

Assignment 1

Metaheuristics

Contents

1	Exercise 1.	2
1.1	How does this algorithm behave as we increase the complexity of the problem (number of cities in the TSP)?	2
1.2	Do you always get the best solution? Why? What does it depend on?	3
1.3	Modify the code to start the search again from another initial solution (Iterated local search). Have you managed to improve? Why?	3
2	Exercise 2.	4
2.1	How does this algorithm behave as we increase the problem (number of cities in the TSP)?	4
2.2	Do you always get the best solution? Why? What does it depend on?	4
2.3	Analyze how the behavior of the algorithm varies as we change the stop criteria and the initial temperature.	4
2.4	Modify the code to use different cooling functions	4
2.4.1	Logarithmic.	4
2.4.2	Geometric.	4
2.5	How do these functions affect the final results? Why? Help yourself by representing the values of these functions. Look for new features and compare them to the old ones.	4
2.6	How would you improve the algorithm? For example, reheating so often. Modify the code with this and any other improvement you guess is appropriate. Analyze the results.	4

List of Figures

1	Time complexity of Hill Climbing	2
2	Length of final path TSP problem	2
3	Computation time changing the iterations	3

1 Exercise 1.

1.1 How does this algorithm behave as we increase the complexity of the problem (number of cities in the TSP)?

For this question I have evaluated the algorithm measuring the time and the length of the final path, to do this I have done the mean of five iterations for each city using eleven cities between 5 and 150.

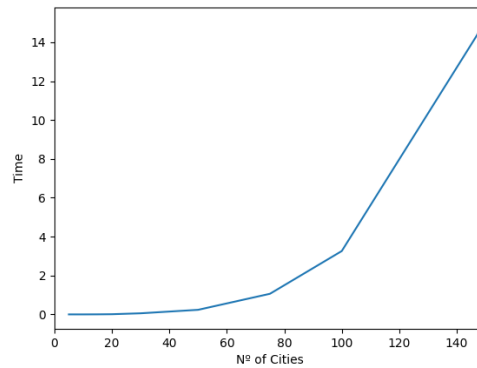


Figure 1: Time complexity of Hill Climbing

From this figure we can say that the complexity of the algorithm increases exponentially as the number of cities increases.

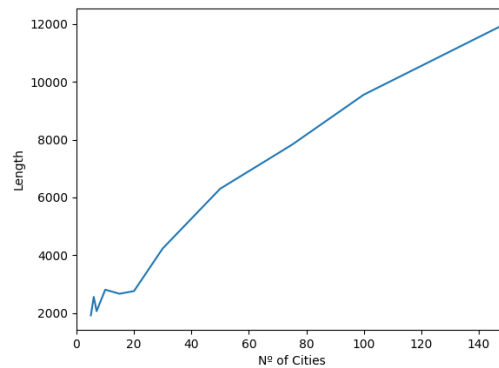


Figure 2: Length of final path TSP problem

From this figure we can say that obviously the more cities there are the longer the path will be, however it seems that the heuristic are finding good solutions because the slope does not increase too much.

1.2 Do you always get the best solution? Why? What does it depend on?

We do not obtain always the best solution, cause of the first randomly generated solution. Sometimes the algorithm get trapped on a local maximum and it never reaches the global maximum.

1.3 Modify the code to start the search again from another initial solution (Iterated local search). Have you managed to improve? Why?

It obtains better results but needs more computation time. On my case I do it for 1,10 and 100 iterations.

Doing that It does not matter if the first random solution is not good, because along the execution of the algorithm it can reach a better neighbour. For each iteration it compares if the solution given is better than the best solution ever and if it is better it store the value of it.

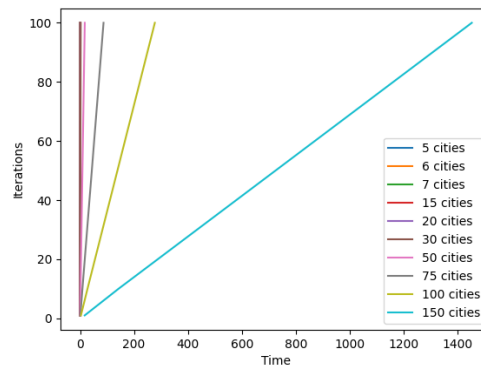


Figure 3: Computation time changing the iterations

2 Exercise 2.

- 2.1 How does this algorithm behave as we increase the problem (number of cities in the TSP)?
- 2.2 Do you always get the best solution? Why? What does it depend on?
- 2.3 Analyze how the behavior of the algorithm varies as we change the stop criteria and the initial temperature.
- 2.4 Modify the code to use different cooling functions
 - 2.4.1 Logarithmic.
 - 2.4.2 Geometric.
- 2.5 How do these functions affect the final results? Why? Help yourself by representing the values of these functions. Look for new features and compare them to the old ones.
- 2.6 How would you improve the algorithm? For example, reheating so often. Modify the code with this and any other improvement you guess is appropriate. Analyze the results.