. The fitness of the new solution is calculated and if it is better than the current one, then replace the current solution with the new one.

Step 5. Send the Scout to search for possible new food sources: This is known to be the colony explorer. It works once a solution is abandoned, i.e., if a solution in the memory has not improved for certain number of iterations as determined by the limit parameter then, the scout bee carries out a random search to replace the abandoned food source. Figure 3 show the detailed process of the scout bee where scout(i) is a vector of size (SN) which contains information related to the improvement of all the food sources.

Step 6. Memorize the best food source: Memorize the fitness of the best food source $\boldsymbol{x}^{best}$ found so far in FSM.

Step 7. Stopping Condition: Steps 3 through 6 are repeated until a stop criterion is met. This is originally determined using MCN value.

COMPUTATIONAL RESULTS AND DISCUSSIONS

The proposed ABC was tested on 21 datasets on CB-CTT and 12 datasets of examination timetabling problems (Carter datasets). The method is coded in Microsoft Visual C++ 6.0 on Windows Vista platform on Intel 2 GHz Core 2 Quad processor with 2 GB of RAM. The parameters chosen for each dataset are the same except in population size where 10

Figure 2. Employed bee phase

for $j = 1 \ldots SN$ **do**
  $rnd \leftarrow u(0,1)$
  if $rnd \geq 0.70$ **then**
    NL-swap($\boldsymbol{x}_j$)
  else if **rnd** ≥ 0.30 **then**
    NL-move($\boldsymbol{x}_j$)
  else
    NL-kempe($\boldsymbol{x}_j$)
  end if
end for

Figure 3. Scout bee phase

for *i = 1 . . . SN* **do**
  if (scout (i) = limit) **then**
    generate a new x_i using SA and BA
  end if
end for

and 100 are used for both UETP and CB-CTT see Table 5 for details. Each problem instance was run 5 times with a different random seed.

Curriculum-Based Course Timetabling Problem

The first series of experiments on CB-CTT is carried out based on ITC-2007 constraints: where all hard and soft constraints were considered. The best results obtained for 10000 iterations; out of five runs using two neighbourhood structures (i.e., move and swap) are presented. Table 6 shows the best, average results and standard deviation, while Table 7 shows the comparison of the final results of ABC in terms of penalty value to other five published methods and best known results in the literature. The best results produced by the ABC as shown in Table 7 are close to previous cited results, but not comparatively better. Notably as shown in Table 7, the Comp11 dataset, the result produced is far from the results produced by other comparative methods. This is because the proposed method cannot reach the optima or near optimal due to the lack of exploitation of the combinatorial search space.

Following is the comparative methods

- T1: The quick-fix repair-based timetable solver by Clark et al. (2008).
- T2: Application of the threshold accepting by Geiger (2008).
- T3: Dynamic Tabu search by De Cesco (2008).
- T4: The integer programming approach by Lach and Lubbecke (2010).
- T5: The constraint satisfaction problem by Atsuta et al. (2007).

Uncapacitated Examination Timetabling Problem

The second experiments on the uncapacitated examination timetabling problem are performed using Carter datasets. Again, the best results obtained for 10000 iterations; out of five runs using three neighbourhood structures (i.e., move, swap and kempe chain) within the process of ABC are presented. Table 8 shows the best, average results and standard deviation of ABC method for UETP. Table 9 shows the comparison of the final results of ABC in terms of penalty value to others 6 population-based and hybrid-population-based methods published in the literature while the last column of Table 9 shows the best known results. The number in these tables represents the penalty cost as calculated by (7). The numbers show the best penalty cost obtained for that corresponding instance (note that the lowest is best). As shown in Table 9, the results of the proposed

Table 5. ABC parameter settings

ABC	Population Size (SN)	MCN	Limit
CB-CTT	100	10000	1000
UETP	10	10000	1000

Table 6. ABC results on CB-CTT problem

Problem	Best	Average	Worst	Std Dev.
Comp01	24	24.6	26	0.89
Comp02	299	300.8	304	1.92
Comp03	270	272.2	274	1.79
Comp04	166	166.8	168	0.84
Comp05	456	465.8	480	9.18
Comp06	255	258.2	262	2.86
Comp07	253	256.6	260	2.70
Comp08	173	178.6	182	3.65
Comp09	271	277.2	280	3.90
Comp10	239	241.6	245	2.79
Comp11	220	221.4	224	1.67
Comp12	751	763.8	780	10.96
Comp13	214	219.4	222	3.85
Comp14	221	225.0	230	3.24
Comp15	238	240.2	245	2.86
Comp16	236	241.0	248	5.39
Comp17	280	287.2	295	6.83
Comp18	173	177.2	181	3.35
Comp19	276	277.0	279	1.22
Comp20	241	242.8	245	1.79
Comp21	364	365.6	368	1.67

ABC on uncapacitated examination timetabling problem is comparable with other methods in the literature.

Following is the comparative methods:

- T6: Harmony Search Algorithm by Al-Betar et al. (2010).
- T7: Ant Algorithms by Eley (2006).
- T8: Multi-Objective Evolutionary Algorithm by Côté et al. (2010).
- T9: An integrated hybrid approach by Turabieh and Abdullah (2011).
- T10: Hybrid Variable Neighbourhood Search with Genetic Algorithm by Burke et al. (2010).
- T11: ABC search algorithm by Alzaqebah and Abdullah (2011a)

CONCLUSION AND POSSIBLE FUTURE DIRECTIONS

This paper presents an Artificial Bee Colony to solve the educational timetabling problems, where neighbourhood structures are incorporated within the components of ABC. In order to test the performance of our technique, experiments are carried out using CB-CTT where two neighbourhood structures are incorporated and three are integrated for the UETP. The proposed technique enhanced the solution quality further by means of refining the search space region of the new solution in each cycle. The performance of ABC on UETP is comparable with other population-based techniques in the literature and it yields a good solution for CB-CTT problem.

Table 7. Comparison results of ABC with others on CB-CTT

Problem	ABC Result	T1	T2	T3	T4	T5	Best Known
Comp01	24	9	5	5	13	5	5
Comp02	299	103	108	75	43	50	29
Comp03	270	101	115	93	76	82	66
Comp04	166	55	67	45	38	35	35
Comp05	456	370	408	326	314	312	292
Comp06	255	112	94	62	41	69	28
Comp07	253	97	56	38	19	42	6
Comp08	173	72	75	50	43	40	38
Comp09	271	132	153	119	102	110	96
Comp10	239	74	66	27	14	27	4
Comp11	220	1	0	0	0	0	0
Comp12	751	393	430	358	405	351	310
Comp13	214	97	101	77	68	68	59
Comp14	221	87	88	59	54	59	51
Comp15	238	119	128	87	-	82	68
Comp16	236	84	81	47	-	40	22
Comp17	280	152	124	86	-	102	60
Comp18	173	110	116	71	-	68	65
Comp19	276	111	107	74	-	75	54
Comp20	241	144	88	54	-	61	4
Comp21	364	169	174	117	-	123	86

However, these are not comparable with the best known solutions obtained in the literature.

It is highly recommendable that future work be directed:

- To improve ABC for ETP by introducing more advanced neighborhood structures into its components.
- To integrate ABC with other metaheuristic algorithms like hill-climbing and simulated annealing acceptance rule in step 3 of ABC.
- To evaluate effect of neighbourhood structures on ABC for ETP.
- To apply ABC for different timetable formulations such as nurse rostering, railway timetabling problem, etc.

ACKNOWLEDGMENT

This research was partially supported by the Universiti Sains Malaysia Postgraduate Fellowship Scheme 2011, awarded to the first author. We would like also to thank the anonymous referees for their insightful comments.

REFERENCES

Abdullah, S., Burke, E. K., & McCollum, B. (2007). A hybrid evolutionary approach to the university course timetabling problem. In *Proceedings of the Congress on Evolutionary Computation* (pp. 1764-1768).

Table 8. ABC results on UETP

Problem instance	Best	Average	Worst	Std Dev.
CAR-S-91-I	5.25	5.42	5.48	0.10
CAR-F-92-I	4.39	4.44	4.47	0.03
EAR-F-83-I	35.22	35.36	35.50	0.39
HEC-S-92-I	10.71	10.85	10.95	0.09
KFU-S-93	14.13	14.25	14.34	0.09
LSE-F-91	11.64	11.67	11.74	0.04
RYE-S-93	9.34	9.41	9.47	0.05
STA-F-83-I	157.08	157.14	157.19	0.04
TRE-S-92	8.58	8.63	8.67	0.03
UTA-S-92-I	3.56	3.58	3.62	0.02
UTE-S-92	26.12	26.66	26.62	0.22
YOR-F-83-I	37.39	37.72	37.98	0.10

Table 9. ABC results with other comparative methods

Problem	ABC Result	T6	T7	T8	T9	T10	T11	Best Known
CAR-S-91-I	5.25	4.99	5.4	5.2	4.80	4.6	5.86	4.50
CAR-F-92-I	4.39	4.29	4.2	4.3	4.10	3.9	4.92	3.90
EAR-F-83-I	35.22	34.42	34.2	36.8	34.92	32.8	38.34	29.3
HEC-S-92-I	10.71	10.40	10.4	11.1	10.73	10.0	11.51	9.2
KFU-S-93	14.13	13.5	14.3	14.5	13.00	13.0	16.04	13.0
LSE-F-91	11.64	10.48	11.3	11.3	10.01	10.0	12.42	9.6
RYE-S-93	9.34	8.79	8.8	9.8	9.65	-	10.73	6.8
STA-F-83-I	157.08	157.04	158.03	157.3	158.26	156.9	158.01	156.9
TRE-S-92	8.58	8.16	8.6	8.6	7.88	7.9	9.58	7.88
UTA-S-92-I	3.56	3.43	3.5	3.5	3.20	3.2	3.99	3.14
UTE-S-92	26.12	25.09	25.3	26.4	26.11	24.8	27.80	24.8
YOR-F-83-I	37.39	35.86	36.4	39.4	36.22	34.9	41.44	34.9

Abdullah, S., Shaker, K., McCollum, B., & McMullan, P. (2010). Dual sequence simulated annealing with round-robin approach for university course timetabling. In P. Cowling & P. Mertz (Eds.), *Proceedings of the Evolutionary Computation in Combinatorial Optimization* (LNCS 6022, pp. 1-10).

Abdullah, S., Turabieh, H., McCollum, B., & Burke, E. K. (2010). An investigation of a genetic algorithm and sequential local search approach for curriculum-based course timetabling problems. In *Proceedings of the 4th Multidisciplinary Conference on Scheduling: Theories and Applications* (pp. 727-731).

Al-Betar, M. A., Khader, A., & Thomas, J. (2010). A combination of metaheuristic components based on harmony search for the uncapacitated examination timetabling. In *Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling*, Belfast, Northern Ireland (pp. 57-80).

Al-Betar, M. A., & Khader, A. T. (2008). A harmony search algorithm for university course timetabling. *Annals of Operations Research*, *194*(1), 1–29.

Al-Betar, M. A., Khader, A. T., & Nadi, F. (2010). Selection mechanisms in memory consideration for examination timetabling with harmony search. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation* (pp. 1203-1210).

Alzaqebah, M., & Abdullah, S. (2011a). Artificial bee colony search algorithm for examination timetabling problems. *International Journal of the Physical Sciences*, *6*(17), 4264–4272.

Alzaqebah, M., & Abdullah, S. (2011b). Hybrid artificial bee colony search algorithm based on disruptive selection for examination timetabling problems. In *Proceedings of the 5th International Conference on Combinatorial Optimization and Applications* (pp. 31-45).

Bolaji, A. L., Khader, A. T., Al-Betar, M. A., & Awadallah, M. A. (2011). An improved artificial bee colony for course timetabling. In *Proceedings of the Sixth International Conference on Bio-Inspired Computing: Theories and Applications* (pp. 9-14).

Burke, E., & Bykov, Y. (2006). Solving exam timetabling problems with the flex-deluge algorithm. In *Proceedings of the 6th International Conference on Practice and Theory of Automated Timetabling*, Brno, Czech Republic (pp. 370-372).

Burke, E., Bykov, Y., Newall, J., & Petrovic, S. (2004). A time-predefined local search approach to exam timetabling problems. *IIE Transactions*, *36*(6), 509–528. doi:10.1080/07408170490438410

Burke, E., Dror, M., Petrovic, S., & Qu, R. (2005). Hybrid graph heuristics within a hyper-heuristic approach to exam timetabling problems. In Golden, B., Raghavan, S., & Wasil, E. (Eds.), *The next wave in computing, optimization, and decision technologies* (pp. 79–91). Berlin, Germany: Springer-Verlag. doi:10.1007/0-387-23529-9_6

Burke, E. K., Eckersley, A. J., McCollum, B., Petrovic, S., & Qu, R. (2010). Hybrid variable neighbourhood approaches to university exam timetabling. *European Journal of Operational Research*, *206*(1), 46–53. doi:10.1016/j.ejor.2010.01.044

Burke, E. K., Mareček, J., Parkes, A. J., & Rudová, H. (2008). A branch-and-cut procedure for the Udine course timetabling problem. *Annals of Operations Research*, 1–17.

Burke, E. K., & Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, *140*(2), 266–280. doi:10.1016/S0377-2217(02)00069-3

Cambazard, H., Hebrard, E., O'Sullivan, B., & Papadopoulos, A. (2012). Local search and constraint programming for the post enrolment-based course timetabling problem. *Annals of Operations Research*, *194*(1), 111–135. doi:10.1007/s10479-010-0737-7

Carter, M. W., Laporte, G., & Lee, S. Y. (1996). Examination timetabling: Algorithmic strategies and applications. *The Journal of the Operational Research Society*, *47*(3), 373–383.

Clark, M., Henz, M., & Love, B. (2008). QuikFix. A repair-based timetable solver. In *Proceedings of the Seventh Practice and Theory of Automated Timetabling Conference*.

Côté, P., Wong, T., & Sabourin, R. (2005). A hybrid multi-objective evolutionary algorithm for the uncapacitated exam proximity problem. In *Proceedings of the 5th Practice and Theory of Automated Timetabling Conference* (pp. 294-312).

De Causmaecker, P., Demeester, P., & Vanden Berghe, G. (2009). A decomposed metaheuristic approach for a real-world university timetabling problem. *European Journal of Operational Research*, *195*(1), 307–318. doi:10.1016/j.ejor.2008.01.043

De Cesco, F., Di Gaspero, L., & Schaerf, A. (2008). Benchmarking curriculum-based course timetabling: Formulations, data formats, instances, validation, and results. In *Proceedings of the 7th Practice and Theory of Automated Timetabling Conference*.

Di Gaspero, L., & Schaerf, A. (2006). Neighborhood portfolio approach for local search applied to timetabling problems. *Journal of Mathematical Modelling and Algorithms*, *5*(1), 65–89. doi:10.1007/s10852-005-9032-z

Eley, M. (2006). Ant algorithms for the exam timetabling problem. In *Proceedings of the 6th International Conference on Practice and Theory of Automated Timetabling* (pp. 364-382).

Geiger, M. (2009). Multi-criteria curriculum-based course timetabling - A comparison of a weighted sum and a reference point based approach. In M. Ehrgott, C. M. Fonseca, X. Gandibleux, J.-K. Hao, & M. Sevaux (Eds.), *Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization* (LNCS 5467, pp. 290-304).

Geiger, M. (2010). Applying the threshold accepting metaheuristic to curriculum based course timetabling. *Annals of Operations Research*, *194*(1), 1–14.

Goltz, H. J., & Matzke, D. (1998). University timetabling using constraint logic programming. In G. Gupta (Eds.), *Proceedings of the First International Workshop on Practical Aspects of Declarative Languages* (LNCS 1551, pp. 320-334).

Johnson, D. S., & Garey, M. R. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. San Francisco, CA: Freeman.

Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization* (Tech. Rep. No. TR06). Erciyes, Turkey: Erciyes University Press.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, *39*(3), 459–471. doi:10.1007/s10898-007-9149-x

Lach, G., & Lübbecke, M. E. (2010). Curriculum based course timetabling: New solutions to Udine benchmark instances. *Annals of Operations Research*, *194*(1), 255–272. doi:10.1007/s10479-010-0700-7

Landa-Silva, D., & Obit, J. (2009). Evolutionary non-linear great deluge for university course timetabling. In *Proceedings of the 4th International Conference on Hybrid Artificial Intelligence Systems* (pp. 269-276).

Lewis, R. (2008). A survey of metaheuristic-based techniques for university timetabling problems. *OR-Spektrum*, *30*(1), 167–190. doi:10.1007/s00291-007-0097-0

Lü, Z., & Hao, J. K. (2010). Adaptive tabu search for course timetabling. *European Journal of Operational Research*, *200*(1), 235–244. doi:10.1016/j.ejor.2008.12.007

Lü, Z., Hao, J. K., & Glover, F. (2011). Neighborhood analysis: A case study on curriculum-based course timetabling. *Journal of Heuristics*, *17*(2), 97–118. doi:10.1007/s10732-010-9128-0

Malim, M. R., Khader, A. T., & Mustafa, A. (2006). Artificial immune algorithms for university timetabling. In *Proceedings of the 6th International Conference on Practice and Theory of Automated Timetabling*, Brno, Czech Republic.

McCollum, B. (2010). *A simulated annealing hyper-heuristic for university course timetabling problem*. Retrieved from http://www.google.com/url?sa=t&rct=j&q=a%20simulated%20annealing%20hyper-heuristic%20for%20university%20course%20timetabling%20problem&source=web&cd=1&ved=0CDUQFjAA&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.66.2203%26rep%3Drep1%26type%3Dpdf&ei=BUWFUMrWBubI0AGDzIGABQ&usg=AFQjCNEkgr-DmTGPLfJ_W0K__hX0lfFhpQ

McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., & Parkes, A. J. (2010). Setting the research agenda in automated timetabling: The second international timetabling competition. *INFORMS Journal on Computing*, *22*(1), 120–130. doi:10.1287/ijoc.1090.0320

Müller, T. (2009). ITC2007 solver description: A hybrid approach. *Annals of Operations Research*, *172*(1), 429–446. doi:10.1007/s10479-009-0644-y

Nothegger, C., Mayer, A., Chwatal, A., & Raidl, G. R. (2012). Solving the post enrolment course timetabling problem by ant colony optimization. *Annals of Operations Research*, *194*(1), 1–15. doi:10.1007/s10479-012-1078-5

Qu, R., Burke, E., McCollum, B., Merlot, L. T. G., & Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*, *12*(1), 55–89. doi:10.1007/s10951-008-0077-5

Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review*, *13*(2), 87–127. doi:10.1023/A:1006576209967

Shaker, K., & Abdullah, S. (2009). Incorporating great deluge approach with kempe chain neighbourhood structure for curriculum-based course timetabling problems. In *Proceedings of the 2nd Conference on Data Mining and Optimization* (pp. 149-153).

Sheau Fen Ho, I., Safaai, D., & Hashim, S. (2009). A study on PSO-based university course timetabling problem. In *Proceedings of the International Conference on Advanced Computer Control* (pp. 648-651).

Socha, K., Sampels, M., & Manfrin, M. (2003). Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. In S. Cagnoni, C. G. Johnson, J. J. R. Cardalda, E. Marchiori, D. W. Corne, J.-A. Meyer et al. (Eds.), *Proceedings of the EvoWorkshops on Applications of Evolutionary Computing* (LNCS 2611, pp. 334-345).

Teodorović, D., & Dell'Orco, M. (2005). Bee colony optimization–A cooperative learning approach to complex transportation problems. In *Proceedings of the 10th Meeting of the EURO Working Group in Advanced OR and AI Methods in Transportation*, Poznan, Poland (pp. 51-60).

Thanh, N. D. (2007). Solving timetabling problem using genetic and heuristic algorithms. In *Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing* (pp. 472-477).

Thompson, J. M., & Dowsland, K. A. (1998). A robust simulated annealing based examination timetabling system. *Computers & Operations Research*, *25*(7-8), 637–648. doi:10.1016/S0305-0548(97)00101-9

Turabieh, H., & Abdullah, S. (2011). An integrated hybrid approach to the examination timetabling problem. *Omega*, *39*(6), 598–607. doi:10.1016/j.omega.2010.12.005

White, G., & Xie, B. (2001). Examination timetables and tabu search with longer-term memory. In E. Burke & W. Erben (Eds.), *Proceedings of the 3rd International Conference on Practice and Theory of Automated Timetabling* (LNCS 2079, pp. 85-103).

White, G. M., Xie, B. S., & Zonjic, S. (2004). Using tabu search with longer-term memory and relaxation to create examination timetables. *European Journal of Operational Research*, *153*(1), 80–91. doi:10.1016/S0377-2217(03)00100-0

Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications* (pp. 169-178).

Asaju La'aro Bolaji received the BTech in Physics from Federal University of Technology, Minna, Nigeria and MSc in Mathematics from University of Ilorin, Nigeria in 2001 and 2006, respectively. He is currently working toward the PhD degree from the School of Computer Sciences, Universiti Sains Malaysia. His current research interests include evolutionary algorithms, nature-inspired computing, and their applications for scheduling problems.

Ahamad Tajudin Khader (M'09) received the BSc and MSc degrees in mathematics from Ohio University, Athens, in 1982 and 1983, respectively, and the PhD degree in computer science from the University of Strathclyde, Glasgow, UK, in 1993. He is currently a Professor and Deputy Dean with the School of Computer Sciences, Universiti Sains Malaysia, Pulau Pinang, Malaysia. He has authored many publications in international journals, conferences, and book chapters. His research interest mainly focuses on optimization, scheduling, and timetabling. In particular, he is interested in evolutionary and metaheuristic algorithms. He is attracted to search in general. Prof. Khader is a member of the Association of Computing Machinery, Special Interest Group for Genetic and Evolutionary Computation, and Computational Intelligence Society.

Mohammed Azmi Al-Betar received the BSc and MSc degrees from the Department of Computer Science, Yarmouk University, Irbid, Jordan, in 2001 and 2003, respectively, and the PhD degree from the School of Computer Sciences, University Sains Malaysia (USM), Pulau Pinang, Malaysia, in October 2010. He is currently an Assistant Professor with the Department of Computer Science, Jadara University, Irbid, Jordan, and a Postdoctoral Research Fellow with the School of Computer Sciences, USM. He has authored a number of publications in international journals, conferences, and book chapters. His research interests are mainly directed to metaheuristic optimization methods and hard combinatorial optimization problems including scheduling and timetabling.

Mohammed A. Awadallah received the BSc in Computer Sciences from Islamic University of Gaza, Palestine, and MSc in Computer Information System (CIS) from Arab Academy for Banking and Financial sciences, Amman, Jordan, in 2002 and 2005, respectively. He is currently working toward the PhD degree from the School of Computer Sciences, Universiti Sains Malaysia. His current research interests include evolutionary algorithms, nature-inspired computing, and their applications for Timetabling problems.