Computers & Operations Research 54 (2015) 142–154

Contents lists available at ScienceDirect

Computers & Operations Research

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

Hybrid bee colony optimization for examination timetabling problems

M. Alzaqebah [a,*], S. Abdullah [b]

[a] Department of Software Engineering, Faculty of Science and Information Technology, Jadara University, Irbid, Jordan
[b] Data Mining and Optimization Research Group (DMO) Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, Selangor, Bangi 43600, Malaysia

ARTICLE INFO

Available online 28 September 2014

Keywords:
Bee colony optimization algorithm
Examination timetabling problems
Late acceptance hill climbing algorithm
Selection strategy
Self-adaptive mechanism
Simulated annealing

ABSTRACT

Swarm intelligence is a branch of artificial intelligence that focuses on the actions of agents in self-organized systems. Researchers have proposed a bee colony optimization (BCO) algorithm as part of swarm intelligence. BCO is a meta-heuristic algorithm based on the foraging behavior of bees. This study presents a hybrid BCO algorithm for examination timetabling problems. Bees in the BCO algorithm perform two main actions: *forward pass* and *backward pass*. Each bee explores the search space in *forward pass* and then shares information with other bees in the hive in *backward pass*. This study found that a bee decides to be either a recruiter that searches for a food source or a follower that selects a recruiter bee to follow on the basis of roulette wheel selection. In *forward pass*, BCO is supported along with other local searches, including the Late Acceptance Hill Climbing and Simulated Annealing algorithms. We introduce three selection strategies (tournament, rank and disruptive selection strategies) for the follower bees to select a recruiter to maintain population diversity in *backward pass*. The disruptive selection strategy outperforms tournament and rank selections. We also introduce a self-adaptive mechanism to select a neighborhood structure to enhance the neighborhood search. The proposed algorithm is evaluated against the latest methodologies in the literature with respect to two standard examination timetabling problems, namely, uncapacitated and competition datasets. We demonstrate that the proposed algorithm produces one new best result on uncapacitated datasets and comparable results on competition datasets.

1. Introduction

Timetabling problems are NP-hard problems [1] that represent an important subdivision of optimization problems in Operations Research. Timetabling problems exist in many forms. This study solely focuses on Examination Timetabling Problems (ETTPs). ETTPs can be considered as the process to allocate a set of examinations into a limited number of timeslots while satisfying a predetermined set of constraints, both hard and soft. Hard constraints should not be violated in any case to achieve a feasible timetable. Soft constraints represent the quality of the timetable, such that their violations must be minimized [2]. Carter et al. [3] introduced a set of 13 real-world ETTPs from three Canadian high schools in 1996 [4]. Burke et al. [2] described the problem constraints. Furthermore, the second International Timetabling Competition (competition datasets) introduced three tracks of timetabling problems (one for examination and two for course timetabling) that included more real world constraints, as explained in Section 2.

Researchers have proposed and applied many meta-heuristic approaches for solving ETTPs [2–4]. Some approaches are population-based, in which an algorithm works on a number of solutions and tries to improve them. Population-based approaches can be categorized as either *evolutionary algorithms* (EA) or *swarm intelligence* (SI)-based algorithms [6,7]. SI relies on the cooperative behavior of self-organized systems to develop meta-heuristics that mimic the system's problem solving [8,9,25]. Local communication between individuals and their environment contributes to the collective intelligence of social colonies [10]. SI characteristics motivated a number of researchers to employ such behavior for optimization problems [11]. SIs, including an ant algorithm [12], fish swarm optimization algorithm [13], and artificial bee colony algorithm [15], have been widely used to solve ETTPs in the literature.

Bee colony optimization (BCO) is a population-based, cooperative search metaphor inspired by the foraging behavior of bees. We propose a hybrid BCO algorithm for ETTPs.

This work has three main contributions to the BCO algorithm. First, a population of individuals that evolve according to the nature of the disruptive selection strategy is maintained. Second, the algorithm is prevented from becoming *stuck* (no available improvement of neighbors) at local optima through a self-adaptive mechanism that monitors the neighborhood search. Third, a local search algorithm

* Corresponding author.
E-mail addresses: malek.zaqeba@gmail.com (M. Alzaqebah), salwani@ukm.edu.my (S. Abdullah).

http://dx.doi.org/10.1016/j.cor.2014.09.005
0305-0548/

[late acceptance hill climbing algorithm (LAHC)] is internally modified by applying a self-adaptive mechanism to monitor the search.

Experimental results show that algorithm behavior improves when we employ the abovementioned modifications. All modifications were compared and analyzed on the basis of their effects. Overall comparison indicates that the proposed algorithm obtains one better result compared with current state-of-the-art approaches.

The remainder of this paper is structured as follows: Section 2 presents a brief description of ETTP benchmarks. Section 3 discusses the details of the original BCO algorithm. Section 4 explains the proposed modifications to the basic BCO algorithm. Section 5 presents the proposed BCO algorithm. Section 6 presents our experimental results and comparisons. Section 7 concludes the research work carried out in this paper.

2. Examination timetabling problems

2.1. Uncapacitated datasets

Carter et al. [3] introduced the uncapacitated dataset ETTP in 1996.The problem does not consider a room capacity requirement when constructing a timetable. This problem has one hard constraint (*clash-free*) in that a student cannot sit two examinations during the same period when producing a feasible timetable. The soft constraint imposed is to spread examinations as evenly as possible throughout the examination period. As in Burke and Newall [8], ETTPs contain inputs as listed below as follows:

- N is the number of examinations.
- E_i is an examination, $i \in \{1\ldots N\}$.
- T is the given number of available timeslots.
- M is the number of students.
- $C=(c_{ij})_{NxN}$ is the conflict matrix with each element denoted by c_{ij}, $i,j \in \{1,\ldots,N\}$, the number of students taking examinations i and j.
- t_k $(1 \le t_k \le T)$ specifies the assigned timeslot for examination k $(k \in \{1,\ldots,N\})$.

The objective function to minimize the sum of proximity costs is formulated below as follows [33]:

$$\frac{\sum_{i=1}^{N-1} F_1(i)}{M} \tag{1}$$

where

$$F_1(i) = \sum_{j=i+1}^{N} c_{ij} \text{ proximity } (t_i, t_j) \tag{2}$$

Table 1
Characteristics of the uncapacitated datasets.

Datasets	Number of timeslots	Number of examinations	Number of Students	Conflict density
car92	32	543	18,419	0.14
car91	35	682	16,925	0.13
ear83 I	24	190	1125	0.27
hec92 I	18	81	2823	0.42
kfu93	20	461	5349	0.06
lse91	18	381	2726	0.06
pur93 I	42	2419	30,032	0.03
rye92	23	486	11,483	0.07
sta83 I	13	139	611	0.14
tre92	23	261	4360	0.18
uta92 I	35	622	21,267	0.13
ute92	10	184	2750	0.08
yor83 I	21	181	941	0.29

and

$$\text{proximity } (t_i, t_j) = \begin{cases} 2^5/2^{|t_i - t_j|} & \text{if} \quad 1 \le |t_i - t_j| \le 5 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

subject to

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c_{ij} \lambda(t_i, t_j) = 0$$

where

$$\lambda(t_i, t_j) = \begin{cases} 1 & \text{if} \quad t_i = t_j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Eq. (2) represents the examination cost, which is the proximity value multiplied by the number of conflicting students. Eq. (3) presents a proximity value between two examinations [3]. Eq. (4) represents a *clash-free* requirement so no student is asked to sit two examinations at the same time. Table 1 shows the characteristics of the uncapacitated dataset [4].

2.2. Competition datasets

The formulation represents an examination timetabling model that incorporates a significant number of real-world constraints. This formulation was introduced as part of Track 3 of the second International Timetabling Competition (competition datasets). Competition datasets contain real-world constraints and are considered to be complex and more practical datasets than uncapacitated datasets. A complete description of the dataset and the objective function are available in [5].

The benchmark instances for this problem are taken from (http://www.cs.qub.ac.uk/itc2007/index.htm). Table 2 shows the characteristics of these datasets.

The feasibility of the timetable in competition datasets relates to the assignment of all examinations to a period and room without violating the hard constraints listed below as follows [5]:

- No students sit for more than one examination at the same time.
- The total number of students assigned to each room cannot exceed the room capacity.
- The length of examinations assigned to each timeslot should not violate the timeslot length.
- The examination sequences must be respected; for example, *Exam_A* must be scheduled after *Exam_B*.
- Room-related hard constraints must be satisfied; for example, *Exam_A* must be scheduled in *Room 2*.

Table 2
Characteristics of the competition datasets.

Datasets	*D1*	*D2*	*D3*	*D4*	*D5*	*D6*	*D7*	**CD**
Exam_1	7891	32	543	54	7	12	0	5.05
Exam_2	12,743	35	682	40	49	12	2	1.17
Exam_3	16,439	24	190	36	48	170	15	2.62
Exam_4	5045	18	81	21	1	40	0	15.0
Exam_5	9253	20	461	42	3	27	0	0.87
Exam_6	7909	18	381	16	8	23	0	6.16
Exam_7	14,676	42	2419	80	15	28	0	1.93
Exam_8	7718	23	486	80	8	20	1	4.55

$D1$=Number of students. $D2$=Number of actual students in the datasets. $D3$=Number of examinations. $D4$=Number of timeslots $D5$=Number of rooms. $D6$=Period hard constraints. $D7$=Room hard constraints. CD=Conflict density.