Computers & Operations Research 65 (2016) 83–92

Contents lists available at ScienceDirect

Computers & Operations Research

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

Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem

Ruggero Bellio [a], Sara Ceschia [b], Luca Di Gaspero [b], Andrea Schaerf [b,*], Tommaso Urli [c]

[a] DIES, University of Udine, via Tomadini 30/A, I-33100 Udine, Italy
[b] DIEGM, University of Udine, via delle Scienze 208, I-33100 Udine, Italy
[c] NICTA, Canberra Research Laboratory, Tower A, 7 London Circuit, Canberra ACT 2601, Australia

ARTICLE INFO

Available online 14 July 2015

Keywords:
Course timetabling
Simulated annealing
Metaheuristics
Feature analysis
Parameter tuning

ABSTRACT

We consider the university course timetabling problem, which is one of the most studied problems in educational timetabling. In particular, we focus our attention on the formulation known as the *curriculum-based course timetabling problem* (CB-CTT), which has been tackled by many researchers and for which there are many available benchmarks.

The contribution of this paper is twofold. First, we propose an effective and robust single-stage simulated annealing method for solving the problem. Second, we design and apply an extensive and statistically-principled methodology for the parameter tuning procedure. The outcome of this analysis is a methodology for modeling the relationship between search method parameters and instance features that allows us to set the parameters for unseen instances on the basis of a simple inspection of the instance itself. Using this methodology, our algorithm, despite its apparent simplicity, has been able to achieve high quality results on a set of popular benchmarks.

A final contribution of the paper is a novel set of real-world instances, which could be used as a benchmark for future comparison.

1. Introduction

The issue of designing the timetable for the courses of the incoming term is a typical problem that universities or other academic institutions face at each semester. There are a large number of variants of this problem, depending on the specific regulations at the involved institutions (see, e.g., [23,26,35]). Among the different variants, two in particular are now considered standard, and featured at the international timetabling competitions ITC-2002 and ITC-2007 [32]. These standard formulations are the *Post-Enrollment Course Timetabling* (PE-CTT) [27] and *Curriculum-Based Course Timetabling* (CB-CTT) [15], which have received an appreciable attention in the research community so that many recent articles deal with either one of them.

The distinguishing difference between the two formulations is the origin of the main constraint of the problem, i.e., the conflicts between courses that prevent one from scheduling them simultaneously. Indeed, in PE-CTT the source of conflicts is the actual student enrollment whereas in CB-CTT the courses in conflict are those that belong to the same predefined group of courses, or curricula. However, this is only one of the differences, which actually include many other distinctive features and cost components. For example in PE-CTT each course is a self-standing event, whereas in CB-CTT a course consists of multiple lectures. Consequently the soft constraints are different: in PE-CTT they are all related to events, penalizing late, consecutive, and isolated ones, while in CB-CTT they mainly involve curricula and courses, ensuring compactness in a curriculum, trying to evenly spread the lectures of a course in the weekdays, and possibly preserving the same room for a course.

In this work we focus on CB-CTT, and we build upon our previous work on this problem [4]. The key ingredients of our method are (i) a fast single-stage Simulated Annealing (SA) method, and (ii) a comprehensive statistical analysis methodology featuring a principled parameter tuning phase. The aim of the analysis is to model the relationship between the most relevant parameters of the solver and the features of the instance under consideration. The proposed approach tackles the parameter selection as a classification problem, and builds a rule for choosing the set of parameters most likely to perform well for a given instance, on the basis of specific features. The effectiveness of this methodology is confirmed by the experimental results on two groups of validation instances. We are able to show that our

* Corresponding author.
E-mail addresses: ruggero.bellio@uniud.it (R. Bellio), sara.ceschia@uniud.it (S. Ceschia), l.digaspero@uniud.it (L. Di Gaspero), schaerf@uniud.it (A. Schaerf), tommaso.urli@nicta.com.au (T. Urli).

http://dx.doi.org/10.1016/j.cor.2015.07.002
0305-0548/

method compares favorably with all the state-of-the-art methods on the available instances.

As an additional contribution of this paper, we extend the set of available problem instances by collecting several new real-world instances that can be added to the set of standard benchmarks and included in future comparison.

The source code of the solver developed for this work is publicly available at https://bitbucket.org/satt/public-cb-ctt.

2. Curriculum-based course timetabling

The problem formulation we consider in this paper is essentially the version proposed for the ITC-2007, which is by far the most popular. The detailed formulation is presented in Di Gaspero et al. [15], however, in order to keep the paper self-contained, we briefly report it also in the following. Alternative formulations of the problem, used in the experimental part of the paper (Section 6.7), are described in Bonutti et al. [8]. Essentially, the problem consists of the following entities:

Days, Timeslots, and Periods: We are given a number of *teaching days* in the week. Each day is split into a fixed number of *timeslots*. A *period* is a pair composed of a day and a timeslot.

Courses and teachers: Each course consists of a fixed number of *lectures* to be scheduled in distinct periods, it is attended by a number of *students*, and is taught by a *teacher*. For each course, there is a minimum number of days across which the lectures of the course should be spread. Moreover, there are some periods in which the course cannot be scheduled (e.g., teacher's or students' availabilities).

Rooms: Each *room* has a *capacity*, i.e., a number of available seats.

Curricula: A *curriculum* is a group of courses that share common students. Consequently, courses belonging to the same curriculum are *in conflict* and cannot be scheduled at the same period.

A solution of the problem is an assignment of a period (day and timeslot) and a room to all lectures of each course so as to satisfy a set of *hard* constraints and to minimize the violations of *soft* constraints described in the following.

2.1. Hard constraints

There are three types of hard constraints:

RoomOccupancy: Two lectures cannot take place simultaneously in the same room.

Conflicts: Lectures of courses in the *same curriculum*, or *taught by the same teacher*, must be scheduled in different periods.

Availabilities: A course may not be available for being scheduled in a certain period.

2.2. Soft constraints

For the formulation ITC-2007 there are four types of soft constraints:

RoomCapacity: The capacity of the room assigned to each lecture must be greater than or equal to the number of students attending the corresponding course. The penalty for the violation of this constraint is measured by the number of students in excess.

MinWorkingDays: The lectures of each course cannot be tightly packed, but they must be spread into a given minimum number of days.

IsolatedLectures: Lectures belonging to a curriculum should be adjacent to each other (i.e., be assigned to consecutive periods). We account for a violation of this constraint every time, for a given curriculum, there is one lecture not adjacent to any other lecture of the same curriculum within the same day.

RoomStability: All lectures of a course should be given in the same room.

For all the details, including input and output data formats and validation tools, we refer to Di Gaspero et al. [15].

3. Related work

In this section we review the literature on CB-CTT. The presentation is organized as follows: we firstly describe the solution approaches based on metaheuristic techniques; secondly, we report the contributions on exact approaches and on methods for obtaining lower bounds; finally, we discuss papers that investigate additional aspects related to the problem, such as instance generation and multi-objective formulations. A recent survey covering all these topics is provided by Bettinelli et al. [6].

3.1. Metaheuristic approaches

Müller [34] solves the problem by applying a constraint-based solver that incorporates several local search algorithms operating in three stages: (i) a construction phase that uses an *Iterative Forward Search* algorithm to find a feasible solution, (ii) a first search phase delegated to a *Hill Climbing* algorithm, followed by (iii) a *Great Deluge* or *Simulated Annealing* strategy to escape from local minima. The algorithm was not specifically designed for CB-CTT but it was intended to be employed on all three tracks of ITC-2007 (including, besides CB-CTT and PE-CTT, also Examination Timetabling). The solver was the winner of two out of three competition tracks, and it was among the finalists in the third one.

The *Adaptive Tabu Search* proposed by Lü and Hao [30] follows a three stage scheme: in the initialization phase a feasible timetable is built using a fast heuristic; then the intensification and diversification phases are alternated through an adaptive tabu search in order to reduce the violations of soft constraints.

A novel hybrid metaheuristic technique, obtained by combining *Electro-magnetic-like Mechanisms* and the *Great Deluge* algorithm, was employed by Abdullah et al. [1] who obtained high-quality results on both CB-CTT and PE-CTT testbeds.

Finally, Lü et al. [31] investigated the search performance of different neighborhood relations typically used by local search algorithms to solve this problem. The neighborhoods are compared using different evaluation criteria, and new combinations of neighborhoods are explored and analyzed.

3.2. Exact approaches and methods for computing lower bounds

Several authors tackled the problem by means of exact approaches with both the goal of finding solutions or computing lower bounds.

Among these authors, Burke et al. [12] engineered a hybrid method based on the decomposition of the whole problem into different sub-problems, each of them solved using a mix of different IP formulations. Subsequently, the authors propose a