European Journal of Operational Research 240 (2015) 637–648

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Analysing the effects of solution space connectivity with an effective metaheuristic for the course timetabling problem

R. Lewis *, J. Thompson

School of Mathematics, Cardiff University, Cardiff CF24 4AG, Wales, United Kingdom

ARTICLE INFO

Article history:
Received 21 November 2013
Accepted 25 July 2014
Available online 4 August 2014

Keywords:
University timetabling
Metaheuristics
Solution space connectivity

ABSTRACT

This paper provides a mathematical treatment of the NP-hard post enrolment-based course timetabling problem and presents a powerful two-stage metaheuristic-based algorithm to approximately solve it. We focus particularly on the issue of solution space connectivity and demonstrate that when this is increased via specialised neighbourhood operators, the quality of solutions achieved is generally enhanced. Across a well-known suite of benchmark problem instances, our proposed algorithm is shown to produce results that are superior to all other methods appearing in the literature; however, we also make note of those instances where our algorithm struggles in comparison to others and offer evidence as to why.

1. Introduction

Timetables are organisational structures that can be found in many areas of human activity including sports (Kendall, Knust, Ribeiro, & Urrutia, 2010), entertainment (Eliashberg et al., 2009), transport (Caprara, Fischetti, & Toth, 2002), industry (Blöchliger, 2004), and education (Lewis, 2008). In the context of higher education institutions, a timetable can be thought of as an assignment of events (such as lectures, tutorials, or exams) to a finite number of rooms and timeslots in accordance with a set of constraints, some of which will be mandatory, and others that may be optional (Corne, Ross, & Fang, 1995). According to McCollum et al. (2010), the problem of constructing such timetables can be divided into two categories: exam timetabling problems and course timetabling problems. It is also suggested that course timetabling problems can be further divided into two sub-categories: "post enrolment-based course timetabling", where the constraints of the problem are specified by student enrolment data, and "curriculum-based course timetabling", where constraints are based on curricula specified by the university. Müller and Rudova (2012) have also shown that these sub-categories are closely related, demonstrating how instances of the latter can be transformed into those of the former in many cases.

The field of university timetabling has seen many solution approaches proposed over the past few decades, including methods based on constructive heuristics, mathematical programming, branch and bound, and metaheuristics (see Carter and Laporte, 1996; Carter and Laporte, 1998; Lewis, 2008; Schaerf, 1999 for surveys of these). A near universal constraint in problems considered in the literature is the "event-conflict" constraint, which specifies that certain pairs of events cannot be assigned to the same timeslot (e.g. there may be some students who need to attend both events). The presence of this constraint allows parallels to be drawn between timetabling and the well-known NP-hard problem of graph colouring, and it is certainly the case that heuristics derived from graph colouring are often used in timetabling algorithms (Burke & Newall, 1999; Carter, Laporte, & Lee, 1996; Thompson & Dowsland, 1998).[1] Beyond this constraint however, timetabling problem formulations have also tended to vary quite widely in the literature because each institution usually has their own specific needs and protocols (hence constraints) that need adhering to. While making the problem area very rich, one noted drawback has been the lack of opportunity for accurate comparison of algorithms over the years (Lewis, 2008).

In the past decade, this situation has been mitigated to a certain extent due to the organisation of a series of timetabling competitions and the release of publicly available problem instances (http://www.idsia.ch/Files/ttcomp2002; http://www.cs.qub.ac.uk/itc2007/; http://www.utwente.nl/ctit/hstt/itc2011/welcome/). In 2007, for example, the Second International Timetabling Competition (ITC2007) was organised by a group of timetabling researchers from different European Universities, which considered the three

* Corresponding author.
E-mail addresses: lewisR9@cf.ac.uk (R. Lewis), thompsonJM1@cf.ac.uk (J. Thompson).

[1] Timetabling problems can be transformed into graph colouring problems by considering the events as the vertices, the timeslots as the colours, with edges being added between any pair of events deemed to be conflicting.

http://dx.doi.org/10.1016/j.ejor.2014.07.041
0377-2217/© 2014 Elsevier B.V. All rights reserved.

types of timetabling problems mentioned above: exam timetabling, post enrolment-based course timetabling, and curriculum-based timetabling. The competition operated by releasing problem instances into the public domain, with entrants then designing algorithms to try and solve these. Entrants' algorithms were then compared under strict time limits according to specific evaluation criteria. Further details can be found in McCollum et al. (2010) and on the competition website (http://www.cs.qub.ac.uk/itc2007/).

In this paper we give a mathematical description of the post enrolment-based course timetabling problem used for ITC2007. This particular formulation models the real-world situation where students are given a choice of lectures that they wish to attend, with the timetable then being constructed according to these choices. We present a two-stage algorithm for this problem, focussing particularly on issues surrounding solution space connectivity. We show that when this is increased via specialised neighbourhood operators combined with sensible design decisions, state of the art performance across a range of problem instances can be achieved. In Section 3, we present a review of the most noteworthy and/or recent high-performance algorithms for this problem, before going on to describe our method and its operators in Sections 4 and 5. The final results of our algorithm are given in Section 6, with a discussion and conclusions then being presented in Section 7.

2. Problem description and preprocessing

As mentioned, the post enrolment-based course timetabling problem was introduced for use in the Second International Timetabling Competition, run in 2007 (http://www.cs.qub.ac.uk/itc2007/; Lewis, Paechter, & McCollum, 2007). The problem involves seven "hard" constraints (described below) whose satisfaction is mandatory, and three "soft" constraints, whose satisfaction is desirable, but not essential. The problem involves assigning a set of events to 45 timeslots (5 days, with 9 timeslots per day) according to these constraints.

The hard constraints for the problem are as follows. First, for each event there is a set of students who are enrolled to attend; thus events should be assigned to timeslots such that no student is required to attend more than one event in any one timeslot. Next, each event also requires a set of room features (e.g. a certain number of seats, specialist teaching equipment, etc.), which will only be provided by certain rooms; thus each event needs to be assigned to a suitable room that exhibits the room features that it requires. The double booking of rooms is also disallowed. Hard constraints are also imposed stating that some events cannot be taught in certain timeslots. Finally, precedence constraints – stating that some events need to be scheduled before or after others – are also stipulated.

More formally, a problem instance comprises a set of events $e = \{e_1, \ldots, e_n\}$, a set of timeslots $t = \{t_1, \ldots, t_{|t|}\}$ (where $|t| = 45$), a set of students $s = \{s_1, \ldots, s_m\}$, a set of rooms $r = \{r_1, \ldots, r_{|r|}\}$, and a set of room features $f = \{f_1, \ldots, f_{|f|}\}$. Each room $r_i \in r$ is also allocated a capacity $c(r_i)$ reflecting the number of seats it contains. The relationships between these sets are defined by five matrices:

- An *attends* matrix $\mathbf{P}^{(1)}_{m\times n}$, where $P^{(1)}_{i,j} = 1$ if student s_i is due to attend event e_j; 0 otherwise.
- A *room features* matrix $\mathbf{P}^{(2)}_{|r|\times|f|}$, where $P^{(2)}_{i,j} = 1$ if room r_i has feature f_j; 0 otherwise.
- An *event features* matrix $\mathbf{P}^{(3)}_{n\times|f|}$, where $P^{(3)}_{i,j} = 1$ if event e_i requires feature f_j; 0 otherwise.
- An *event availability* matrix $\mathbf{P}^{(4)}_{n\times|t|}$, where $P^{(4)}_{i,j} = 1$ if event e_i can be assigned to timeslot t_j; 0 otherwise.
- A *precedence* matrix $\mathbf{P}^{(5)}_{n\times n}$, where $P^{(5)}_{i,j} = 1$ if event e_i should be scheduled to an earlier timeslot than event e_j; $P^{(5)}_{i,j} = -1$ if event e_i should be scheduled to a later timeslot than event e_j; and 0 otherwise.

For the precedence matrix above, two conditions are necessary for the relationships to be consistent: (a) $P^{(5)}_{i,j} = 1 \iff P^{(5)}_{j,i} = -1$, and (b) $P^{(5)}_{i,j} = 0 \iff P^{(5)}_{j,i} = 0$. We can also observe the transitivity of this relationship:

$$\left(\exists e_i, e_j, e_k \in e \middle| \left(P^{(5)}_{i,j} = 1 \wedge P^{(5)}_{j,k} = 1\right)\right) \Rightarrow P^{(5)}_{i,k} = 1 \qquad (1)$$

As noted by Lewis (2012), in some of the competition problem instances this transitivity is not fully expressed; however, observing it enables further 1's and −1's to be added to $\mathbf{P}^{(5)}$ during preprocessing, allowing the relationships to be more explicitly stated.

Given the above five matrices, in our approach two further matrices are calculated, which allow fast detection of hard constraint violations during execution of the algorithm (Kostuch, 2005; Cambazard, Hebrard, O'Sullivan, & Papadopoulos, 2012).

- A *room suitability* matrix $\mathbf{R}_{n\times|r|}$, where:

$$R_{i,j} = \begin{cases} 1 & \text{if } \left(\sum_{k=1}^{m} P^{(1)}_{k,i} \leqslant c(r_j)\right) \wedge \left(\nexists\, f_k \in f \middle| \left(P^{(3)}_{i,k} = 1 \wedge P^{(2)}_{j,k} = 0\right)\right) \\ 0 & \text{otherwise.} \end{cases} \qquad (2)$$

- A *conflicts* matrix $\mathbf{C}_{n\times n}$, where:

$$C_{i,j} = \begin{cases} 1 & \text{if}\left(\exists s_k \in s \middle| \left(P^{(1)}_{k,i} = 1 \wedge P^{(1)}_{k,j} = 1\right)\right) \\ & \vee \left((\exists r_k \in r | (R_{i,k} = 1 \wedge R_{j,k} = 1)) \wedge \left(\sum_{k=1}^{|r|} R_{i,k} = 1\right) \wedge \left(\sum_{k=1}^{|r|} R_{j,k} = 1\right)\right) \\ & \vee \left(P^{(5)}_{i,j} \neq 0\right) \\ & \vee \left(\nexists t_k \in t \middle| \left(P^{(4)}_{i,k} = 1 \wedge P^{(4)}_{j,k} = 1\right)\right) \\ 0 & \text{otherwise.} \end{cases} \qquad (3)$$

The matrix $\mathbf{R}$ specifies the rooms that are suitable for each event (i.e. rooms that are large enough for all attending students and that have all the required features). $\mathbf{C}$, meanwhile, is a symmetrical matrix ($C_{i,j} = C_{j,i}$) that specifies pairs of events that should not be assigned to the same timeslot (i.e. those that conflict). According to (3) this will be the case if two events e_i and e_j share a common student, require the same individual room, are subject to a precedence relation, or have mutually exclusive subsets of timeslots for which they are available.

Having defined the input to this problem, a solution is represented by an ordered set of sets $\mathcal{S} = (S_1, \ldots, S_{|t|})$ subject to the satisfaction of the following hard constraints.

$$\bigcup_{i=1}^{|t|} S_i \subseteq e \qquad (4)$$

$$S_i \cap S_j = \emptyset \quad (1 \leqslant i \neq j \leqslant |t|) \qquad (5)$$

$$\forall e_j, e_k \in S_i,\, C_{j,k} = 0 \quad (1 \leqslant i \leqslant |t|) \qquad (6)$$

$$\forall e_j \in S_i,\, P^{(4)}_{j,i} = 1 \quad (1 \leqslant i \leqslant |t|) \qquad (7)$$

$$\forall e_j \in S_i,\, e_k \in S_{l<i},\, P^{(5)}_{j,k} \neq 1 \quad (1 \leqslant i \leqslant |t|) \qquad (8)$$

$$\forall e_j \in S_i,\, e_k \in S_{l>i},\, P^{(5)}_{j,k} \neq -1 \ (1 \leqslant i \leqslant |t|) \qquad (9)$$

$$S_i \in \mathcal{M} \ (1 \leqslant i \leqslant |t|) \qquad (10)$$

Constraints (4) and (5) state that $\mathcal{S}$ should partition the event set e (or a subset of e) into an ordered set of sets, labeled $S_1, \ldots, S_{|t|}$. Each set $S_i \in \mathcal{S}$ contains the events that are assigned to timeslot t_i in the timetable. Constraint (6) stipulates that no pair