U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Meta Heuristics and Optimization Problems: Mentorship and Teamwork

Carlos Eduardo Coelho Veríssimo - 201907716

Frederico Manuel Lopes - 201904580

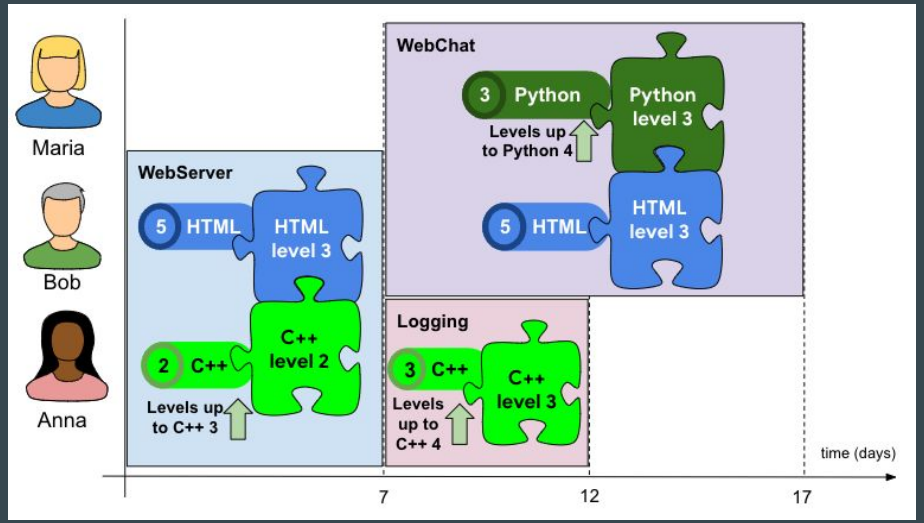
Nuno Miguel Carvalho de Jesus -201905477

Work specification

The selected theme was Optimization Problems.

More specifically, our project is an adaptation of Google Hash Code 2022 for Teamwork and Mentorship scheduling.

The goal of this project is to assign members to roles in projects based on a set of skills needed.



Formulation of the Problem

Solution Representation:

- List of lists with the specification of who will work in which project. **Ex**: [[1,1,0], [0,1,1]] (project 1 - assigned to 1st and 2nd person, project 2 - assigned to 2nd and the 3rd).

Neighborhood:

- Switch the participation of 1 person in 1 project.
- Switch the participation of 2 person in 2 random projects.
- Randomize between the 2 options above.

Hard Constraints:

- The number of projects.
- The skills needed to complete a project.
- The duration of each project.

Evaluation Function:

- The sum of the score of all completed projects minus the number of members in the solution that are not needed for the project completion.

Implementation work 1/2

The selected language to picked for development was **Python**.

Implemented Algorithms:

- **Hill Climbing**,
- **Simulated Annealing,**
- **Genetic Algorithm,**
- **Tabu Search.**

Data structures used:

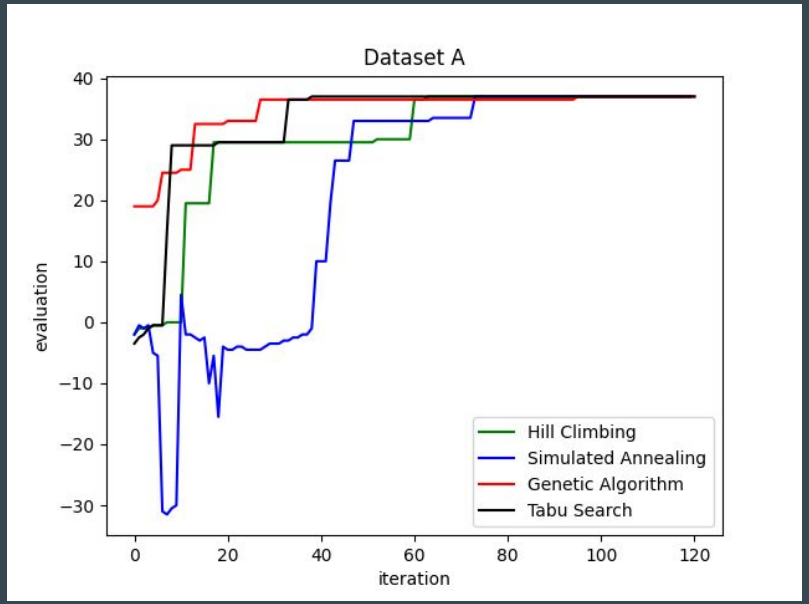
- **Skill**: Specifies a level and a name of a certain ability.
- **Member**: Indicates the name, set of skills, if they are assigned to a project or not and which one they are assigned to.
- **Project**: Defines the name, duration, score, list of roles needed for a project and if it has started/ended.
- **Team**: Groups the dataset together, i.e, has a list of members and a list of projects.

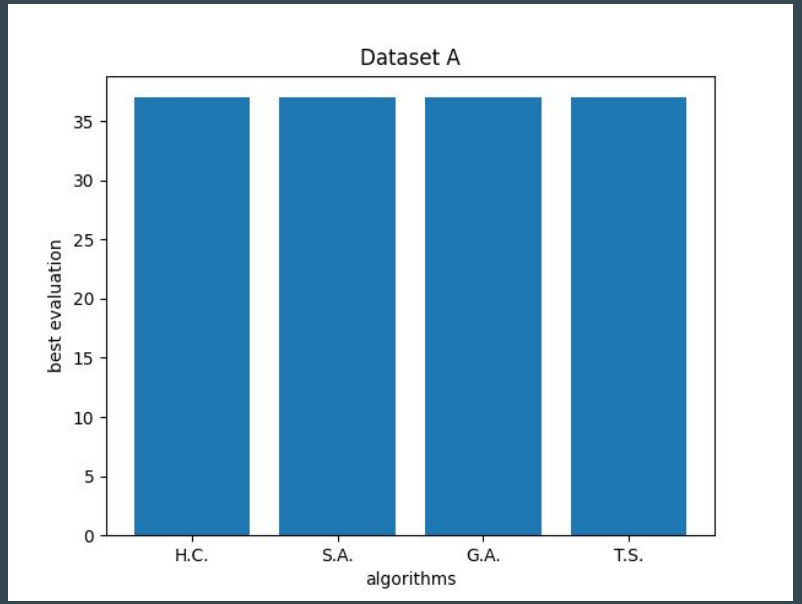
Implementation work 2/2

Evaluation function:

- Infinite cycle that, in every iteration, loops through the projects and checks:
  - If the project has started:
    - Increment the day count of that project.
    - If the day count reaches the project duration, the project ends and the members assigned to it are released.
  - If the project has not started:
    - Check if there are conditions to start it in this iteration. If so, initialize the day count and lock the members needed for the project.
  - If there are no ongoing projects, break the cycle.
- Function returns the result of ***score - penalties***.
  - **Score:** The sum of the score of all completed projects
  - **Penalties:** Worsens a solution evaluation if there are members in the solution that are not needed for the project completion.

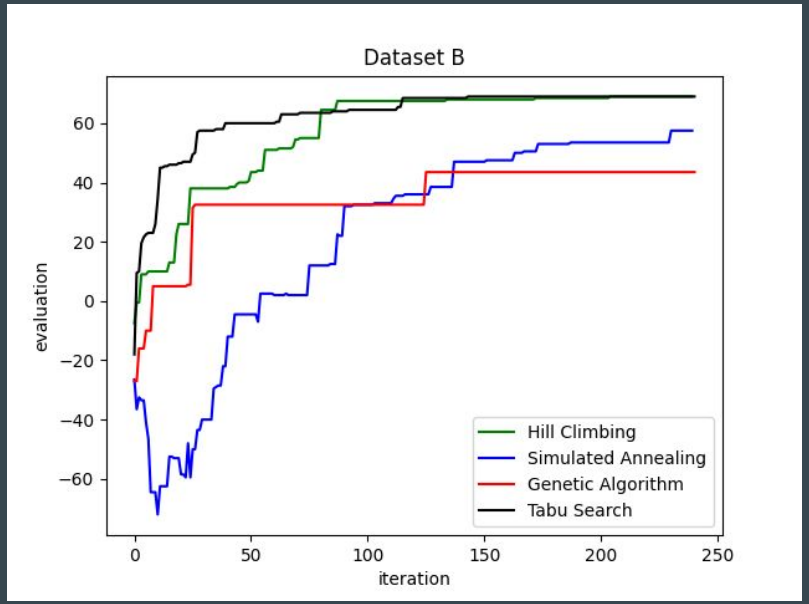
Experimental Results 1/4

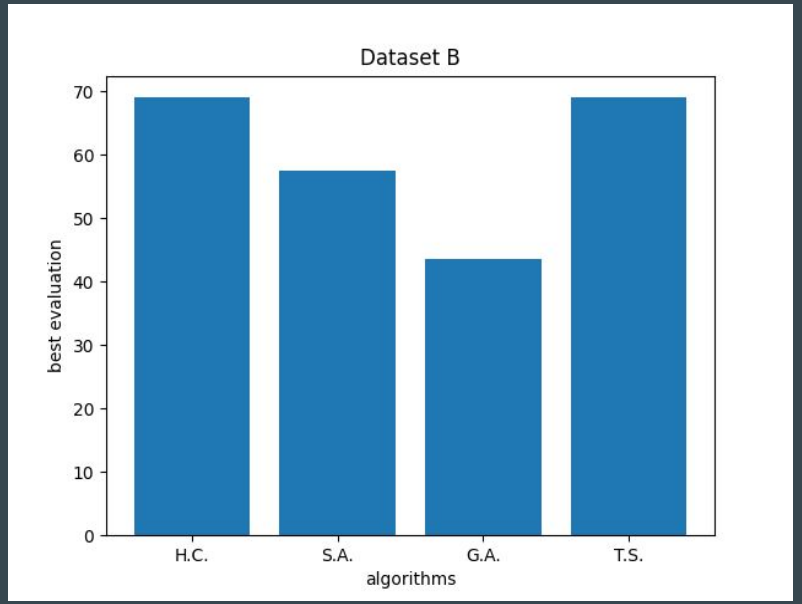




- Cooling algorithm chosen: Exponential multiplicative
- Population size for the genetic algorithm: 20
- Parent selection: Tournament Selection
- Crossover function: 3

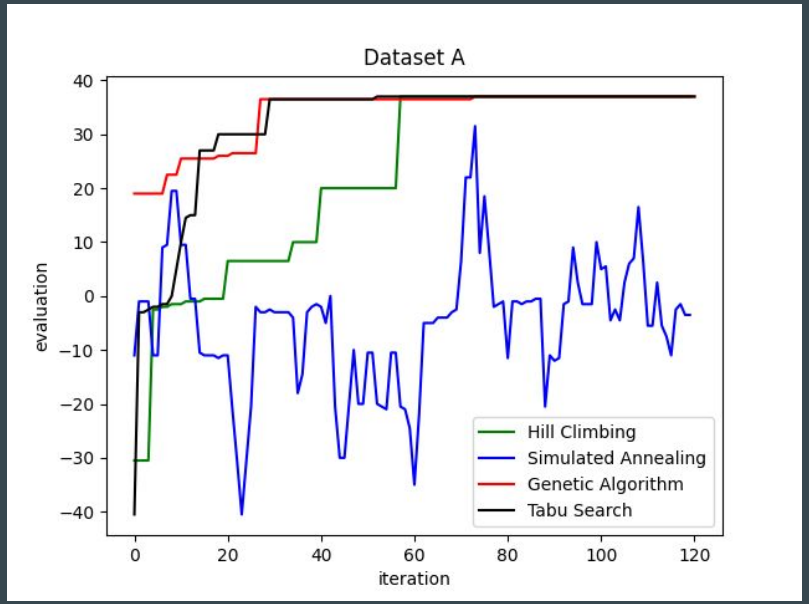
Experimental Results 2/4

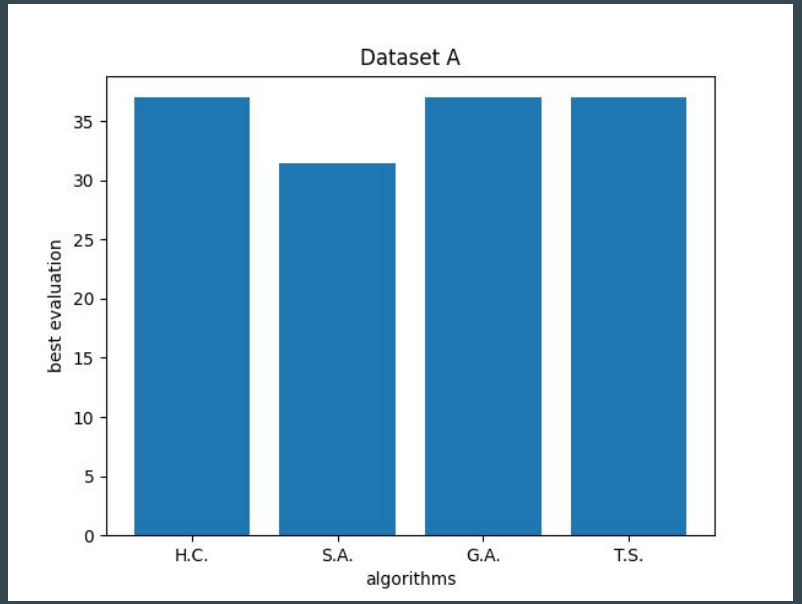




- Cooling algorithm chosen: Quadratic multiplicative
- Population size for the genetic algorithm: 20
- Parent selection: Tournament Selection
- Crossover function: 3

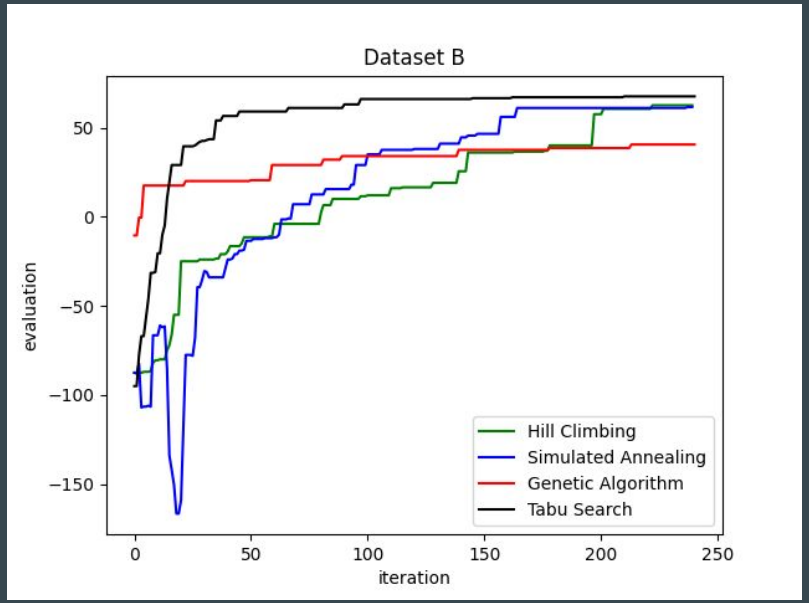
Experimental Results 3/4

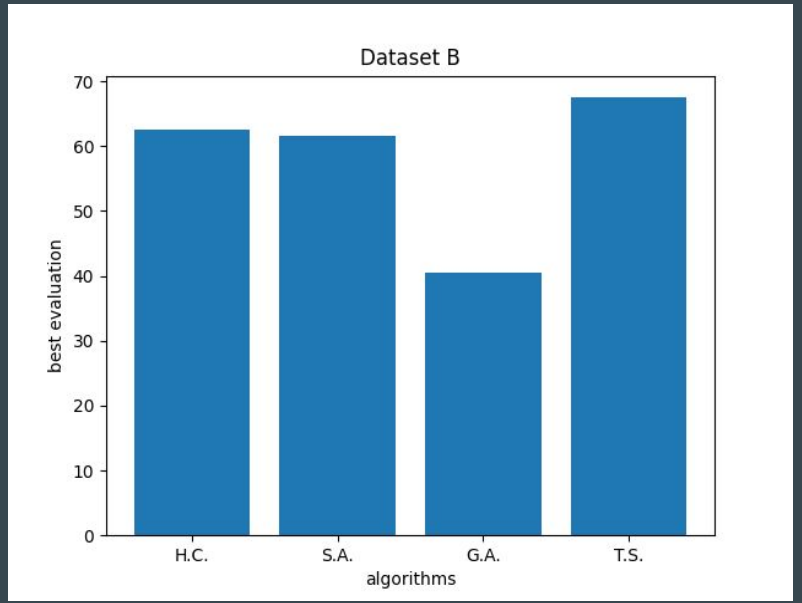




- Cooling algorithm chosen: Logarithmic multiplicative
- Population size for the genetic algorithm: 20
- Parent selection: Tournament Selection
- Crossover function: 3

Experimental Results 4/4





- Cooling algorithm chosen: Exponential multiplicative
- Population size for the genetic algorithm: 20
- Parent selection: Roulette Selection
- Crossover function: 3

Conclusions

Using the practical and theoretical work we were able to implement the algorithms without many hiccups. By developing this project, we were given the chance to deepen our knowledge about Optimization Problems and their corresponding algorithms. The path wasn't smooth, but it was definitely worth it, if we think about we learned so far.

We feel that we fulfilled everything that was proposed and that we did it in a communicative and objective way.

Related work and References

- Russell, S., Norvig, P. and Chang, M., n.d. Artificial intelligence: A modern approach. 3rd ed.
- https://www.geeksforgeeks.org/genetic-algorithms/
- https://towardsdatascience.com/optimization-techniques-tabu-search-36f197ef8e25
- https://machinelearningmastery.com/simulated-annealing-from-scratch-in-python/