Project Report - Group 24

Antonella Cardillo Maria Castaldo Marian Alexandru Ilies
Fabio Gavinelli Nicola Sabino

Optmization Methods and Algorithms - 2017/2018

1 Introduction

This report aims to present a possible way to face the time-tabling problem. As long as both hard and soft contraints are involved, the issue has been divided into two subparts: the first one consists of finding feasible solutions while the second one deals with the reduction of penalty that should be paid if conflicting exams are placed in sufficently near timeslots. Therefore the algorithm presented firstly ensure satisfaction of the conditions for a feasible timetable, and then optimizes the number of desiderable but not essential constraints. The next sections of this report are meant to deepen the techniques used.

2 Initial Population Generation

The initial population is generated using a greedy scheduling of exams.Instead of randomly choosing the next examination to allocate, the examinations are scheduled sequentially according to their difficulty. A low-level heuristic is used to assess the difficulty of examinations. The order used to place them into the timetable is based on their *Saturation Degree*, namely the number of timeslots in which the exams e can still be placed after the scheduling of the previous exams. This low-level heuristics, compared to other that have been tested (such as *Largest Degree*, i.e. the number of conflicts an examinantion is involved in, *Largest enrolment*, i.e. the examination with the largest student enrolment is the most difficult to schedule), gave the best results, rarely causing conflicts. In order to keep the variability of the generated scenarios , the algorithm starts from a random exam and, any time more than one exam has got the lowest number of possible timeslots in which to be scheduled, the algorithm randomly selects within them which one should be allocated. A pseudocode of this algorithm is shown below

```
min_nr_of_possibilities = tMax+1;
for each exam e
        nr_of_possibilities(e) = number of timeslots where it still can
            be scheduled
        if nr_of_possibilities(e) < min_nr_of_possibilities
                min_nr_of_possibilities = nr_of_possibilities
                clear the list of choosable_exams
                add exam e to the list of choosable_exams
         else if  nr_of_possibilities(e)== min_nr_of_possibilities
                add exam e to the list of choosable_exams
end
randomly choose an exam e_j within choosable_exams
place exam e in the first timeslot available
```

As the previously presented heuristic might cause conflicts, a Tabu Search algorithm is called to guarantee the feasibility of scenarios. At each iteration it evaluates a randomly choosen set of neighbors (new scenarios obtained by moving an exam from its current timeslot to a randomly selected one). After the evaluation, the neighbor with the least number of conflicts is kept as the new scenario. In order not to explore only a subregion of space a tabu list is introduced: as long as

the numebr of conflicts of the choosen scenario is not less than the previous one, previous moves cannot be done again .Worsening moves are allowed in order to escape from local minimum.

3 Genetic Algorithm with Local Search

Once the original population has been generated, a *Genetic Algorithm*, combined with a *Local Search*, has been implemented to minimize the penalty of solutions. The addition of local search to the normal genetic operators inevitably has some computational expense but this can be justified by the reduction in search space that must be explored in order to find the optimum solution.

3.1 The Evolutionary Operator

Improvement of feasible solutions is performed by genetic algorithm through three different operators. Each element of the population has a uniform probability to undergo each of the three methods.

- **Light Mutation**: A random exam e and a random timeslot t is selected. If exam e doesn't create any conflict in timeslot t, then it is moved from its current timeslot to the new one.
- **Heavy Mutation**: Two random different timeslot are choosen. The exams assigned to those timeslots are then swapped. This operator always grants feasibility as long as only set of exams that generate no conflicts whithin them are moved. It has shown to play a significant rule, expecially at the beginning of the algorithm as the heuristic used to generate the original polulation favours specific timeslots.
- **Exams displacement**: This operator tries to disrupt a random period in the timetable. After choosing a random timeslot, it tries to reallocate all possible exams. Eventhough it rarely improves solutions on its own, it gives good results combined with a local search operator.

3.2 Local Search Operator

This algorithm is used after each of the Evolutionary Operators in order to locally explore the space of solutions and try to reach all possible local minumums. It can be summarized as follows:

For each exam e

- Evaluate the cost of placing the exam in each timeslot, given that the other events are fixed.
- Schedule e in the period causing least penalty

3.3 Selection Method and Fitness Function

After evolutionary and local operators generated new scenarios, a deterministic selection is applied to the population. The best half scenarios are selected. However it would be risky to copy the same scenario more than once, as, in the long run, it could lead to a population filled with just one individual. Therefore results of evolutionary operators are kept only if not already present in the population.

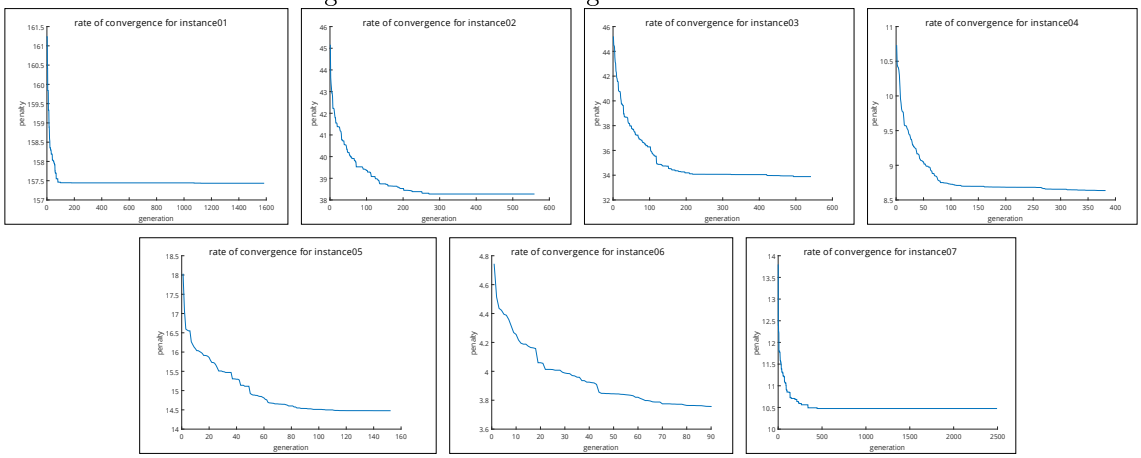
4 Parameters and Results

The results, reported in the Excell file with the benchmarks, have been analyzed also in terms of rete of convergences. Figure 1 shows how fast each istance converges using the algorithm presented in the previous sections.

Some parameters might play an important role in the final results. Both the *Tabu Search* and the Genetic Algorithm receive parameters:

- the Tabu List length

Figura 1: Rate of convergence of each instance



- the Size of the Neighborhood that should be explored
- the Size of the Generation

Since Tabu Search is rarely used, only the Size of Generation can be relevant in this analysis. In the table that follows some t-test have been performed in order to evaluate the meanfullness of the difference in mean between samples of 10 results obtained for different istances with different parameters.