Automated Course Timetabling at Escuela Politécnica Nacional

Luis M. Torres · Ramiro Torres

Abstract The course timetabling problem consists in the allocation, subject to side constraints, of a number of class sessions to a limited number of time slots and classrooms, so that each session is taught by a lecturer in a suitable classroom and at a given time period within a planning horizon, which is usually a week. The timetabling problem has been extensively studied in the fields of combinatorial optimization and operations research for about 50 years due to its wide applicability (see, for instance, [1,3,4] for detailed surveys). An established way to model this problem consists in searching for an assignment of class sessions to time slots and classrooms, subject to hard and soft constraints. Hard constraints are those that must be fulfilled by any feasible solution and soft constraints are those that are desirable, but not essential.

The timetabling problem considered in this talk consists in scheduling the teaching activities at the undergraduate level at the Escuela Politécnica Nacional (EPN) in Quito, Ecuador. Teaching in this Higher Education institution is organized on a semester basis, and the following activities are involved in the planning for each term: the list of courses to be offered is determined, lecturers are assigned to these courses, the expected number of enrolled students is estimated for each course, and a timetable is computed from this information. Students enroll in the courses only after the timetable has been made available to the public. Moreover, students are not allowed to register in courses that have conflicting timetables.

Students pursuing some specific degree program must approve a set of *subjects* organized in a plan that we will denote in the following as the *curriculum* of the program. Among other things, this curriculum specifies for each subject

L.M. Torres
Centro de Modelización Matemática, Escuela Politécnica Nacional, Quito, Ecuador
Tel.: +593-2-2976300-1535
E-mail: luis.torres@epn.edu.ec

R. Torres
Departamento de Matemática, Escuela Politécnica Nacional, Quito, Ecuador

Proceedings of the 12th International Conference on the Practice and Theory of Automated Timetabling (PATAT-2018), Vienna, Austria, August 28–31, 2018

a referential *level* , i.e., the term during his degree course in which a student should enroll in the subject (e.g. in all engineering programs, Calculus belongs to level I, Multivariate Calculus belongs to level II).

Depending on the number of expected enrolled students, one or more *student groups* are defined for each level of each degree program. Teaching of subjects is organized in *courses*, each one assigned to a lecturer and to one or more student groups. A given course may be attended by student groups from different degree programs. Two courses are *conflicting* if they are either assigned to the same lecturer, or have at least one student group in common.

The teaching activities for each course are carried out in a number of weekly *sessions* whose duration range from one to three periods of 60 minutes each. For each course session a required *classroom type* is specified (e.g., small room, large room, computer lab, etc.)

A course timetable is an assignment that specifies for each session of each course (i) a valid starting time (usually, sessions may start only at whole hours between 7 a.m. and 6 p.m.) at a certain day in the week, and (ii) a classroom of a valid type and capacity large enough to host the expected number of enrolled students. Two courses are said to be *clashing* if they are conflicting and some of their sessions overlap in time. Moreover, two courses are *room-overlapping* if some of their sessions overlap in time and are assigned to the same room.

The task is to find a timetable with no clashing or room-overlapping courses which additionally satisfies the following two hard constraints:

– the availability requirements of lecturers is taken into account, and,
– for each course, no more than one session is scheduled per day.

Moreover, four weak constraints should be taken into account when computing a timetable: (i) *schedule preferences* specified by the lecturers; (ii) upper bounds on the *length* of the daily timetables for lecturers; (iii) *compactness* of the daily timetables for lecturers; and (iv) minimization of *weak clashes*. A weak clash occurs when two courses involving student groups corresponding to neighboring levels of the same degree program are assigned schedules that overlap in time.

The timetabling problem at EPN fits very well into the generic description of the University Course Timetabling Problem [3]. At the same time, the problem reveals specific features that are not considered commonly, for instance the upper limit of one daily session for each course and the minimization of weak clashes.

We have formulated a linear integer programming model for the above problem, which involves binary variables for assigning sessions to periods and rooms, binary variables for signaling weak clashes between pairs of courses, and integer variables for determining the start and end of daily schedules of lecturers. Hard constraints are implemented as linear restrictions while soft constraints are expressed through corresponding terms in the objective function.

Preliminary computational results have been obtained by solving a simplified version of the model (obtained from dropping the second and third

Table 1 Preliminary results on two instances from the current planning at EPN.

Faculty	Sciences	Chem. Eng.
Programs	4	2
Courses	115	162
Sessions	272	252
Rooms	22	42
Lecturers	68	54
Time (s)	10,041	5,508
Obj.	852	835
LB	805	835
Gap	–	5.5%

weak constraints) on an Intel Core i7 3.60 GHZ computer with 8 GB RAM, running the commercial IP solver Gurobi 7.01 [2] under Ubuntu 14.01. The test instances stem from two faculties of EPN: the Faculty of Sciences, which offers four undergraduate programs, and the Faculty of Chemical Engineering, which offers two undergraduate programs. Some figures concerning instance data and results are provided in Table 1.

In both cases, the models turn out to be too large to be tackled directly by the solver. Hence, restrictions forbidding clashes of courses are separated dynamically and implemented as lazy constraints.

The preliminary results suggest that the exact integer programming approach seems to be well suited for this application: in less than three hours computing time one instance is solved to optimality and the other one is solved within an optimality gap of less than 6%.

Keywords university course timetabling · integer programming models · integer optimization

References

1. Burke, E. and Petrovic, S., Recent Search Directions in Automated Timetabling, European Journal of Operational Research, vol. 140, no. 2, pp. 266-280, (2002).
2. Gurobi Optimization, Inc., Gurobi Optimizer Reference Manual, (2017). http://www.gurobi.com.
3. Mühlenthaler, M., Fairness in Academic Course Timetabling. Springer International Publishing, Switzerland (2015).
4. Schaerf, A., A Survey of Automated Timetabling, Artificial Intelligence Review, vol. 13, pp. 87-127, (1999).