Journal name manuscript No.
(will be inserted by the editor)

International Timetabling Competition 2019: A Mixed Integer Programming Approach for Solving University Timetabling Problems

Efstratios Rappos · Eric Thiémard · Stephan Robert · Jean-François Hêche

Received: date / Accepted: date

Abstract This summary article presents the mathematical programming approach used to solve and optimize the problem instances of the International Timetabling Competition 2019. The optimization problem was modeled as a mixed integer program which was solved using traditional branch-and-cut methods. Several innovative elements enabled us to achieve very good performance, such as the precalculation of several characteristics of the instances, the aggregation of constraints and the efficient use of auxiliary variables in the formulation. The computational implementation consisted of a first stage algorithm to obtain a feasible solution and an iterative local search algorithm which improves the objective function and the quality of the resulting timetable.

Keywords ITC 2019 · timetabling problems · integer programming · combinatorial optimization

1 Introduction

This summary paper describes the method used to solve the timetabling problems of the 2019 International Timetabling Competition [1]. Overall, we were able to solve 29 out of the 30 problem instances; one instance was not solved in the required timescales due to resource limitations.

2 Model formulation

The mathematical modeling approach consists of a linear mixed integer program. The formulation uses four sets of binary 0-1 variables, x, y, z and Z, representing the class times, class rooms, student-class allocation and student-course configuration allocation respectively. The indices of these variables are described in Table 1 and their values uniquely represent a solution to a timetabling problem.

Efstratios Rappos · Eric Thiémard · Stephan Robert · Jean-François Hêche
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud (HEIG-VD)
University of Applied Sciences of Western Switzerland (HES-SO)
Yverdon-les-Bains, Switzerland
E-mail: efstratios.rappos@heig-vd.ch

Variable	*Value*
$x_{c,t}$	1 if the class c takes place at time t, 0 otherwise
$y_{c,r}$	1 if the class c takes place in room r, 0 otherwise
$z_{s,c}$	1 if the student s is assigned to the class c, 0 otherwise
$Z_{s,f}$	1 if the student s follows the course configuration f, 0 otherwise

Table 1 Variables used in the formulation

The binary variables are linked between them with a set of linear constraints to represent the characteristics of the timetabling problems of the 2019 International Timetabling Competition. These constraints must be satisfied by any feasible solution and are summarized in Table 2.

Constraint	*Meaning*
C-1	Every class must be assigned a time
C-2	Every class must be assigned a room, where applicable
C-3	Every student must attend exactly one class from each subpart of the selected course configuration for each course that he must attend
C-4	For two classes with a parent-child relationship, if a class is assigned to a student then the parent class must also be assigned
C-5	Every student must be assigned a course configuration for every course that he follows
C-6	The capacity of each class in terms of the number of students must be satisfied
C-7	A room cannot be used when it is unavailable
C-8	Two classes cannot take place at the same time in the same room
C-9	Any hard distribution constraints must be satisfied

Table 2 Constraints used in the formulation

The first seven constraints are straightforward to formulate as linear inequalities using the above decision variables. The last two constraints C-8 and C-9 are modeled as inequalities of the form

$$x_{c_1,t_1} + y_{c_1,r_1} + x_{c_2,t_2} + y_{c_2,r_2} \leq 3 \tag{1}$$

for every combination of class times and rooms which leads to a violated constraint. For the hard distribution constraints which are not representable as inequalities of the form "for each pair of classes" of the constraint (namely, the special MaxDays, MaxDayload, MaxBreaks and MaxBlock constraints), an additional step is taken to check that these constraints are satisfied when a potential new solution is found, and to reject the solution if they any special constraints are violated.

The objective function consists of four terms which are described in the problem specification [1], namely the class time, class room, distribution and student clash terms. The first two terms are simply the weighted sum of the x and y variables. For the soft distribution constraints an auxiliary variable is introduced in equation (1) which determines if the constraint is satisfied or not, and the sum of these variables is used as the objective function. For the cases where this is not possible (the special constraints) a list of such violated constraints is maintained throughout the optimization which is read and included in the model at the start of each optimization round, so that they are also partially included in the optimization objective.

A similar method is used for the student clashes. An auxiliary variable is introduced for every student and pair of classes he may follow, specifying whether a clash between these classes exists or not. As the number of potential student clashes is large, we introduce an indicator variable which specifies, for each pair of classes, whether a student follows both classes or not. This enables us to aggregate the penalties associated with all students for a particular pair of classes into one equation, greatly reducing the size of the formulation. Note that we consider both student clashes where two classes overlap and when the travel time between the classes is insufficient.

Several innovative techniques are used to deal with the very large number of constraints of the above formulation, which makes it possible to solve the competition instances in a reasonable amount of time. The efficiency optimizations include:

– Extensive use of precalculated problem characteristics to save time between runs, for example the minimum and maximum gap and travel distances between classes
– Four types of checks to eliminate variables whose value is fixed
– Removal of constraints that are always satisfied
– Reduction of the problem size by constraint aggregation

The computational implementation was done in Java using the commercial software CPLEX and Gurobi as the mixed integer programming solvers. The solution strategy consisted of two stages: the first stage focused at obtaining a feasible solution which satisfies all the constraints, whereas the second stage aimed at minimizing the objective function.

The first stage performs a progressive addition of constraints into the model while fixing (usually at random) a number of variables to their current solution values. Any violated constraints are added to the model using slack variables whose sum we aim to minimize. This yields a sequence of mixed integer programs containing an increasing number of satisfied constraints, converging over time to a feasible solution.

Once a feasible solution is obtained, the second stage begins where the four terms of the objective function are added, while again fixing a number of variables to their current values. Several strategies of fixing the variables (local search) were developed and implemented sequentially, such as fixing only the x or y variables, fixing all variables within a class or fixing all classes within a soft distribution constraint. Depending on the problem, this stage can converge quickly to a very good or optimal solution, or in other problems it keeps producing better and better solutions until the optimization is stopped for practical reasons.

3 Conclusions

This summary article presented a mixed integer programming approach for solving the timetabling problems of the 2019 International Timetabling Competition. Although the problem size for a typical timetable is very large to be solved exactly using traditional mixed integer programming tools, several improvements can significantly reduce the size to manageable levels. Once a feasible solution was obtained, the use of mixed integer programming for the local search optimization stage proved to be very powerful in improving the quality of the solutions very quickly.

References

1. Müller, T., Rudová, H., Müllerová, Z.: University course timetabling and International Timetabling Competition 2019. In: PATAT 2018—Proceedings of the 12th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2018) (2018)