Computers & Industrial Engineering 86 (2015) 43–59

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Review

A survey of approaches for university course timetabling problem

Hamed Babaei [a,*], Jaber Karimpour [b], Amin Hadidi [c]

[a] Department of Computer Engineering, College of Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran
[b] Department of Computer Sciences, University of Tabriz, Tabriz, Iran
[c] Department of Mechanical Engineering, College of Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran

ARTICLE INFO

Article history:
Available online 21 November 2014

Keywords:
University course timetabling problem (UCTTP)
Multi-agent based approach (Cooperative Search)
Operational research methods
Metaheuristic methods

ABSTRACT

Scheduling is one of the problems which so many researches have been conducted on it over the years. The university course timetabling problem which is an NP-hard problem is a type of scheduling problem. Timetabling process must be done for each semester frequently, which is an exhausting and time consuming task. The allocation of whole of events in timeslots and rooms performs by the university course timetabling process considering the list of hard and soft constraints presented in one semester, so that no conflict is created in such allocations. In the university course timetabling problem (UCTTP), the hard constraints should not be violated under any conditions; soft constraints also should not be violated as much as possible. The aim of the present paper is to analyze available approaches in the study of university course timetabling problems, including operational researches, metaheuristic methods and intelligent novel methods; also the distributed multi agent systems based approach (Cooperative Search method) is investigated due to its scalability which enables the timetabling of common events between departments. In addition, in this work a complete introduction of reliable datasets has been given to test and evaluation of the structure of considered algorithms.

1. Introduction

The goal of the university course timetabling problem (UCTTP) is to find a method to allocate whole events to fix predefined timeslots and rooms, where all constraints within the problem must be satisfied. Events include students, teachers and courses where resources encompass the facilities and equipment's of classrooms such as theoretical and practical rooms. Also timeslots include two main components, namely daily and weekly timeslots which it varies from one institution to another. However, each classroom also has its own components including audio-visual equipment's (video projector), number of chairs necessary for courses allocated to those classrooms (the capacity of theory and practical rooms), number of blackboards and whiteboards related to each theory and practice classroom and etc.

1.1. Scope and purpose

Object and method of this research in review of University Course Timetabling Problem is presented in Fig. 1.

1.2. Description of the problem

UCTTP is a hybrid optimization problem in the class of NP-hard problems occur at the beginning of each semester of universities and includes the allocation of events (courses, teachers and students) to a number of fixed timeslots and rooms. This problem must satisfy both hard and soft constraints during allocation of events to resources, so that the possible timetables are obtained after full satisfaction of whole hard constraints and also soft constraints to increase and promote the quality of possible generated timetables as necessary (Asmuni, 2008; Obit, 2010; Redl, 2004).

There are some problems and complexities in UCTTP process; firstly, the scheduling process is an NP-complete problem, then it could not be solved in the polynomial time classes because of the exponential growth of this problem and the existence of some variations in the fast growth of students' numbers in this problem, so we must seek heuristic approaches. Secondly, the number of constraints (hard and soft) in this problem differs from one institution to another. Therefore, the main aim of all of the mentioned algorithms is to maximize the number of soft constraints satisfied in the final timetables (Feizi-Derakhshi, Babaei, & Heidarzadeh, 2012; Obit, 2010).

* Corresponding author. Tel.: +98 937 056 4179.
E-mail addresses: hamedbabaei63@gmail.com, h-babaei@iau-ahar.ac.ir (H. Babaei), karimpour@tabrizu.ac.ir (J. Karimpour), amin.hadidi@yahoo.com, a-hadidi@iau-ahar.ac.ir (A. Hadidi).

http://dx.doi.org/10.1016/j.cie.2014.11.010
0360-8352/

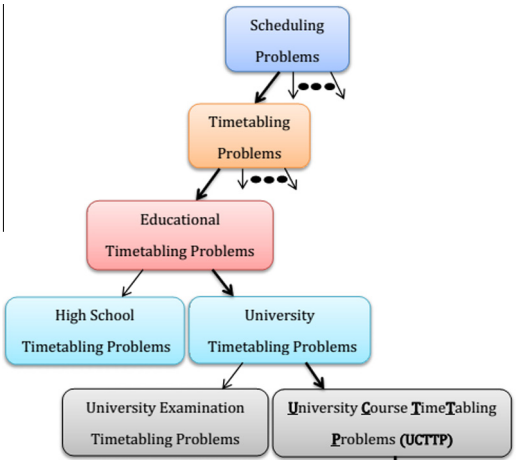

Fig. 1. Diagram of university course timetabling problem.

1.3. The basic definitions of the problem

- *Event:* a scheduled activity, like: teacher, course, and student.
- *Timeslot:* a time interval in which each event is scheduled, like: weekly timeslot such as Tuesday and daily timeslot such as 8–9 a.m. and etc.
- *Resource:* resources are used by events, like: equipment's, rooms, timeslots and etc.
- *Constraint:* a constraint is a restriction in scheduling of events, categorized into two types of hard and soft constraints, like the capacity of classrooms, given timeslot and etc.
- *People:* people include lecturers, students and are a part of events.
- *Conflict:* the confliction of two events with each other, like: scheduling of more than one teacher for one classroom at the same time.

1.4. Different types of constraints in the problem

Constraints in UCTTP problem are classified into two classes of hard and soft constraints. Hard constraints must be satisfied in the problem completely so that the generated solution would be possible and without conflict; no violation is allowed in these constraints. Soft constraints are related to objective function; objective function is to maximize the number of satisfied soft constraints. Unlike hard constraints, soft constraints are not necessarily required to satisfy; but as the number of these satisfied constraints increases, the quality of solutions of objective function increases. In the following, a list of hard and soft constraints presented which are taken from literature (Asmuni, 2008; Feizi-Derakhshi et al., 2012; Gotlib, 1963; Lewis, 2006; Obit, 2010; Redl, 2004).

1.4.1. Hard constraints

- A teacher could not attend two classes at the same time.
- A course could not be taught in two different classes at the same time.
- A teacher teaches only one course in one room at each timeslot.
- At each daily timeslot in one room only one group of students and one teacher could attend.
- A teacher teaches for only one group of students at each daily timeslot.
- There are some predefined courses which are scheduled in a given timeslots.
- The capacity of the classrooms should be proportional to the number of students of the given course.

1.4.2. Soft constraints

- The teacher can have the choice to suggest priority certain timeslots for her/his courses either public or private times.
- A teacher may request a special classroom for a given course.
- The courses should be scheduled in a way that the empty timeslots of both teacher and student to be minimized.
- Timetabling of the courses should be conducted in a way that the courses not scheduled at evening timeslots, as it is possible; unless an evening timeslot has been requested by a particular teacher.
- The lunch break is either 12–13 p.m. or 13–14 p.m., usually.
- The start time of classes may be 8 a.m. and the ending time may be 20:30 p.m. (evening), usually.
- The maximum teaching hours for teachers in a classroom are 4 h.
- The maximum learning hours for students is 4 h.
- Scheduling should be conducted in a way that one or a group of students not attend university for one timeslot in a day.

1.5. Mathematical formulation of the problem

Formal definition of UCTTP problem includes n: the number of events $E=\{e_1, e_2, \ldots, e_n\}$, k: the number of timeslots $T=\{t_1, t_2, \ldots, t_k\}$, m: the number of rooms $R=\{r_1, r_2, \ldots, r_m\}$, L: the number of rooms' features $F=\{f_1, f_2, \ldots, f_l\}$ and s: the set of students $S=\{s_1, s_2, \ldots, s_s\}$. For example, if the number of daily timeslots is 9 and the number of weekly timeslots is 5, then the total timeslots will be $T = 9 \times 5 = 45$ (Asmuni, 2008; Obit, 2010; Redl, 2004; Wangmaeteekul, 2011).

The input data for each sample problem (data sets) include the size and features of each room, the number of students in an event and information about conflicting events. So, we should know the procedure of measuring violation and non-violation of hard and soft constraints in order to have the ability to replace events within matrixes. At first the penalty function per violation from soft constraint must be calculated for each solution which is corresponding to a timetable, as bellow (Asmuni, 2008; Obit, 2010; Redl, 2004; Wangmaeteekul, 2011):

$$PF(S) = \sum_{j=1}^{SC} W_j \times (-1) \tag{1}$$

In Eq. (1), S is the solution, W_j is the weight of each soft constraint (value 0 means non-violation, value 1 means violation and -1 shows the cost of each violation per soft constraint) and SC is the number of soft constraints. However, PF represents the penalty function. Value of objective function per solution considering hard constraints can be calculated as:

$$\mathrm{OF}(S) = \sum_{i=1}^{HC} W_i \times (-1) + \mathrm{PF}(S) \tag{2}$$

In Eq. (2), W_i is the weight of each hard constraint where value 0 means non-violation, value 1 means violation and -1 shows the cost of each violation per hard constraint. Also HC and OF are the number of hard constraints, and the objective function, respectively. Always the value of first term of right hand side of the Eq. (2) is equal to zero ($\sum_{i=1}^{HC} W_i \times (-1) = 0$), this means that the violation of hard constraints is not feasible. So $OF(S) = 0 + PF(S)$, consequently $OF(S) = PF(S)$.

In order to determine the violation of solutions, from hard and soft constraints, results of sample problems are stored in five matrixes namely STUDENT-EVENT, EVENT-CONFLICT, ROOM-FEATURES, EVENT-FEATURES and EVENT-ROOM which is introduced in the following.