POLITECNICO
DI TORINO

Exam Time Table
Report

Group 3 [MA-ZZ]

Jacopo Maggio
Stefano Munna
Jacopo Nasi
Andrea Santu
Marco Torlaschi

8 gennaio 2018

1 Report

This document provides some explenation about our solution to the problem proposed. The project is developed with C with support of CLion (JetBrains IDE) and GitHub.

1.1 Data Structure

The main structure is a graph, build on a adjacency matrix. This matrix have value:

$$adjM_{e1,e2} = \begin{cases} 1 \rightarrow \textit{If } \boldsymbol{e1} \textit{ and } \boldsymbol{e2} \text{ can't be sustained in the same ts} \\ -1 \rightarrow Otherwise \end{cases} \tag{1}$$

The choice over this structure is related to its ease, it allows a fast reading of the instances, also the benchmark and feasability become easy to check.

1.2 Workflow

The first problem to be solved is to finding a **feasible** solution to be used for the future improvement.
Our solution, is divided in two parts, a greedy data preparation and a tabu search. Using only the tabu, that will work anyway, will require too much time for finding the first feasible solution.

Greedy The first phase try do generate a feasible solution adding one at the time the exam. The idea is to reduce "the complexity" of the problem. It start from a sorted data, from the exam with more collision to the fewer ones. The flow is the following:

1. Use the first available timeslots that not generate conflicts for each exams.
2. If there aren't enough timeslots it will add a new timeslot.

Of course adding more timeslots not fit the problem constraints. These adding will be solved with the next step. The advantages of these implementations are the reduction of computational time and the easiness to implement it.

Tabu Search The goal of the second step is to reduce the number of added, by the previous step, timeslots to the correct number, generating a feasible solution. The workflow is:

1. Decrease the number of timeslots.
2. Try to resolve the conflict.

- If not: Backtrack.
- If yes: Restart from point 1.

Using the tabu search is a really reliable, the main problem is its slowness that is partially solved by the greedy step.

The next step try to improve our founded solution by using some metaheuristic algorithms.

1.3 Execution