Final Project

Optimization Algorithms

Using GAs for problem solving

April 10^{th} - May 26^{th}

NOVA IMS

2023

Contents

1 Project Overview

This project aims to evaluate your knowledge acquired during both the practical and theoretical classes of Optimization Algorithms while also assessing your researching capabilities and your capability of solving problems in a creative and technical way.

The groups are build by 3 to 4 students and are chosen by yourself. You must Submit your groups on the corresponding moodle link from ***April 10th*** to ***April 14th, 23:59h***. Groups who enrol after the enrollment deadline (or deliver the project without enrolling) will be given a 10% penalty on their final grade. If you do not have a group, a specific section on moodle will be available to you where you can sign up individually and have a group created for you by the professors.

The deadline for the delivery of this project is **May 26$^{\mathbf{th}}$**. Groups who will not respect this deadline will suffer a penalty of one value for each day of delay (up until 5 days, afterwards the project will not be accepted and the students will receive a project evaluation value of 0).

All projects will be defended orally during the schedule of the OA classes on **May 29$^{\mathbf{th}}$** and **May 30$^{\mathbf{th}}$**.

A detailed schedule will be provided to you in the month of May. In the project defense, you will be asked questions regarding your implementation of the project as well as questions about the theoretical concepts.
All members of the group will be asked questions and students who will not be able to provide satisfactory answers will be discounted.

Projects will be delivered through Moodle (using the given delivery link).

A specific list of the elements of the project delivery and their corresponding rules are provided in Section 4.

Make sure you read every section carefully!

2 Project Description

Your goal in this project is to further improve the base implementation of Genetic Algorithms (GAs), that was developed with and by you during the semester. More specifically, you will be asked to research, implement and test different Genetic Operators (specifically different types of crossover and mutation) in order to solve an Optimization Problem (OP) that is given to you. Additionally, you will need to adapt/add-to the current implementation's code so that the representations/constrains of the OP are respected.

Your imagination, research and implementation abilities are going to be tested! You will be evaluated not only by the quantity and quality of your different implementations but also on their corresponding creativity and difficulty levels.

You will be given all the information you need with regarding to the OP that you need to solve. However, no data will be given to you. At the end, the final model you propose will be run by us (on a previously established data that, as previously mentioned, will not be given to you) and you will be given a model performance score based on the comparison of your model's performance with the performance of other groups' models.

Your model's performance will not be the only evaluation component that will be obtained by the comparison of your work with the work of the other groups. Your implementation score will also have a comparative component in which your implementations are compared to the other groups (see Section 3).

The specific requisites for the project (and their further explanation) are found below. Make sure you read the rules carefully!

3 Project Rules

Overall you are requested to:

- Implement all the necessary functions/elements for the representation of the OP, respecting its constraints.
- Implement **at least three** different mutation operators
- Implement **at least three** different crossover operators
- Propose a final model (with the specific operators and parameters used) that is the result of a grid search of at least 15 runs.
- **Note:** Since you are not given the data but just its format (see Section 3.2), it is imperative for the performance of your model and its robustness that you don't only run different configurations but also use different random values for your data!

3.1 Project Evaluation

Your final grade will be given in the following way:

- **Project Report - 20%**. In the report you must describe the methodology used, provide the literature review of Genetic Operators that you used and present the results (with plots) of your final proposed model

- **Implementations - 45%**. This part of the evaluation corresponds to the quality and efficiency of your implementations. Note that quality refers to both the quality of your code and the complexity/originality of the genetic operators used.

- **Project Methodology and Requirements - 10%**. This part of the evaluation corresponds to the quality of your methodology with regards to the grid search strategies you chose and your overall project organization, as well as your compliance with the rules

- **Implementations (competition value) - 15%**. This part of the evaluation corresponds to the comparison of your implementations with the ones provided by the other groups. You will be given a score based on:

  - The number of operators you implemented in comparison to other groups

  - The complexity of the operators you implemented in comparison to other groups

  - The originality of the operators you implemented in comparison to other groups

- **Model performance (competition value) - 10%**. This part of the evaluation corresponds to the performance of your final proposed model on the unseen data, in comparison to the performance of the models proposed by the other groups.

3.2 The Optimization Problem

Imagine that NOVA IMS contacts you and requests for your help in the organization of their next event. More specifically, they need your help with regards to the organization and planning of the routes that the different groups that attend the event will take, while seeing all the different lectures and exhibitions. You are requested to try and optimize the route the groups will take, finding the path that will minimize the overall loss in focus (in %) before reaching the final room, where a very important lecture will be held.

Overall, the assumption is that moving between different rooms can cause more or less loss in focus. For instance, passing from room A to room C cause less loss in focus than passing from room A to B.

Naturally, you do not know what the loss in focus will be between each rooms, so you must try and create a pipeline that allows the algorithm to learn in the most robust way possible, considering different losses in focus between the different rooms.

NOVA IMS was, however, able to give you some outlines that you can (and must) already consider and keep in mind for your algorithm. The outlines are the following:

- There will be a total of 8 rooms for the event. For now, they are calling rooms by the first 8 letters of the alphabet.

- Room C will not be mandatory in the group's tour, as long as they visit room B **right after** they visit room F, with **no** rooms in between (as they will be of similar contents).

- Room A **cannot** be seen after room F, as the concepts between the two will be too dissimilar. Considering the two previous steps, when trying different distances keep in mind that the loss in focus when moving from A to C has to be at least 4% more than the maximum other loss of focus.

- Room H is the final room of the event and shall only be seen **last**

- **Very important:** You are to organize this data in the form of list of lists (i.e., a matrix) where each list represents a room and contains the losses of focus by moving from that room to all the other rooms.

Note that the goal is to obtain **one and only one route**! The different groups will all take the same route, the one proposed by you!

3.2.1 Example of a valid dataset

An example of a valid dataset respecting the aforementioned constraints can be found below:

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**
A	-	10.2	16	4	6	12	7.5	9
B	10.2	-	4.3	8	11	2.2	3	11.5
C	16	4.3	-	9	3	6	7	5
D	4	8	9	-	11	10.5	5	2
E	6	11	3	11	-	9.8	4	10
F	12	2.2	6	10.5	9.8	-	8	8
G	7.5	3	7	5	4	8	-	10.1
H	9	11.5	5	2	10	8	10.1	-

4 Delivery

The delivery of this project **must** contain the following:

1. Your code. You must **comment** your code well!
   - You must have a main.py file where your final model with the final parameters and operators are ready to run, needing only the insertion of data.
     - More details regarding the constraints and specific details of the optimization problem were given in Section 3.2. It is mandatory that this file contains code ready to simply receive data in the format of a list of lists, so that the only thing needed in order to obtain the output from your model is to insert the data. Keep that into consideration in your implementation!
   - All your methodology, data creation strategies and grid search components must be in the code.
2. A report, as indicated in Section 3.1. Do not forget to use proper citations and tell the store of your methodology, strategies chosen and the final model.

You must deliver a zip file with both your code and the report moodle, using the link that will be provided to you at the time.

5 Final Notes

Make sure you impress us with your implementation of the operators! This means taking time to work on the different operators, researching as many as you can and trying to be as creative as possible with them, not being afraid of challenging task!

You can try different selection algorithms! Be as creative as possible. The more, the better!

Take your time reflecting and deciding on strategies to how to make your model as robust as possible and build your pipeline. Remeber: trash in, trash out!

Make sure you read the project requisites (found in Section 3 and Section 3.2) thoroughly so you don't miss a mandatory requisite!

Good work! ☺