Academic Scheduling In a College With Multiple Courses, Multiple Disciplines Based on the Staff's Availability

Braintastics 6, IIT Palakkad

Abstract

With start of each new session every college has to prepare a timetable for courses. For a college with many courses, teachers and departments, it becomes a cumbersome task to prepare time table manually and it consumes considerable amount of time. Unambiguously, an automatic teacher scheduler would reduce manual labour for timetable preparation and would give best possible schedule based on restrictions imposed on courses and teachers.

1 Theory

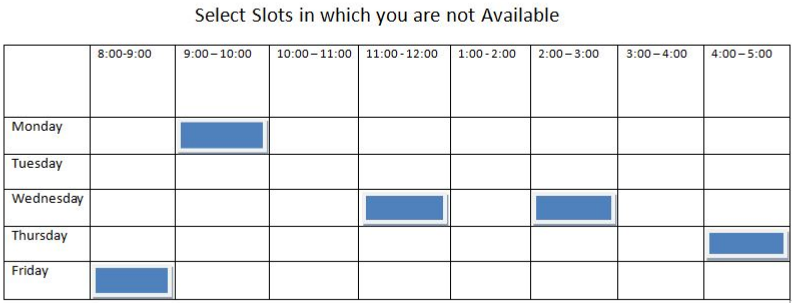
Academic Scheduling problem is theoretically an NP-Hard problem and along with constraints imposed and non-Boolean variables it can be modeled as a SMT (SAT Modulo Theories) problem. There are many SMT solvers available to solve this class of problems. Z3 is one of them. It is an open source high performance theorem prover developed at Microsoft Research. We can model our problem as an SMT optimization problem and use Z3 to get a schedule (refer to mathematical model section for model details).

2 Software

2.1 GUI

1. The GUI allows university administration to add Courses, corresponding Teachers and corresponding Study Groups.

2. Teacher can select slots on which he/she is unavailable.

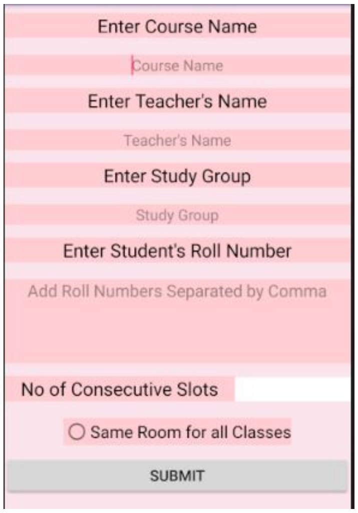


Select Slots in which you are not Available

	8:00-9:00	9:00 – 10:00	10:00 – 11:00	11:00 - 12:00	1:00 - 2:00	2:00 – 3:00	3:00 – 4:00	4:00 – 5:00
Monday		🟦						
Tuesday								
Wednesday				🟦		🟦		
Thursday								🟦
Friday	🟦							

3. If a teacher wants he/she can demand for the same classroom for all classes of his/her course.

4. Teacher can demand for consecutive slots (for e.g. lab sessions etc.).



Enter Course Name

Course Name

Enter Teacher's Name

Teacher's Name

Enter Study Group

Study Group

Enter Student's Roll Number

Add Roll Numbers Separated by Comma

No of Consecutive Slots

○ Same Room for all Classes

SUBMIT

5. The GUI has features where teachers can express personal choices for time slots (e.g. he/she can at most take 2 classes per day). These can be modeled as hard constraints (refer to mathematical section for details).

2.2 Requirements

For consistent performance of software following requirements are necessary

1. No two classes should be scheduled at same room at same time.

2. All time slots assigned to a teacher must be distinct.

3. Scheduling should be consistent with the availability of teacher i.e. No teacher should be assigned a time slot in which he/she is unavailable.

4. No two course having common students in their Study Groups, can be scheduled at the same time.

5. The teacher should be assigned time slots according to specification mentioned in the GUI and should be accordingly burdened. A teacher can select an option in GUI under which his/her schedule is evenly distributed in week to reduce burden on him/her.

6. Classes for students should be evenly distributed over the week to reduce burden on the students.

7. Room should be allotted according to specifications of the course mentioned in the GUI (for e.g. having option of having same class room for all classes of a course).

Following the above specifications not only makes our software results correct but it also makes it considerate towards students and teachers and reduces workload for both of them.

3 Mathematical Model

To model our requirements, we have some hard constraints and some soft constraints. Weekly Timetable is divided into time slots of equal length. Suppose we have d days and l time slots per day, total number of slots will be $d * l$. Each $(Course, Teacher)_i$ has variables t_{ij} and r_{ij} which represents the time slot required and room required respectively by that course, where $j = 1$ to N_i and $1 \leq t_{ij} \leq d{*}l$, here N_i is number of slots required for that course and $1 \leq i \leq M$, where M is number of courses. $1 \leq r_{ij} \leq R$ where R is total number of rooms available.

SG_i denotes the set of student's roll number that are enrolled in i^{th} course.

3.1 Hard Constraints

1. A teacher taking more than one course should not have conflicting slots allocated for his/her courses.

   Suppose T_k denotes set of all time slots (t_{ij}) where i^{th} course is taught byk^{th} teacher and j^{th} time slot for the course $(1 \leq j \leq N_i)$ is allocated to k^{th} teacher , then all elements of T_k must be distinct.

2. Let N_k denotes set of time slots for k^{th} teacher for which he/she is unavailable.

   $T_k \cap N_k = \Phi$ (Teacher should not be allotted slot for which he/she is unavailable).

3. Two different Courses, having classes scheduled at the same time slot should not be assigned same room.

Mathematically, for all courses C_l and $C_m(l \neq m)$, $\forall i, j$, $i \neq j$, $t_{li} = t_{mj} \Rightarrow r_{li} \neq r_{mj}$.

4. For any two courses, if their student group has common students then they should not be scheduled at the same time slot.

   Mathematically, for all courses C_l and $C_m(l \neq m)$, $SG_l \cap SG_m \neq \phi \Rightarrow \forall i, j$, $i \neq j$, $t_{li} \neq t_{mj}$.

3.2 Soft Constraint

1. if k^{th} Teacher selects option of having his/her time slots evenly distributed throughout the week, $\sum_{t_{lm}, t_{ij} \in T_k} |t_{lm} - t_{ij}|$ should be maximized

2. Students should not be burdened i.e. they should have a timetable fairly distributed over entire week. For this software clusters Courses based on Study Groups and after clustering each Course belongs to a cluster, this cluster signifies a group of students who have a lot of common courses. All courses has associated time slots with them so if we maximize $\sum_{i=1}^{\varpi} \sum_{t_{li}, t_{mj} \in \Omega_i} |t - t_{mj}|$,C_l, C_m belongs to the same cluster Ω_i and ω is total number of clusters obtained, we make our time table more distributed.

All these constraints can be modeled in Z3 and optimal scheduling can be found.

4 Example

A toy example which has following features

- We have 2 days and 5 slots per day. So total we have 10 slots.
- We have 3 rooms (from Room0 to Room2).
- There are 3 Teachers and Courses.
- First teacher teaches 2 courses, each course needs 3 slots.
- Second teacher and third teacher teaches 1 course which needs 3 slots.
- Here SG_i denotes Study Group for course i.

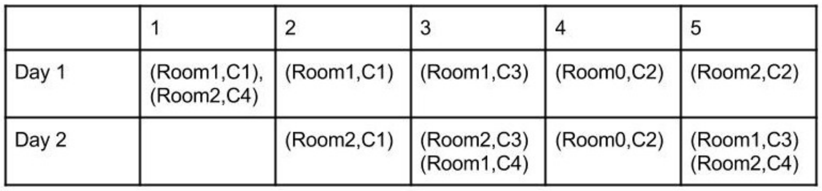
  $SG_1 \cap SG_2 = \phi$

  $SG_2 \cap SG_3 \neq \phi$

  $SG_1 \cap SG_3 \neq \phi$

  SG_4 is disjoint from all other Study Groups.
- The Clusters are $[C_1, C_2, C_3]$ and $[C_4]$.
- We consider reducing burden for students and teachers, while solving this problem.

Result of example using primitive software:-



	1	2	3	4	5
Day 1	(Room1,C1), (Room2,C4)	(Room1,C1)	(Room1,C3)	(Room0,C2)	(Room2,C2)
Day 2		(Room2,C1)	(Room2,C3) (Room1,C4)	(Room0,C2)	(Room1,C3) (Room2,C4)

Slots allocated to teacher

- Teacher 1 : 1, 2, 4, 5, 7, 9.
- Teacher 2 : 3, 8, 10.
- Teacher 3 : 1, 3, 5.

5 Conclusion

Thus our software not only outputs a schedule but it has features which makes it highly flexible and considerate towards teachers and students. It will surely help any institution in making timetable and would reduce manual work.

6 References

1. http://www.cs.tau.ac.il/~msagiv/courses/asv/z3py/guide-examples.htm
2. https://ocharles.org.uk/blog/guest-posts/2013-12-09-24-days-of-hackage-sbv.html