

International Timetabling Competition

The solution is reached with the use of Evolutionary Algorithms. In order for the chromosome to be evolved the GAJLib library [1] for Evolutionary Algorithm's was used in Linux environment.

The Chromosome's structure was the number of Periods, containing each one of them the number of Rooms. Each Room could host a number of exams at each period, which was calculated by the size of exams at each dataset.

The Chromosome was initialized with the use of 3 Vectors so as to avoid the dependencies between students (students having more than one exams at the same time).

The steps of the algorithm's initialization are listed below

1. Creation of a Vector with Vectors, named V,(it is an implementation of ours, containing an id which shows the exam number). The size of the array is the number of the exams.
2. Each Vector has a number of exams containing the dependencies of exams with the id.
3. In addition there are two more Vectors. One of them, named randomSelect, keeps all the ids that exist in the first Vector and the other one, named notInserted, keeps all the ids that are not inserted in the Chromosome from the first Vector.
4. We choose randomly from the randomSelect Vector and we check in the Chromosome if the exam we will add has dependencies with the other exams that already exist in the same Period. If the exam does not have dependencies is added.
5. If the exam is added then we remove that element from Vector V.
6. When the randomSelect Vector is Empty we move on to the next Period and we clone the notInserted Vector to the randomSelect.

The steps from 4 – 6 are repeated until the size of Vector V equals to zero.

In order to run the algorithm the command

```
./evolve <*.exam>
```

should be typed.

The configuration used is listed below

Mutation: Swap Mutations 0.1

Recombination: PMX Recombinator 0.9

Selection: Tournament Selection

Elitism: true

References

1. <http://www.iit.demokritos.gr/~anskarl/pub/>