DEEC
DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA E DE COMPUTADORES
TÉCNICO LISBOA

Applied Computational Intelligence 2022/2023

Project 2 – EAs for 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 solved and the expected tests are described. In section 3, the multi-objective problem to be solved, 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 a traveling salesman should visit all the specified 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. The Variant to the TSP

First, assume you are the distributor of a certain product, e.g., drinks, cloths, etc. … anything portable. Consider you have the products stored in a warehouse and you, also, have a truck capable of transporting n of these products at a time. The objective is to distribute the products to customers using the truck in the most efficient way possible.

1.2. Data Set

In this problem, the data to be considered will be read from csv files.
The data is divided in 3 files:

- the file CustDist_WHCentral.csv, see table 1, consists of a customer-to-customer road distances matrix;
  - The data is organized as a symmetric matrix where lines and columns represent the warehouse and customers. The warehouse is identified by 0 and the customers are identified from 1 to 50.
- the file CustXY_WHCentral.csv, see table 2, includes the customer XY coordinates;
  - The data is organized as a 3-column matrix, where the first column identifies the warehouse and customers, and the remaining two columns give the X and Y coordinates for the location in a scenario of size 100x100.
- the file CustOrd.csv, see table 3, includes the customers' orders, i.e., the number of ordered products.
  - The data is organized as a 2-column matrix, where the first column identifies the warehouse and customer, and second column indicates the number of products ordered by each customer.

Table 1: CustDist_WHCentral.csv (sample)

Distances between Customers and Warehouse	0	1	2	3	4	5	6	7	8	9	10
0	0	43	44	37	39	53	24	45	26	36	15
1	43	0	31	59	47	92	67	67	65	70	30
2	44	31	0	37	68	97	62	84	70	53	40
3	37	59	37	0	75	79	38	81	55	18	45
4	39	47	68	75	0	58	55	23	40	75	31
5	53	92	97	79	58	0	41	38	28	66	63
6	24	67	62	38	55	41	0	52	21	26	39
7	45	67	84	81	23	38	52	0	31	76	45
8	26	65	70	55	40	28	21	31	0	46	36
9	36	70	53	18	75	66	26	76	46	0	49
10	15	30	40	45	31	63	39	45	36	49	0

Table 2: CustXY_WHCentral.csv (sample)

Customer XY	X	Y
0	50	50
1	92	40
2	76	14
3	39	15
4	72	82
5	15	90
6	26	51
7	53	95
8	34	70
9	24	25
10	65	52

Table 3: CustOrd.csv (sample)

Customer	Orders
0	0
1	40
2	90
3	90
4	80
5	40
6	10
7	20
8	60
9	20
10	60

Fig. 1, illustrates the maps with the location of the first 10 and 50 customers, the red square indicates the warehouse location. The map with the 10 customers also includes the number of orders for each customer.

Fig 1: Customer and Warehouse location for #Customers 10 and 50.

Note that the above files should be used when considering the warehouse in the middle of the area (50, 50), as illustrated in fig. 1. When considering the warehouse in the corner, coordinates (0,0), the files to be used are: CustDist_WH**Corner**.csv, CustXY_WH**Corner**.csv and CustOrd.csv (this last file is similar in both cases)

1.3. Single-Objective Optimization Problem

The single objective optimization applied to the TSP problem will consist of minimizing the total distance the truck must travel considering different set of parameters, e.g., different number of customers and the different warehouse location. Notice that if the truck's capacity to transport products is less than the total number of ordered products, the truck must return to the warehouse to load new products and initiate a new run to distribute the products to customers.

1.4. Multi-Objective Optimization Problem

The multi-objective optimization problem applied to the TSP problem will add a new objective to the one considered for the single-objective problem described above. The new objective will be a measure of the cost of transporting the products, i.e., the cost of transporting the products will be higher when the truck is transporting a larger number of products. For example, if the truck is transporting 10 products between 2 points that are 5 km apart, the cost will be 50, but if the truck is transporting 5 products the cost will be 25. In conclusion, to minimize the cost the truck should first deliver the product to customers ordering the large number of products but that is not necessarily the shortest path, which means that the result should be a Pareto front presenting the non-dominated solutions for these 2 objectives.

2. Implementing the Single-Objective Optimization Problem

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 the following cases: 10, 30 and 50 customers (consider always the customers starting from 1 on the csv file, i.e., from 1 to 10, from 1 to 30 and from 1 to 50). The label 0 is used for the warehouse.

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 (10, 30 and 50 customers) and for the four variations on the single-objective optimization problems (one considering the warehouse on the central (X=50, Y=50) location and other considering the warehouse on the corner (X=0, Y=0), and for each warehouse location consider two cases, customer order read from the CustOrd.csv and number of product orders equal to 50 for all customers). Finally, consider the truck is limited to transport a maximum of 1000 products, which means that, at least, after delivering 1000 products the truck must return to the warehouse and load again.

2.2.1. Results

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

#Customers	WHCentral_OrdFile		WHCentral_Ord50		WHCorner_OrdFile		WHCorner_Ord50	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD
10								
30								
50								

Generate the convergence curve (horizontal axis - #Generations; vertical axis the total distance for the best run for each of the 12 case studies. Generate only three graphs one for each number of customers and superimposed the curves from the 4 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 customers have XY coordinates in a range between 0 and 100, as shown in fig. 2. Generate a heuristic solution by splitting the horizontal axis in two and start traveling from the warehouse to the customer 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. 2. For such a simple case it looks like that generating a solution by inspection and using a good heuristic would solve the problem but imagine the problem with the 50 customers or even more, that won't be feasible without a searching mechanism.

Fig. 2: Heuristic solution to the TSP

2.3.1. Results

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

	WHCentral_OrdFile		WHCentral_Ord50		WHCorner_OrdFile		WHCorner_Ord50	
#Customers	Mean	STD	Mean	STD	Mean	STD	Mean	STD
10								
30								
50								

Generate the convergence curve (horizontal axis - #Generations; vertical axis the total distance for the best run for each of the 12 case studies.

3. Implementing the Multi-Objective Optimization Problem

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. Moreover, the candidate evaluation is now a pair of values, one the distance to complete the product delivery and other the overall cost. In order to obtain the cost for each solution follow the description given in 1.4.

3.2. Solving the Optimization Problem

Solve the problem considering for the following cases: 10, 30 and 50 customers (consider always that the deliveries start and end at the warehouse and the **warehouse is in the center** (X=50, Y=50)). 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. Finally, consider just the non-uniform list of orders given in the file CustOrd.csv and that the truck is limited to transport a maximum of 1000 products, which means that, at least, after delivering 1000 products the truck must return to the warehouse and load again.

3.2.1. Results

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

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

#Customers	Min Cost		Min Dist	
	Dist	Cost	Dist	Cost
10				
30				
50				

Generate the hypervolume evolution curve for each of the 3 case studies (without the use of heuristic for the initial population).

4. Work Submission

The work should be **submitted on the course website by November 4th, 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).