RCSR Report TP2

Tiago Soares 52965 & Xavier Pacheco 52748

Phase 1

We will begin by giving an overview of some of the more complicated rules we used to optimize the first phase of the project.

:-#sum{N, R : room(R, N), courseRoom(C, R)} > E + N2, enrolled(C, E), room(R2, N2), courseRoom(C, R2).

This rule states that we can't allocate more seats than are strictly needed.

:- courseRoom(C, R1), assign(C, D, S), not roomAssign(R2, D, S), room(R1, N1), room(R2, N2), N1 < N2, M!=N2, M!=#sup, M=#min{N : room(R, N), courseRoom(C, R)}.

This rule states that a model is rejected if it allocates smaller rooms when bigger ones are available.This rule is relaxed for the smallest room (because it might not be fully filled).

Phase 2

For the 2nd phase we chose to extend our program to also minimize the number of students with tests on consecutive days. We did it by adding the following minimization.

#minimize{N @ 2, C1, C2 : assign(C1, D1, _), assign(C2, D2, _), overlap(C1, C2, N), D1=D2+1}.

We also introduced the concept of a watcher. Due to this, for our project to run, you must also use the prof01.lp file. A watcher is the Professor responsible to watch the students in a given room. Additionally, this professor cannot teach the course it is supervising. There is one and only one watcher per room and a watcher cannot supervise more than one room at any given slot.

```
1{watcher(P, R, C) : professor(P, _) }1 :- courseRoom(C, R).
%Há um e um só watcher por sala.

:- watcher(P, _,C), professor(P, C).
%O watcher de uma sala não pode lecionar esse curso

:-watcher(P, R1, C1), watcher(P, R2, C2), assign(C1, D, S), assign(C2, D, S), C1 != C2.
%Um watcher não pode vigiar duas salas ao mesmo tempo
```

Test Files

The prof02.lp file has professors that all teach the same course, when running this file with any other that has the course “am1” it will not find any satisfiable models seeing as a teacher cannot supervise his own test.

The config05.lp has only one day with one slot, running this with any tests with overlap will cause the program not to find any satisfiable models, due to the restriction that a student can only do one test per day.