G Model
JOCS-272; No. of Pages 10

ARTICLE IN PRESS

Journal of Computational Science xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Journal of Computational Science

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

University course timetabling using hybridized artificial bee colony with hill climbing optimizer

Asaju La'aro Bolaji [a,b,*], Ahamad Tajudin Khader [a], Mohammed Azmi Al-Betar [a,c], Mohammed A. Awadallah [a]

[a] *School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia*
[b] *Department of Computer Science, University of Ilorin, Ilorin, Nigeria*
[c] *Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, P.O. Box 50, Al-Huson, Irbid, Jordan*

ARTICLE INFO

Article history:
Received 31 December 2012
Received in revised form 18 January 2014
Accepted 8 April 2014
Available online xxx

Keywords:
Artificial bee colony
Timetabling problem
Nature-inspired computing
Swarm intelligence
Hybrid metaheuristic

ABSTRACT

University course timetabling is concerned with assigning a set of courses to a set of rooms and timeslots according to a set of constraints. This problem has been tackled using metaheuristics techniques. Artificial bee colony (ABC) algorithm has been successfully used for tackling uncapaciated examination and course timetabling problems. In this paper, a novel hybrid ABC algorithm based on the integrated technique is proposed for tackling the university course timetabling problem. First of all, initial feasible solutions are generated using the combination of saturation degree (SD) and backtracking algorithm (BA). Secondly, a hill climbing optimizer is embedded within the employed bee operator to enhance the local exploitation ability of the original ABC algorithm while tackling the problem. Hill climbing iteratively navigates the search space of each population member in order to reach a local optima. The proposed hybrid ABC technique is evaluated using the dataset established by Socha including five small, five medium and one large problem instances. Empirical results on these problem instances validate the effectiveness and efficiency of the proposed algorithm. Our work also shows that a well-designed hybrid technique is a competitive alternative for addressing the university course timetabling problem.

1. Introduction

University course timetabling problem (UCTP) is one of the hardest problems faced by academic institutions throughout the world. The UCTP is known to be a difficult combinatorial optimization problem that has been widely studied over the last few decades. The problem involves scheduling a given number of courses to a limited number of periods and rooms subject to satisfying a set of given constraints. The constraints in UCTP are usually classified into hard and soft constraints. The satisfaction of hard constraints is compulsory for the timetabling solutions to be *feasible*, while the satisfaction of the soft constraints is required but not essential. The quality of the UCTP solution is normally determined by the cost of violating the soft constraints. Basically, there are two forms of UCTP: curriculum-based course (CB-CTT) and post-enrolment course timetabling (PE-CTT); the latter is the concern of this research.

The UCTP has been tackled with series of proposition of optimization techniques by the researchers across the fields of operation research and artificial intelligence over the past few years. The problem has been surveyed in [1,2] where metaheuristic-based techniques have shown to be the most frequently used. The metaheuristics can be categorize into local search-based and population-based techniques [3]. Some examples of local search-based techniques that tackled UCTP include tabu search [4], simulated annealing [5,6], great deluge [7,8], variable neighbourhood structures (VNS) [9]. On the other hand, examples of population-based techniques used for UCTP are ant colony optimization [10,11], genetic algorithm (GA) [12,13], harmony search algorithm (HSA) [14,15], particle swarm optimization [16]. Similarly, hyper-heuristic approaches and hybrid metaheuristic approaches have also been used recently to tackle course timetabling problems [17]. Other approaches applied to UCTP include early heuristic approaches derived from graph colouring heurstics [18], constraint-based techniques [19], case-based reasoning [20]. Although the success of the metaheuristic algorithms when applied independently to optimization problems have

* Corresponding author at: School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia. Tel.: +2348146820888.
E-mail addresses: abl10_sa0739@student.usm.my (A.L. Bolaji), tajudin@cs.usm.my (A.T. Khader), mohbetar@cs.usm.my (M.A. Al-Betar), mama10_com018@student.usm.my (M.A. Awadallah).

http://dx.doi.org/10.1016/j.jocs.2014.04.002
1877-7503/

Table 1
The PE-CTP constraints.

	Hard constraints		Soft constraints
H_1	A student should not be assigned to more than one event at a time	S_1	A student should not have a class in the last slot of the day
H_2	The capacity of the room should be big enough for both the students and the event	S_2	A student should not have more than two classes consecutively
H_3	Only one event takes place in each room at any timeslot	S_3	A student should not have a single class in one day

Table 2
The notation used in the PE-CTP formulations.

Symbols	Description
D	Total number of working days per week
H	Total number of timeslots for each working days
M	Total number of rooms
N	Total number of events
P	Total number of timeslots
V	Total number of Students
Z	Total number of features
$\mathbb{F}$	Set of features, $\mathbb{F}=\{z_0, z_1, \ldots, z_{z-1}\}$
$\mathbb{R}$	Set of rooms, $\mathbb{R}=\{r_0, r_1, \ldots, r_{M-1}\}$
$\mathbb{S}$	Set of students, $\mathbb{S}=\{s_0, s_1, \ldots, s_{v-1}\}$
$\mathbb{T}$	Set of timeslots, $\mathbb{T}=\{t_0, t_1, \ldots, t_{p-1}\}$
$\boldsymbol{x}$	The solution to the timetabling problem is give as $\boldsymbol{x}=(x_1, x_2, \ldots, x_N)$
b_i	The total of students taking event i
g_i	The capacity of room i
x_i	The room and timeslot of event i in the timetable
$a_{s,j}$	Student-day matrix element: whether student i assigned to only one event in day j $a_{s,j}=\begin{cases}1 & \sum_{i=0}^{H-1} sa_{s,i+j\times H}=1, \quad j\in[0, D-1].\\ 0 & \text{otherwise.}\end{cases}$
$c_{i,j}$	Conflict matrix element: whether events i and j are in conflict with each other $c_{i,j}=\begin{cases}1 & \text{if } u_{k,i}=1\wedge u_{k,j}=1 \quad \exists k\in\mathbb{S}.\\ 0 & \text{otherwise.}\end{cases}$
$q_{i,j}$	Event-room matrix element: whether events i and room r_j are compatible in terms of features and size $q_{i,j}=\begin{cases}1 & \text{if}(y_{j,k}\geq w_{i,k}\wedge(b_i\leq g_j) \quad \forall k\in\mathbb{F}.\\ 0 & \text{otherwise.}\end{cases}$
$sa_{j,k}$	Student availability matrix element: whether student s_j is in timeslot t_k in the timetable $\boldsymbol{x}$ $sa_{j,k}=\begin{cases}1 & \exists x_i\in\boldsymbol{x}, k=x_i \mod P\wedge u_{j,i}=1.\\ 0 & \text{otherwise.}\end{cases}$
$u_{i,j}$	Student-event matrix element: whether student s_i has attended an event j. $u_{i,j}=\begin{cases}1 & \text{if student } s_i \text{ attends event } j\\ 0 & \text{otherwise.}\end{cases}$
$w_{i,j}$	Event-feature matrix element: whether event i requires feature z_j. $w_{i,j}=\begin{cases}1 & \text{if event } i \text{ requires feature } z_j\\ 0 & \text{otherwise.}\end{cases}$
$y_{i,j}$	Room-feature matrix element: whether Room r_i contains feature z_j. $y_{i,j}=\begin{cases}1 & \text{if room } r_i \text{ includes feature } z_j\\ 0 & \text{otherwise.}\end{cases}$

been reported, substantial research efforts come from hybridization especially for large scheduling/timetabling problems. Few examples of hybridization between population-based and local search-based algorithms that were applied to timetabling problem appeared in [21,13,22]. Another hybrid technique which combined two population-based algorithms for real-world course timetabling (i.e. bee algorithm with harmony search algorithm) can be found in [23].

Artificial bee colony (ABC) algorithm is a nature-inspired search mechanism introduced by Karaboga in [24], based on concepts of simulating intelligence foraging behaviour of honey bee. Recently, researches and applications of ABC have been rapidly proposed and widely utilized. The methodology of ABC has been used to solve the complex problem in the field of science and engineering. Its introduction has brought a significant impact on many growing fields, such as artificial intelligence, decision analysis, knowledge management, image processing, pattern matching and recognition, product process design, scheduling and timetabling, and stock market analysis [25,26]. ABC was tailored in our previous work, for both curriculum-based course timetabling and uncapacitated examination timetabling problems with success stories [27–29]. To validate the success of the ABC in other scheduling/timetabling problems, in this paper, PE-CTP is being tackled with a hybridized artificial bee colony with hill climbing optimizer. This hybridization involves the combination of hill climbing optimizer (HCO) with the employed bee phase of original ABC. The main aim of this hybridization is to enhance the exploitation power of ABC by searching the PE-CTP solution space rigourously in order to obtain good solution.

The remainder of this paper is organized as follows: Section 2 gives the background descriptions and formulations of PE-CTP. Section 3 provides an overview of the original ABC, and Section 4 describes the proposed hybridization of HCO with ABC algorithm. Section 5 presents experimental analysis and discussions of the results of the proposed hybridized ABC, and compares the results with 24 other techniques using the Socha dataset. Section 6 presents the conclusion and future directions.

2. Post-enrolment-course timetabling problems

2.1. Problem descriptions

The descriptions of PE-CTP is given as an assignment of a set of courses to a set given timeslots and rooms in accordance with a set of given hard and soft constraints. The problem consists of three hard constraints which *must* be satisfied in order to obtain *feasible* solution i.e. (H_1, H_2, H_3) and three soft constraints (S_1, S_2, S_3) as shown in Table 1. The minimization of the soft constraints in a feasible timetable is the basic objective of PE-CTP.

2.2. Problem formulations

The PE-CTP consists of N events, each of which is attended by a number of students and demand specific room features; a set of M rooms, $\mathbb{R}=\{r_0, r_1, \ldots, r_{M-1}\}$ each with a specific capacity and features; a set of P timeslots[1] $\mathbb{T}=\{t_0, t_1, \ldots, t_{p-1}\}$, a set of Z room features, $\mathbb{F}=\{z_0, z_1, \ldots, z_{z-1}\}$, and a set of V students, $\mathbb{S}=\{s_0, s_1, \ldots, s_{v-1}\}$ each attending one or more events.

Table 2 shows the notation for the formulation of the PE-CTP. The solution to PE-CTP is given by the vector $x=(x_1, x_2, \ldots, x_N)$ of courses, where the value of x_i is a map of room r_j with timeslot t_k for course i:

$$x_i = j\times P + k \tag{1}$$

For instance, if $\boldsymbol{x}=(590,\ldots)$, then event 1 is map to room 13 and timeslot 5 (where $P=45$).

2.3. Definitions

The formal definitions of both hard and soft constraints are as follows:

Definition 1. The position l is feasible for event x_i to be assigned to the timetable $\boldsymbol{x}$ if and only if the following conditions are met:

[1] Generally, $P=45$ (i.e. $D=5$ working days, each with $H=9$ timeslots).