Introduction to Intelligent Systems
Lab 3

Team 15:
Sara Bardají Serra and Albert Sallés Torruella

September 2019

Assignment

The Travelling Salesman Problem (TSP) is a well-known optimization problem. That is because of its complexity and because it is considered an NP-hard problem. The problem consists on, given N cities, a salesman must travel trough all of them. The order in which he does doesn't matter, as long as he visits every one of them and ends in the same city where he started. The idea is to keep the journey as short as possible, so the path must be minimum.

In this practical, we're going to approach the problem using the Metropolis algorithm.

Modify the code in such a way that your program:

- **Runs the Metropolis version of the optimization for a number of N cities, but at constant temperature parameter T.**

  In order to set parameter T as a constant, we just comment the comand where this parameter is changed:

```
    %   lower temperature by 0.1 percent
    %   temp = temp*0.999;
```

Code for the first objective

- **Performs (at least)** $100 \times N$ **single steps, i.e. set the parameter.**

  If the maxsteps parameters is set to 100 when the value given is lower than 100, the program will always perform at least $100 \times N$ single steps.

```
    %  performs at least 100 x N single steps
       if (maxsteps < 100)
           maxsteps = 100;
       end
```

Code for the second objective

- Calculates the mean value $< l >$ and the variance $var(l) =< l^2 > - < l >^2$ where averages are computed over the last 50 measured values maxsteps=100 in the code.

  We take the last 50 measured values and compute the mean and variance.

```
    %  compute the mean and variance of the last 50 measured values
        mean_last50 = mean(lt(end-50:end));
        var_last50 = var(lt(end-50:end));
```

Code for the third objective

- Outputs the results $< l >$ and $var(l)$.

  To output the results we create a Map as a return value of the function and we store the mean and variance we computed before.

```
        result = containers.Map;
        result('mean') = mean_last50;
        result('var') = var_last50;
```

Code for the forth objective

After modifying the code correctly, we have run our program where we took N = 100 cities with maxsteps = 500 and different temperatures: T = [1, 0.5, 0.25, 0.2, 0.1, 0.05, 0.03, 0.02, 0.01, 0.005, 0.001]. Then, for each temperature we have run 50 simulations and to compute the mean and variance of the lengths we have taken the mean of the 50 simulations, so that the result is as accurate as possible. So the graph we got as a result is as follows:

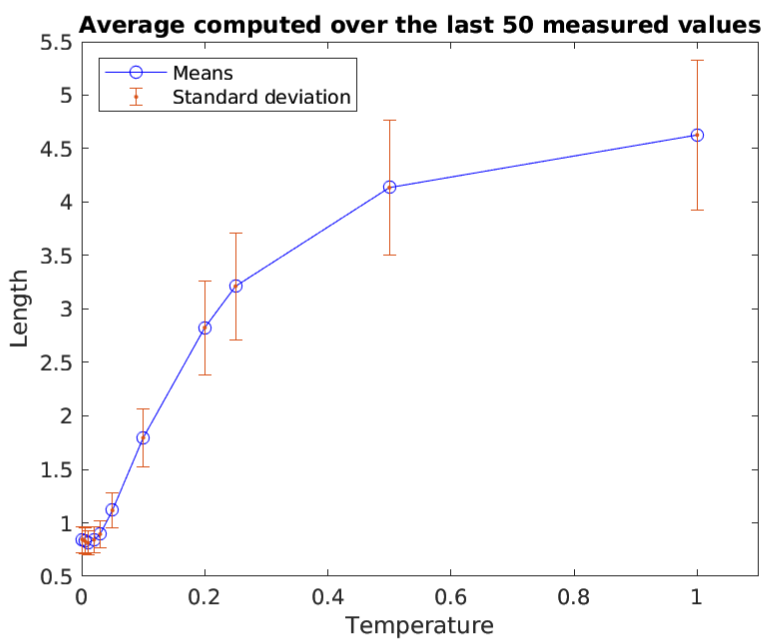

Figure 1: Plot of the simulation

Conclusion

We can clearly see that the smaller the temperature values are, the shorter length and variance we get. However, that is not entirely true, if we zoom into the smallest values for T, we can see that we have a minimum there:

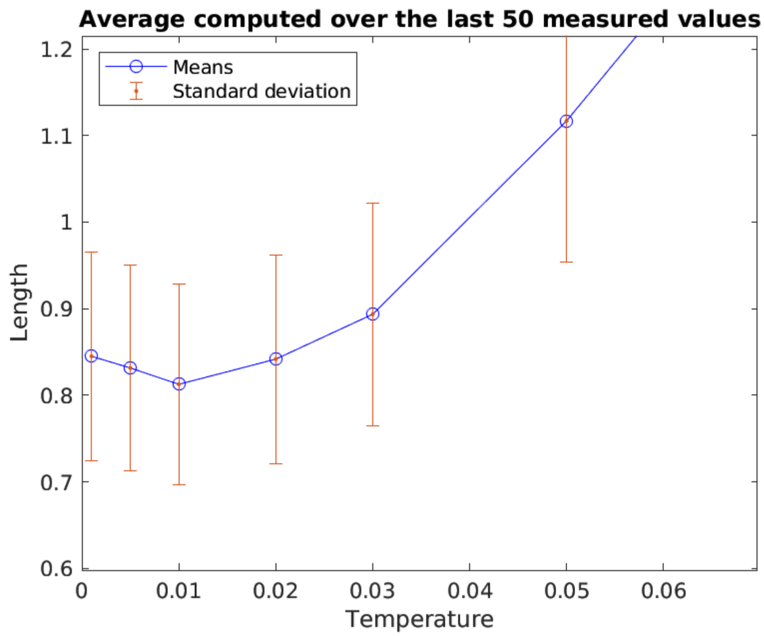

Figure 2: Zoom of the plot

Looking at the previous graph, we see that T = 0.1 is a minimum in this graph. This fact has a simple explanation. If the temperature is very close to 0, the chance that we accept a random change even though it is a higher length is practically 0, therefore we get stuck in a local minima. If the temperature is close to 1, then we are changing the path most of the times without improving our length, therefore we practically never find a minimum. Nevertheless, there is a value in between that minimizes both errors and, in our case, it is around 0.01, where we have our minimum value.

Moreover, we can also see that the standard deviation decreases as we approach to 0, and that is for the same reason as before, because in those cases we are practically always stuck in a local minimum.

Work division

During the realization of this assignment, we have been working and making decisions together, so we have divided the work equally. Also, the both of us have written and checked this report.