PPC/PL REPORT

Constraint programming for the university timetabling problem

Produced by
Alessandro Pepegna
Meryem Boufalah

Supervised by
Arnaud Malapert

Master Informatique
at Université Côte d'Azur
First semester of the year 2021-2022

Abstract

In this report, we show the implementations of two models to resolve the university timetabling problem. The modeling has been realized thanks to the Choco-solver java library. Various constraints are implemented, the objective being to find not only a correct timetable, but also to find a solution that meets some of the expectations that a student or a teacher would have for such a timetable.

Contents

1 Problem definition

Given the following data :

- Number of weeks
- Number of days per week
- Number of timeslots per day
- Number of courses
- Number of lectures per course

The goal is to assign lectures to timeslots with respect to some constraints. Said constraints are stated in the following section.

2 Constraints

The implemented constraints are as follows :

1. There is at most one lecture assigned to a time slot.
2. Lectures of a course are ordered, meaning the first lecture happens before the second.
3. There is a maximum and minimum number of timeslots between two lectures of the same course.

4. There is a maximum number of different days of the week for a given course.
Ex : lectures of course A are scheduled on at most 2 different week days, such as Mondays and Wednesdays.

5. Maximum number of weeks for a course. Said differently, all lectures of a given course must be completed in a certain weekspan.

6. All lectures of a course should be scheduled at the same period of day.
Ex : lectures of course B are only scheduled on the morning timeslot of a day.

2.1 Parameters

The *maximum* and the *minimum* we referred to in the previous section are set as model parameters in the `ConstraintParameters` class as follows :

- `minDisInSlots/maxDisInSlots` : See constraint n°3.
- `maxDiffDaysForACourse` : See constraint n°4.
- `maxWeeksForCourse` : See constraint n°5.

3 Models

3.1 Boolean model

This model makes use of a single matrix of boolean variables we call X_{ij} where $i \in [1, numberOfWeeks \times numberOfDays \times numberOfTimeslots]$ and $j \in [1, numberOfCourses]$.

The interpretation of this matrix is the following : the first column corresponds to all timeslots. Each variable of the first column is either set to 1 if there is a lecture of the first course scheduled at that timeslot, 0 otherwise.

Only the first three of the previously defined constraints are implemented for this model.

3.2 Integer model

For this model, we work with three matrices of integer variables :

- W_{jk} indicates the week the k-th lecture of course j is scheduled on.
- D_{jk} indicates the day the k-th lecture of course j is scheduled on.
- T_{jk} indicates the timeslot the k-th lecture of course j is scheduled on.

where $j \in [1, numberOfCourses]$ and $k \in [1, numberOfLecturePerCourse]$. For the matrices of weeks, days and timeslots the variables are respectively bounded between $[0, numberOfWeeks-1]$, $[0, numberOfDays - 1]$ and $[0, numberOfTimeslots - 1]$.

In order to define constraints 1 and 3 and print a solution, we also needed to flatten the three matrices into an array. The formula used for that is $w \times (numberOfDays \times numberOfTimeslots) + d \times numberOfTimeslots + t$. The length of this array is the total number of lectures, which is the result of $numberOfCourses \times numberOfLectures$. Each variable of the array represents the timeslot at which a given lecture takes place, so the domain for each variable is $[0, numberOfWeeks \times numberOfDays \times numberOfTimeslots - 1]$.

All constraints defined previously are implemented for this model.

4 Results

In this section, we present the results of time measurements for each model on different instances. The measured time is the duration of execution of the solver to find all solutions. For the experiments, the number of slots per day is always 2 (morning, afternoon), the goal being to mimic our own timetable. Also, the constraint parameters are the same for every instance.

We define I1 as the integer model with all constraints and I2 as the integer model but with the same constraints as the boolean model. This is done to make the comparison possible between the two models.

weeks	days	timeslots	courses	lectures	time(I1) (s)	time(I2) (s)	time(Boolean Model) (s)
2	3	2	2	2	0.047	0.091	0.051
3	3	2	3	3	1.127	16.732	12.075
3	4	2	5	4	39.371	-	-
3	4	2	4	4	110.519	-	-

The following table shows the number of solutions found by the solver with both models.

weeks	days	timeslots	courses	lectures	N° solutions (I1)	N° solutions (I2)	N° (Boolean Model)
2	3	2	2	2	138	884	884
3	3	2	3	3	16,128	1,663,416	1,663,416
3	4	2	5	4	114,720	-	-
3	4	2	4	4	1,374,096	-	-

5 Conclusion

From our observations and experiments, we can conclude that the I1 is the fastest model and produces better timetables, given that he enforces more constraints than the other two models. But when we implement the same constraints for both the integer model and the boolean model, we find that the first is slower than the second. Further results for the boolean model and the I2 are not shown here due to the magnitude of time needed by those models to enumerate all solutions. One hypothesis for the resolution time of I1 is that by enforcing more constraints, this leads to less valid solutions being found, hence the faster resolution time for enumerating all solutions.