

## Homework #2

### Solving the p-Median Problem using Simulate Annealing and VNS

In this homework, you are going to solve the p-median problem for three data sets. They are called eil51, eil76, and eil101, and consist of 51, 76, and 101 customer locations, respectively. Each data set includes the x-coordinates, y-coordinates, and demand of customers. Customer locations are also potential facility location for opening facilities. For each data set, you are going to locate 4, 6, and 8 facilities. This makes a total of 9 instances (3 data sets, 3 different  $p$  values). The distances between potential facility locations and customer locations are measure via **Euclidean distance** (not Squared Euclidean distance). This is important because the optimal objective values I will provide are also based on the Euclidean distance. Please round all the distance values to two digits after the decimal point. This is also true for the objective values.

For the Simulated Annealing, you will use the **1-Swap** move which removes an existing facility and adds a new facility. It is your responsibility to assign proper values to the parameters of the algorithm. For the Variable Neighborhood Search, you will use **1-Swap**, **2-Swap**, and **3-Swap** moves in the shaking step. You can use any algorithm for the local search part. You will run both algorithm **10 times** starting with **10 different initial solutions**, but please use the same initial solution at the same run for both algorithms. In other words, run 1 for both SA and VNS should start from the same initial solution.

I would like to remind you the following points which you should consider when you submit your homework. It will consist of two parts: your **code** and **report**. First, your code must be clear and you should define the following using comment lines in the code: **variables names** and their **purpose**, **function names** and their **purpose**. For example, you should write “X is the assignment variable”, “CompObj calculates the objective value”, etc. Or you can use a function name that is self-explanatory e.g., ApplyMove.

In the report part, you have to mention which solution representation and neighborhood structure you used as well as other pertinent and tiny details worth pointing out. You can use the following table for the output of your solutions. A separate table for each of 9 instances:

Instance: eil51-4				
Run No.	Solution (opened facilities)	Obj. Value	No. of iterations	CPU Time (s)
1	(4,17,23,34)	2300.12	15	20
2	(6,12,21,45)	4600.45	10	25
3	.	.	.	
4	.	.	.	
5	(10,20,30,40)	2100.78	12	30
6	.	.	.	
7	.	.	.	
8	.	.	.	
9	.	.	.	
10	.	.	.	
Best Run				
5	(10,20,30,40)	2100.78	Write here the average no. of iterations	Write here the total CPU Time