Computational Intelligence for Optimization
Project Description - Draft

Project Objective

The main objective of this project is to explore the Optimization concepts, especially related to the problem design (problem building blocks) and the Genetic Algorithm learned in this course.

In this way, you will design and develop the specific problems from scratch, using ProblemTemplate as a reference, so you must extend the base ProblemTemplate class). Attention: you must extend ProblemTemplate, as we did with Knapsack Problem.

Genetic Programming will be approached with some exercises in the Saturday class. Only Genetic Algorithm will be evaluated in the project.

This document has two sections to explain the project:

1. SECTION ONE: containing rules

2. SECTION TWO: problem description

1. Project Overview

In this project, you must design and develop two combinatorial optimization problems and some other mandatory techniques. The objective is to provide more power to the metaheuristics to search for solutions. The problems are the following:

A) Develop two problems:

- TSP - Travel Salesman Problem
- PIP - Portfolio Investment Problem

Remark: the problems will be described in section 2: Problem Description

Remark: You must extend the class ProblemTemplate, implementing the specific behavior for each problem.

Remember, for each problem, you must design and develop the required building blocks of a combinatorial optimization problem, such as:

- Solution Encoding and representation;
- Objective or Fitness Function;
- Constraints (it will be defined by problem)
- Admissibility function (if the problem requires to check if a solution is admissible regarding constraints);
- Decision Variables (you will receive the data);

Objective (identifying if the problem objective is maximization or minimization);

B) Develop one SELECTION approach (GA)

- Implement the Roulette Wheel for minimization, to use in Travel Salesman Problem.

Remark: additional efforts to develop more approaches and techniques will be considered positively in the final grade.

C) Develop two CROSSOVER approaches (GA)

- PMX - Partial Match Crossover
- Cycle Crossover

Remark: These crossover approaches must be implemented to deal with TSP solution encoding rules.

D) Develop one MUTATION approach (GA)

- Swap Mutation

Remark: This mutation approach must be implemented to deal with TSP solution encoding rules.

E) [OPTIONAL] Try other approaches, go further. Be creative!

Suggestions:

1. Try different selection policies.
2. Try other variation operators (crossover and mutation).
3. Implement population diversity measures and try several diversity preservation mechanisms.
4. Go further. Be creative.

SUGGESTIONS:

Develop Genetic Algorithms INITIALIZATION approaches

- One initialization using the Hill-Climbing (HC) Algorithm
- Other initialization using the Simulated-Annealing (SA) Algorithm

Remark: remember the GA initialization generates a population, so these initializations approaches will run "n" times ("n" = population size)

Restriction:
Maximum Iterations = 500 (for both)

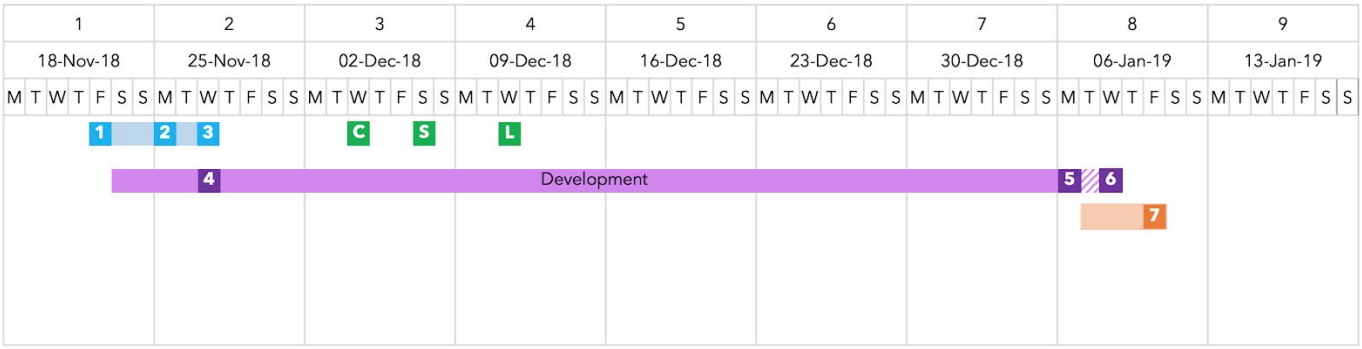
Remark 2:
For Simulated Annealing, remember that you need to calibrate the initial Control Parameter according to the fitness.

Since you need to develop two initialization considering local search algorithms (HC and SA), you will need to develop a neighborhood function for each problem (TSP and PIP). Therefore you will need to develop:

- **One neighborhood function for TSP - Travel Salesman Problem**
- **One neighborhood function for PIP - Portfolio Investment Problem**

Remark: the neighborhood can be big (Number of Neighbors), remember you can set the size of neighborhood set returned by neighborhood function

2. Schedule



Schedule Highlights

1	**Nov 22nd:** the date that you received the project description. Read the document and bring your doubts to the next session. You can start working on research and design for the problems.
2	**Nov 25th:** The composition of each group should be defined until November 25th of 2019. Please, complete this information in the google sheet: Groups Definition
3	**Nov 27th:** Project discussion and definitions. Your opportunity to ask and understand the project.
CSL	(C regular class, S - Saturday class, and L - last class)
5	The delivery deadline is Jan 6th of 2020 (until 23:59). For each day of delay will be discounted 2 (values) points of total possible grade (20).
6	Projects delivered after Jan 8th of 2020 will not be accepted.
7	The defense of the project will occur on Jan 11 of 2019. The final defense schedule will be available by the 5th of January*. Groups of students that do not make the project defense will not pass the exam.

3. What you will receive (we will build together):

A genetic algorithm (GA) class, containing a standard version of GA. You can improve the performance, calibrating the parameters of GA and developing new Genetic Operators or Approaches.

You also receive other classes, like **ProblemTemplate** Class, containing the base implementation of any combinatorial optimization problem for GA. To implement TSP (Travel Salesman Problem) and PIP (Portfolio Investment Problem) you must extend this class as we did with Knapsack Problem in our classes.
[check the class diagram attached]

You also received the problems implemented in our practical classes (labs). You can use these source codes as a reference to create the two specific problems that you need to implement.

An initial application to configure your search task using GA.

The GA has a logger class (utility) to save the history of the generations.

Data to develop / test Travel Salesman Problem

Data to develop / test Portfolio Investment Problem

4. Deliverables

I. **Report:** to explain the results, analysis and project decisions, a maximum of 20 pages must be provided (including the cover page and annexes).

   There is no template or a document structure. However, use this document to tell the story about your project, explain your organization/process, decisions, results (performance analysis), and conclusions.

   Please do not add unnecessary theoretical concepts, we perfectly know what they are. You should be focused on your results.

II. **Source-code:** the code used in the project in Python. If needed, explain specific things about your code.

III. **Presentation:** the main conclusions. (You can deliver in defense day, it will be used to guide your presentation)

5. Delivery comments

The source code and the report must be contained in one .zip folder with Name and students numbers separated by _ (underscore)

e.g.: **Joao20170101_Maria20170304_Pedro20170909.zip**

The delivery must be performed through the MOODLE platform. A link will be shared with you.

Remark: A project containing errors will be penalized; it will be discounted of 2 values of 20.

6. Defense

During the defense, you and your group members will be asked to explain specific parts of your code and aspects regarding the theory (Oral exam). Each group will have 30 minutes. Where the first 15 minutes the group present the results and after 15m for the discussion (directed questions); the professors will ask the group's members questions.

Each student will be evaluated separately, according to his/her involvement in the project and quality of the oral defense (20% of the final grade - individual evaluation).

Oral Presentation: Presentation Document Layout

There is no template or a limit in the number of slides, **but remember that you have only 15 minutes to present your results and the selected solutions**. It is an optimization problem, so present the optima solution and describe the process, highlighting the main achievements and conclusions.

Please notice that:

- there is absolutely no need for *agenda* or *introduction* to the project - we perfectly know what the project is all about and what is the *agenda*;
- take care of the appearance of presentation document: do not mix too many things in the same slide, consider minimalism, avoid visual garbage;
- the sections presented by each student will be chosen by the teacher, at the beginning of the presentation.

7. Project Grade Rules

Evaluation Criteria

30%	**10%**	**10%**	**50%**
Report organization, clarity, and conclusions	Selected Solutions Performance	Oral Questions	Code and added techniques *Innovations can add value to the final grade.*

* Each group member will be assessed individually. Therefore, there are group grades, only individual grades.

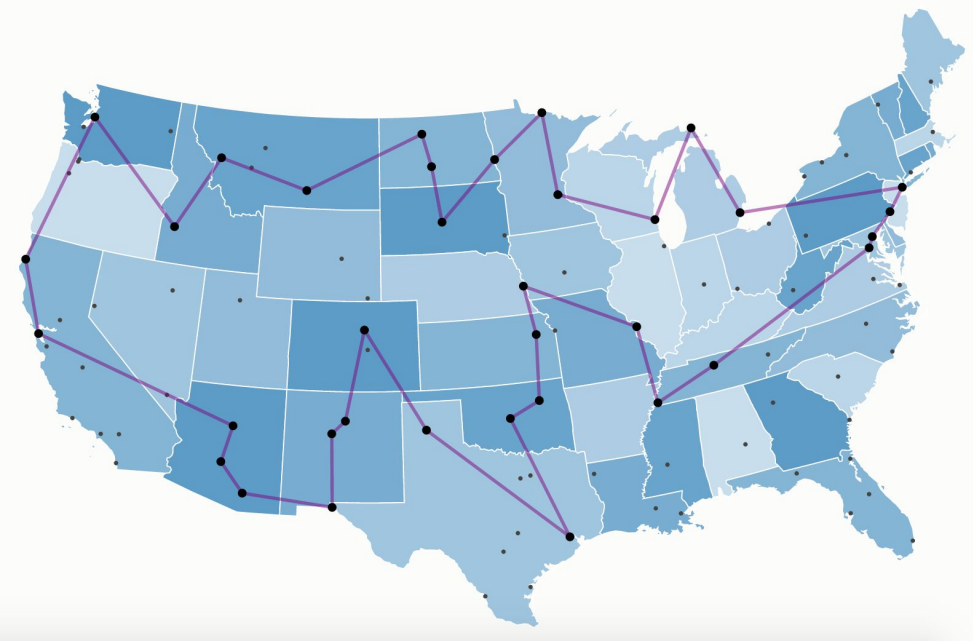
8. Ethics and conduct rules

The groups can share information, ideas, and can study together. However, groups cannot exchange source-code between them, groups with the same or similar code will be penalized.

You can (should) use on-line examples as a reference. Since they are common optimization problems this research can help you to know different possible approaches for the problem. However, you will need to develop your own version in accordance with the definitions contained in this document. Remember, you must extend the ProblemTemplate and the new approaches must follow the approaches methods signature (this the way that the GA current implementation is prepared to work with many possible approaches.

1. Travel Salesman Problem (TSP)

Finding an optimal route!!



1.1 Project's Task

Your main goal for this project is to minimize (in the best way you can) the distance traveled by a hypothetical salesperson. First, you need to develop a crossover and mutation to deal with TSP encoding rules. After this, you must:

- explore the new solution; and
- explore the influence of different parameter sets for this problem class.

Remark: a dataset will be provided to enable you to develop and test your algorithm extension.
It must be prepared to accept other datasets (decision variables)

1.2 About the problem

"Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?"

The Traveling Salesman Problem is an academic problem that aims to find the shortest route between a set of points and locations that must be visited. The points are the cities a salesperson must visit. The salesperson's goal is to keep the distance traveled as low as possible. Therefore, it is a MINIMIZATION problem. Keep in mind that all locations must be visited — starting and finishing at the same point.

The decision variable is a matrix containing all cities distances from one city to others:

```
data = [
      [0, 2451, 713, 1018, 1631, 1374, 2408, 213, 2571, 875, 1420, 2145, 1972],
      [2451, 0, 1745, 1524, 831, 1240, 959, 2596, 403, 1589, 1374, 357, 579],
      [713, 1745, 0, 355, 920, 803, 1737, 851, 1858, 262, 940, 1453, 1260],
      [1018, 1524, 355, 0, 700, 862, 1395, 1123, 1584, 466, 1056, 1280, 987],
      [1631, 831, 920, 700, 0, 663, 1021, 1769, 949, 796, 879, 586, 371],
      [1374, 1240, 803, 862, 663, 0, 1681, 1551, 1765, 547, 225, 887, 999],
      [2408, 959, 1737, 1395, 1021, 1681, 0, 2493, 678, 1724, 1891, 1114, 701],
      [213, 2596, 851, 1123, 1769, 1551, 2493, 0, 2699, 1038, 1605, 2300, 2099],
      [2571, 403, 1858, 1584, 949, 1765, 678, 2699, 0, 1744, 1645, 653, 600],
      [875, 1589, 262, 466, 796, 547, 1724, 1038, 1744, 0, 679, 1272, 1162],
      [1420, 1374, 940, 1056, 879, 225, 1891, 1605, 1645, 679, 0, 1017, 1200],
      [2145, 357, 1453, 1280, 586, 887, 1114, 2300, 653, 1272, 1017, 0, 504],
      [1972, 579, 1260, 987, 371, 999, 701, 2099, 600, 1162, 1200, 504, 0],
  ]
```

Remember, for this problem you must implement:

- Problem Encoding;
- Build Solution Function;
- Fitness Function;
- You do not need an admissibility function, so it always returns true, because the GA always ask if it is admissible.

Since you will need to create initialization based in Hill Climbing and Simulated Annealing, you also will need a neighborhood function for this problem.

1.3 Search Rules

To establish a common ground between groups (e.g. avoiding hardware performance to 'tip the balance'), there will be two restrictions on search parameters:

- Maximum number of iterations (Generations for G.A.) — 2000*
- Maximum number of population size (specific to G.A.) — 20*

2. Portfolio Investment Problem (TSP)

Finding an optimal investment portfolio!

The main objective of this project is to explore the Optimization concepts covered during the course. We'll do this through the application of meta-heuristic methods to approximate an optimal Portfolio Investment solution.

2.1 Project's Task

Your main goal for this project is to outperform (in the best way you can) the baseline version of the Genetic Algorithm that was provided.

What's expected from groups:

- Explore the baseline implementation.
- Explore the influence of different parameter sets for this problem class.
- Implement and track the effect of other algorithmic features.

Remark: a dataset will be provided to enable you to develop and test your algorithm extension.
It must be prepared to accept other datasets (decision variables)

Remember, for this problem you must implement:

- Problem Encoding;
- Build Solution Function;
- Fitness Function;
- Admissibility Function, so it always returns true, because the GA always ask if it is admissible.
- The constraints are defined below.

Since you will need to create initialization based in Hill Climbing and Simulated Annealing, you also will need a neighborhood function for this problem.

2.2. About the problem

In this scenario, we'll use real data from stock options of all companies comprised of the S&P 500's index.

symbol	name	price	exp_return_3m	std
A	Agilent Technologies	78.52	10.4	1.3931007143457097
AAL	American Airlines Gp	28.23	9.29	1.562678279362785
AAP	Advance Auto Parts Inc	161.74	19.47	0.5010106471159823
AAPL	Apple Inc	263.19	25.11	2.1073323711812764
ABBV	Abbvie Inc	87.2	31.31	1.557373762523036
ABC	Amerisourcebergen Corp	87.55	1.45	0.20206536500152056
ABMD	Abiomed Inc	185.25	-3.83	2.0745192039172307
ABT	Abbott Laboratories	83.89	-0.77	1.6615339052796005
ACN	Accenture Plc	197.71	1.81	0.4149087711222485
ADBE	Adobe Systems Inc	300.1	5.04	0.0696262018210489
ADI	Analog Devices	108.58	-1.55	2.010778203954326
ADM	Archer Daniels Midland	42.5	13.39	2.001876721674004

Note: For this initial dataset, the standard deviation was simulated. Therefore, the entire dataset will soon be updated.

The objective is to achieve a portfolio that maximizes the "Expected Return".
An example of how it is calculated:

> Let's say a portfolio has three investments with weights of 35% in asset A, 25% in asset B and 40% in asset C. The expected return of asset A is 6%, the expected return of asset B is 7%, and the expected return of asset C is 10%. Therefore, the expected return of the portfolio is 7.85% (35%*6% + 25%*7% + 40%*10%).

However, the problem is subject to **two constraints**:

- Investment budget — must be ≤ $1,000,000.00*
- Risk Tolerance (Sharpe Ratio) — must be ≥ 0.5*

**For now, these values serve as placeholders. They will be updated with the new dataset.*

2.3 Search Rules

To establish a common ground between groups (e.g. avoiding hardware performance to 'tip the balance'), there will be two restrictions on search parameters:

- Maximum number of iterations (Generations for G.A.) — 2000*
- Maximum number of population size (specific to G.A.) — 40*
-

**As the other parameters, these values will be updated with the new dataset.*

CHECKLIST

GROUP	DELIVERABLE	CHECK
TSP	Problem Design (description in the report)	
TSP	Solution Encoding / representation	
TSP	Build Solution	
TSP	Fitness Function	
TSP	Roulette Wheel selection for Minimization	
TSP	PMX Crossover	
TSP	Cycle Crossover	
TSP	Swap Mutation	
TSP	Save the best solution	
PIP	Problem Design (description in the report)	
PIP	Solution Encoding / representation	
PIP	Build Solution	
PIP	Fitness Function	
PIP	Admissibility Function	
PIP	Save the best solution	
MISC	Report	
MISC	Zip Source Code and best solutions	

TST - Travel Salesman Problem | PIP - Portfolio Investment Problem

If you develop additional approaches

GROUP	DELIVERABLE	CHECK