DEEC
DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA E DE COMPUTADORES
TÉCNICO LISBOA

Applied Computational Intelligence 2021/2022

Project 2 – EAs Single and Multi-Objective Optimization

(Week 7)

This project aims at applying Evolutionary Computation methods to solve the **Traveling Salesman Problem** (TSP) problem. In section 1, the problem is presented. In section 2, the details on the single objective problem to be implemented and the expected tests are described. In section 3, the multi-objective problem to be implemented, as well as the desired output figures are described. Finally, in section 4, the details on the submission (code and report) are given.

1. Problem Description

The Traveling Salesman Problem (TSP) is a classical combinatoric problem where the traveling salesman should visit all the n cities, **only once**, minimizing the traveling cost. The variant to this problem that should be implemented in this project is described in the following sections.

1.1. Data Set

In this problem, the data to be considered will be read from csv files. The data is organized as a symmetric matrix where lines and columns represent cities numbered from 0 to n-1 (where n has the maximum value of 100). The data is divided in 5 files: (1) the file CityDistCar.csv consists of a city-to-city road distances matrix (in terms of the required time to go from one city to any other city, by car); (2) the file CityDistPlane.csv consists of a city-to-city flight distances matrix (in terms of required time in minutes to go from one city to any other city by plane); (3) the file CityCostCar.csv consists of a matrix with the costs to go from one city to any other city by car; (4) the file CityCostPlane.csv consists of a matrix with the costs to go from one city to any other city by plane; (5) the file CitiesXY includes the cities XY coordinates. Please check examples of these files from table 1 to table 5.

Table 1: CityDistCar.csv sample

Distances of Cities by Car (min)	0	1	2	3	4	5
0	0	167	648	82	178	399
1	167	0	794	171	313	516
2	648	794	0	623	651	739
3	82	171	623	0	253	474
4	178	313	651	253	0	221
5	399	516	739	474	221	0

Table 2: CityDistPlane.csv sample

Distances of Cities by Flight (in min)	0	1	2	3	4	5
0	0	242	362	221	245	300
1	242	0	399	243	278	329
2	362	399	0	356	363	385
3	221	243	356	0	263	319
4	245	278	363	263	0	255
5	300	329	385	319	255	0

Table 3: CityCostCar.csv sample

Cost of Cities by Car (€)	0	1	2	3	4	5
0	0	33	130	16	36	80
1	33	0	159	34	63	103
2	130	159	0	125	130	148
3	16	34	125	0	51	95
4	36	63	130	51	0	44
5	80	103	148	95	44	0

Table 4: CityCostPlane.csv sample

Cost of Cities by Flight (€)	0	1	2	3	4	5
0	0	132	520	64	144	320
1	132	0	636	136	252	412
2	520	636	0	500	520	592
3	64	136	500	0	204	380
4	144	252	520	204	0	176
5	320	412	592	380	176	0

Table 5: CitiesXY.csv sample

City	x	y
0	822	509
1	983	464
2	394	996
3	859	582
4	675	409
5	506	266

1.2. Single-Objective Optimization Problem

The single objective optimization problem applied to the TSP problem will be divided in 4 different minimization problems, one for each data set described in section 1.1. In any of the 4 cases the objective is to minimize, in the first two, the required time and in the last two the overall trip cost.

1.3. Multi-Objective Optimization Problem

The multi-objective optimization problem applied to the TSP problem will consider the four data sets described in section 1.1 simultaneously, i.e., when generating a candidate solution, it should include the sequence of cities to be visited, as well as, the mean of transport to be used (car or flight).

In this case the goal is to generate an optimal Pareto front for the following two-objective problem: (1) minimize the time; and (2) minimize the cost required two visit the n cities.

The multi-objective problem should be subjected to the following **hard constraint**: the plane cannot exceed 3 travels in a row, i.e., flying for 4 or more consecutive cities is not acceptable in any solution.

2. Implementing the Single-Objective Optimization Problem

Implement the 4 single-objective optimization problems described in 1.2.

2.1. Problem Formulation and EA Set Up

First, start by defining how to represent the candidate solutions and identify which would be the evolutionary operators to consider. Afterwards, select an evolutionary approach and use either functions from an existing library or implement the EA from scratch (both solutions have their pros and cons … multi-objective problem 😉)

2.2. Solving the Optimization Problem

Solve the problem considering for the following cases: 20, 30 and 50 cities (consider always the cities starting from 0 on the csv file, i.e., from 0 to 19, from 0 to 29 and from 0 to 49).

Consider a maximum number of evaluations 10000, e.g., for a population of 40 elements and generating 40 offsprings limit the number of generations to 250. Remember that you are free to choose the EA approach. Explore and adjust the parameters of the chosen EA approach for the 3 case studies (20, 30 and 50 cities) of the 4 single-objective optimization problems.

2.2.1. Results

Complete the following table based on the execution of 30 runs (with different random seed) of each case.

#Cities	CityDistCar		CityDistPlane		CityCostCar		CityCostPlane	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD
20								
30								
50								

Generate the convergence curve (horizontal axis - #Generations; vertical axis Dist/Time or Cost depending on the case) for the best run of each of the 12 case studies.

2.3. Using Heuristics

Consider the use of the following heuristic to generate one candidate solution to be included in the first population. Assume the cities have XY coordinates in a range between 0 and 1000, as shown in figure. Generate an heuristic solution by splitting the horizontal axis in two and start traveling from the city on the left from lower to higher values on the vertical axis, than move to the right and travel from high to low values on the vertical axis. This would lead you to the solution presented in fig. 1.

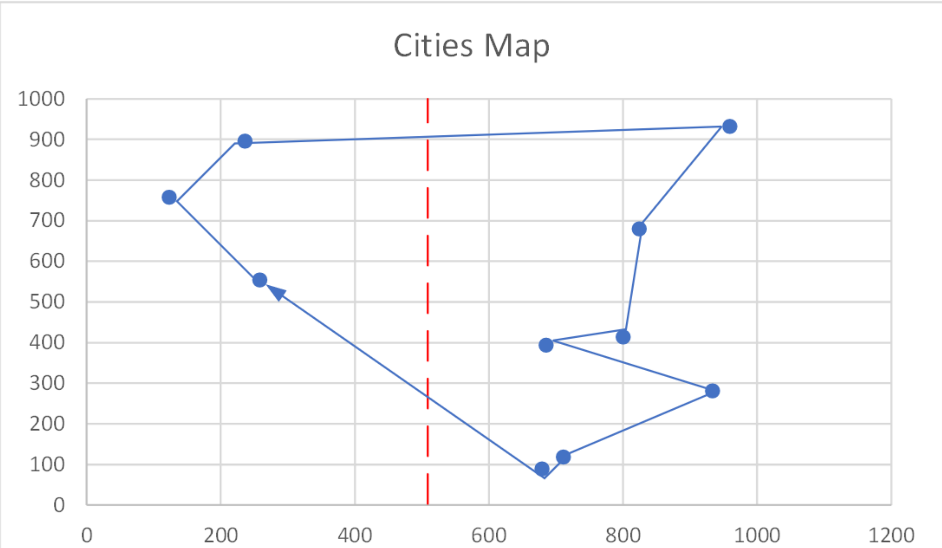

Fig. 1: Heuristic solution to the TSP

2.3.1. Results

Complete the following table based on the execution of 30 runs (with different random seed) of each case.

#Cities	CityDistCar		CityDistPlane		CityCostCar		CityCostPlane	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD
20								
30								
50								

Generate the convergence curve (horizontal axis - #Generations; vertical axis Dist/Time or Cost depending on the case) for the best run of each of the 12 case studies.

3. Implementing the Multi-Objective Optimization Problem

Implement the multi-objective optimization problems described in 1.3.

3.1. Problem Formulation and EA Set Up

First, start by defining how to represent the candidate solutions and identify which would be the evolutionary operators to consider. Remember that, in this case, each candidate solution should have information not only on the sequence of cities but also on the mean of transport used from city to city. Moreover, the candidate evaluation is now a pair of values, one corresponding to the time to complete the visit and the other overall cost. In order to obtain these values, you should use the four matrixes from the csv files.

3.2. Solving the Optimization Problem

Solve the problem considering for the following cases: 20, 30 and 50 cities (consider always the cities starting from 0 on the csv file, i.e., from 0 to 19, from 0 to 29 and from 0 to 49).
Consider a maximum number of evaluations 10000, e.g., for a population of 40 elements and generating 40 offsprings limit the number of generations to 250. Remember that you are free to choose the EA approach. Explore and adjust the parameters of the chosen EA approach for the 3 case studies.

3.2.1. Results

Generate the Pareto curve (horizontal axis - Cost; vertical axis Dist/Time) for each of the 3 case studies.

Complete the following table for the Pareto front obtained for each case.

#Cities	Min Cost		Min Dist	
	Dist (Time)	Cost	Dist (Time)	Cost
20				
30				
50				

Generate the hypervolume evolution curve for each of the 3 case studies.

4. Work Submission

The work should be **submitted on the course website by November 12th, 23:59**. The submission must include a zip package with: (1) the code (commented) to be tested in lab environment, also, include a README file describing the use and the sequence of results the user will see when testing your work; (2) the report should include (2.1) a brief graphical description of the chosen representation for each problem (single and multiple objective), (2.2) the tables and graphs from sections 2.2.1, 2.3.1 and 3.2.1, and (2.3) a final section with concluding remarks. The report should not exceed 4 pages.

5. Evaluation

The work evaluation is based on the work submitted and on the oral discussion.
The **single objective part** corresponds to 3/4 of the work classification (15 over 20).
The **multiple-objective part** corresponds to 1/4 of the work classification (5 over 20).