

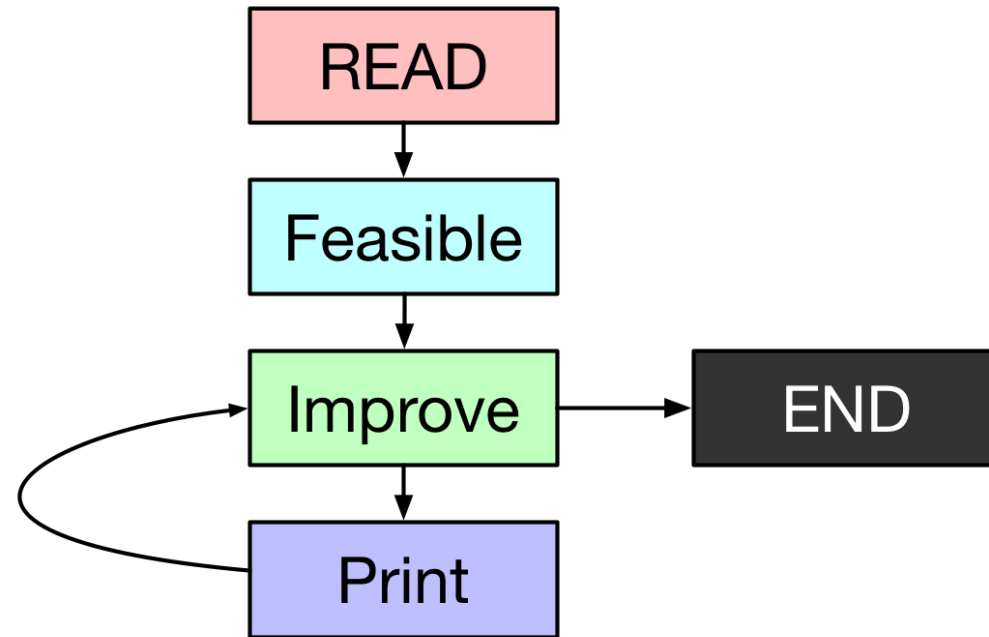
OMA Assignment 2017-2018

Examination Time Tabling : Group 03

Jacopo Maggio – Stefano Munna – Jacopo Nasi – Andrea Santu – Marco Torlaschi

The Problem

- Solving a time scheduling problem trying to minimize the penalties of exam proximities.²
- Step:
 1. Read & Store data input
 2. Finding a feasible solution
 3. Improving found feasible



Data Structure

- The software is written in **C** due to its high performance.
- The main structure is a **GRAPH** built on a adjacency matrix.
- When two exams CAN'T be held in the same time slot $\text{adjM}[e1][e2] = 1$, otherwise -1.
- The TABU is of 1000 moves with 7 iterations.



Feasible

Finding Feasible

- The initial idea was to implement a **TABU SEARCH** over the data. This was a little bit slow in case of high number of exams.
- The final version use a **GREEDY** algorithm to reduce the complexity and than it pass this partial solution to the **TABU** implementation.
- From ~3 minutes to ~10 seconds.

Greedy

- Trying to **reduce** "the complexity" of the problem.
- Initial data is sorted from the exam with more collision to the the fewer ones.
- **Workflow:**
 - Use the FIRST available (no collision) time slot for each exam.
 - If there aren't enough timeslots with this configuration it adds more timeslots.
- The added timeslots will be removed with the next step.

Tabu Search

- It reduces the number of added (by the greedy) time slot till the correct number.
- TO DO...



Improve

Used techniques

Local Search

Local Swap

Simulated Annealing

Improve

Other techniques tried but not used in the final
version...

Benchmark Insertion

- Starting by selecting a random timeslot and then adding one at the time the best one by checking the benchmark of that temporary solution.
- **ADVANTAGES:**
 - Avoid "local minimum" problem
 - Fast and easy to be implemented
- **DRAWBACKS:**
 - Too random

Genetic

- Starting from two or more parents it performs a cross-over between them and occasionally a mutating.
- This algorithm not fit very well this problem because it could generate a lot of unfeasible solution due to its randomness.
- **DRAWBACKS:**
 - Not too easy
 - Lot of unfeasible solution

A large red speech bubble graphic with a white outline, pointing downwards. The word "Performance" is written in white text inside the bubble.

Performance

Repository

- The whole project with code, math model, instances, presentation and benchmark is available in the **GitHub** repository:



https://github.com/Jacopx/OMA_ExamTimeTable



Thanks for the attention!