University of Amsterdam

Stochastic simulation, Assignment 3

TSP-Problem: Simulated Annealing

13-12-2021

Florian Tiggeloven, Lars Grim
11872802, 13511157
florian.tiggeloven@student.uva.nl,
lars.grim@student.uva.nl

Abstract

In this paper, the effect of changing temperature functions of simulated annealing are analysed using the TSP-problem. Three different functions were analysed: linear, quadratic and exponential. There was found that the exponential function converges the quickest to a low value of total distance. Besides this, the Markov chain length was increased for all functions. The exponential shape still returned the lowest value, but converged quicker for higher chain lengths. This is however not advised when not a lot of computational time is available.

1 Introduction

The travelings salesman problem (TSP) is one of the most famous examples of an NP-hard combinatorial optimization problem (Hoffman and Padberg, 2001). It can be used in daily life through routing logistics or even through DNA sequencing. In general, and in this study, TSP is represented as a list of cities in which the total distance between the consecutive cities must be minimized while other representations also exist. Each city has a fixed coordinate and an index within the list containing all cities. The question asked is to find the order of the n number of cities in the list for which the distance $d_{tot} = \sum_{i=1}^{n} ||\binom{x_{i+1}}{y_{i+1}} - \binom{x_i}{y_i}||_2$ is minimal. Note that city $\binom{x_{n+1}}{y_{n+1}}$ is the first city of the list again.

A possible way to find the order of the cities for which the total distance is minimal is, is to compute all $\binom{n}{p}$ combinations and to compute and compare all the total distances. But for lists with a large number of cities this will result in a high computational cost. Simulated Annealing (SA) is a global optimization technique which drastically reduces the computational cost (Bertsimas and Tsitsiklis, 1993). SA is often used when the search space is discrete, as is the case with TSP. The idea behind optimization is to only accept a new solution when it is closer to the optimum than the current solution. Other optimization techniques like the hillclimber, gradient descent or branch bround are perfect to use when there is just one optimum to be found while SA can also be used when several local optima exist.

The strength of SA is due to the correlation the algorithm has with a cooling/temperature function. The cooling function allows the algorithm to accept a new solution which is not closer to the optimum than the current solution. Therefore it allows the algorithm to extensively jump across the entire search space and possibly leaving local optima in hope of finding the global optimum. When the temperature is high (generally at the beginning of the iteration period) the chance of jumping to a more worse solution is higher than when the temperature is lower. Thus often the cooling function decreases the temperature over time to eventually allow the algorithm to search and stay in a smaller part of the search space which preferably contains the global op-

tima. In this paper three different cooling functions are implemented: a exponential function (T_{exp}, a quadratic function (T_{quad}) and a linear function (T_{lin}). Where T_0 represents the initial temperature, i represents the current iteration and α represents a constant depending on the distributions (**Gelatt**):

$$T_{exp}(i) = T_0 \cdot \alpha^i \tag{1}$$

$$T_{quad}(i) = \frac{T_0}{1 + \alpha \cdot i^2} \tag{2}$$

$$T_{lin}(i) = \frac{T_0}{1 + \alpha \cdot i} \tag{3}$$

For the exponential function (equation 1) it is desired that the slope starts moderate to strong negative, then becoming less negative and finally approaching 0. For the quadratic function (equation 2) it is desired that the slope starts flat, then becoming more negative and finally approaching 0. For the linear function (equation 3) it is desired that the slope starts strongly negative, then becoming less negative pretty fast and slowly approaching 0. As can be seen in equations 1-3, the behaviour of each function is very much dependent on α. For the exponential function a higher α, with $0 < \alpha < 1$, will result in slower converging function. For the quadratic function a higher α, with $\alpha > 0$, will result in a faster converging function. For the linear function a higher α, with $\alpha > 0$, will result in a faster converging function.

The probability to accept a more worse solution is defined in this paper as: $p(T) = e^{-\frac{|d_t - d_i|}{T_i}}$. Where e is the euler constant, d_t is the total distance of the possible new tour, d_i is the current total distance and T_i is the current temperature. Due to the way this probability is calculated, it can be seen that the likelihood for the algorithm to accept a more worse solution decreases over time since T decreases over time.

A useful way of viewing the TSP problem is through describing it as a Markov chain system. Many stochastic processes can be described through the usage of a Markov chain, like for instance the random walk-like behaviour of the stock market. In short, a Markov chain is a mathematical system that describes a probabilistic sequence of events that each have a probability of occurring that is based only on the value of its previous event. Therefore Markov chains are often called "memoryless", since they base their new decisions only on the single previous event. The earlier mentioned TSP problem faced this same structure. Routes between cities are combined and re-arranged probabilistically, but when a bad solution is found it will only be accepted based on a certain chance α. In this paper, the lengths of each of the cooling functions' Markov chains will be analysed. By observing each of the Markov chain lengths and the rate of decrease to a certain distance, conclusions can be made about the convergence per chain length of each method.

In this study SA, along with the three different cooling functions, is used to search for a list of cities in which the distance d_{tot} is minimum. The goal of this study is to discuss the influence of the cooling function on the convergence rate of d_{tot}.

2 Methodology

Simulations were executed with `Python 3.8` using the `Scipy`, `Matplotlib` and `Copy` packages.

2.1 Parameter Selection

As mentioned in the introduction, the parameters controlling the cooling functions are T_0, α and the current level of iteration. For all simulations T_0 was set to 18000. This was done to assure a high starting temperature so the algorithm is allowed to extensively jump through the search space at the beginning. Every 50 iterations parameter i was increased with 1. This was done to assure that the cooling function returned low temperatures around iteration 5000

To ensure all cooling functions are somewhat similar but still obtain their own function characteristics as described in the introduction, the following values for α were used: $\alpha_{lin} = 0.125$, $\alpha_{quad} = 0.005$ and $\alpha_{expo} = 0.95$. These values are derived with trial and error to eventually get the results as visualized in figure **??**. The cooling function is only visualized for the first 5000 number of iterations since after this the cooling is approaching a temperature of 0. This is desired since eventually the algorithm must stay within a certain interval of the search space to find the optimum.

In figure 1 the different cooling functions can be seen for i ranging from 0 to 5000. It can be seen that the characteristics described in the introduction are met for the parameters described in the previous paragraph. This is the representation of how the cooling functions should look like, The actual cooling functions used are described in the results section.

2.2 Cooling functions

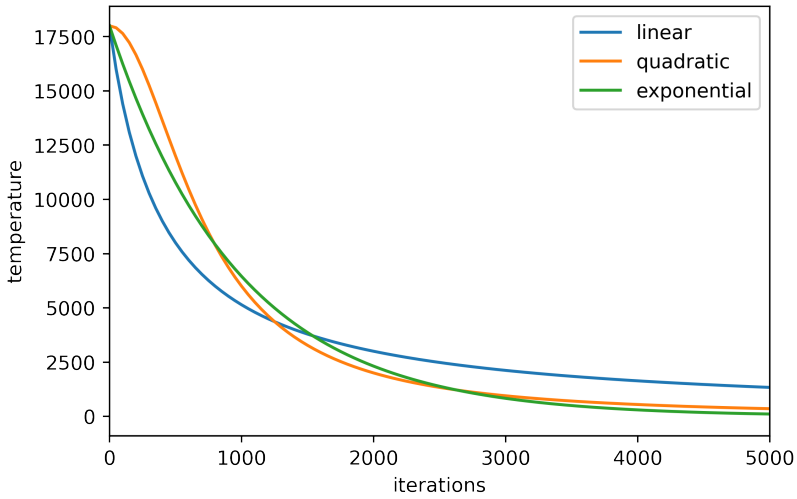

Figure 1: Visualisation of the linear, quadratic and exponential cooling functions for iterations in [0, 5000].

2.3 Pseudocode

Simulated Annealing is done through running through a list of cities, each being an object containing an x and a y value. Through this description, the distance between each city can be easy computed through pythagoras' theorem. The algorithm looks for the distance between each of the elements in the shuffled cities list, and accepts the distance directly if it is shorter than its previously found distance. If not, then it has a chance of being accepted based on the cooling distribution that is used. Options for this are lin(), exp() or quad(), referring to the linear, exponential and quadratic functions respectively.

3 Results

3.1 Cooling functions

In figure 2 the different cooling functions can be seen for i ranging from 0 to 5000. This figure slightly deviates from figure 1. It can be seen that the cooling functions are not as smooth as in figure 1.

Algorithm 2 Simulated Annealing TSP

```
procedure SimAnneal(cities)
    for i,city in enumerate(cities): do
        getdistance(city[i],city[i+1])          ▷ Each city an object with x and y
        Accept temporary distance if smaller or if
        random number is smaller than the distribution
        if c is within lin(),exp() or quad(): then
            distance= temporary distance
            markov += 1
            Update cooling function's iteration
        end if
        return totaldistance,markov
    end for
end procedure
```

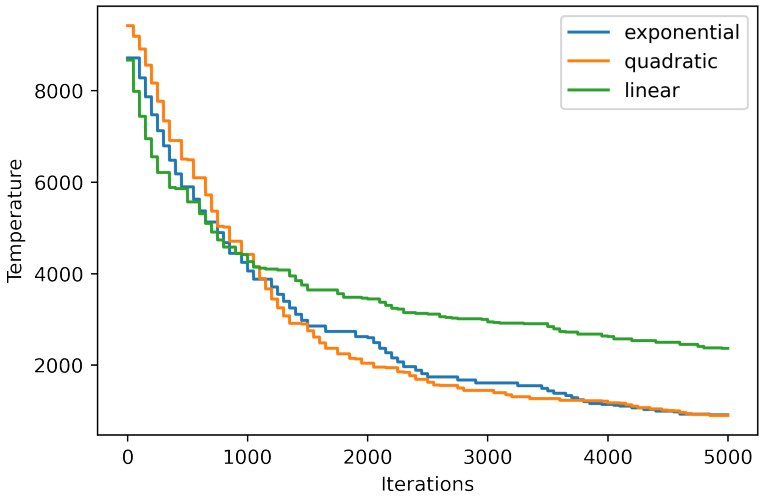

Figure 2: Visualisation of the actual linear, quadratic and exponential cooling functions for iterations in [0, 5000].

3.2 Convergence of d_{tot}

All different cooling functions are shown plotted over a length of 10000 iterations in figure 3. Exponential being shown in blue, linear in orange and quadratic in green. The standard deviation of each of the iterations is shown around each line in a lighter shade. Each of the functions beginning at a value of around 25000m for max distance. The exponential seems to converge to the lowest minimal distance of around 2000m, followed by the quadratic function at around 5000m and the linear function lastly remaining around 20000m.

3.3 Markov Chain Exponential

To visualize the behaviour of the total distance convergence, the functions were simulated once more with different Markov chain lengths. The exponential function's convergence is shown below in figure 4. For a Markov chain size of 1, the same result as earlier is seen with a low distance of around 5000m. The functions converges faster for higher values of Markov chain length, since a length of 3 results in a value of about 200m and a length of 5 going even lower than that to about 150m.

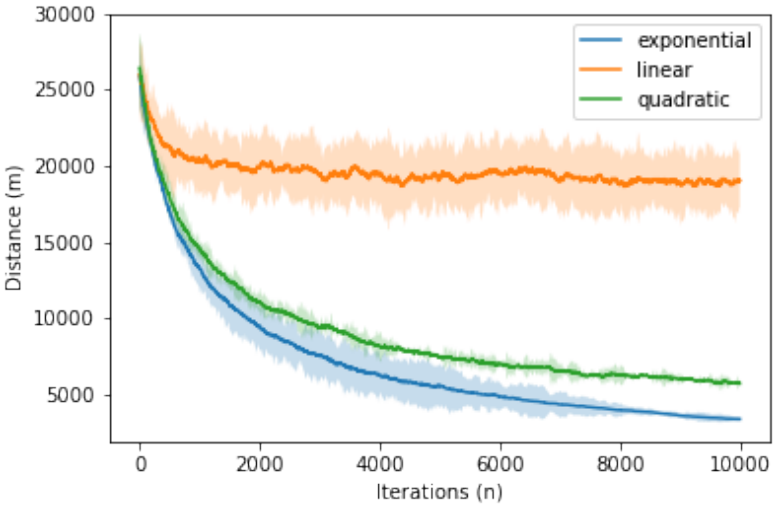

Figure 3: Visualisation of the linear, quadratic and exponential functions over their iteration length.

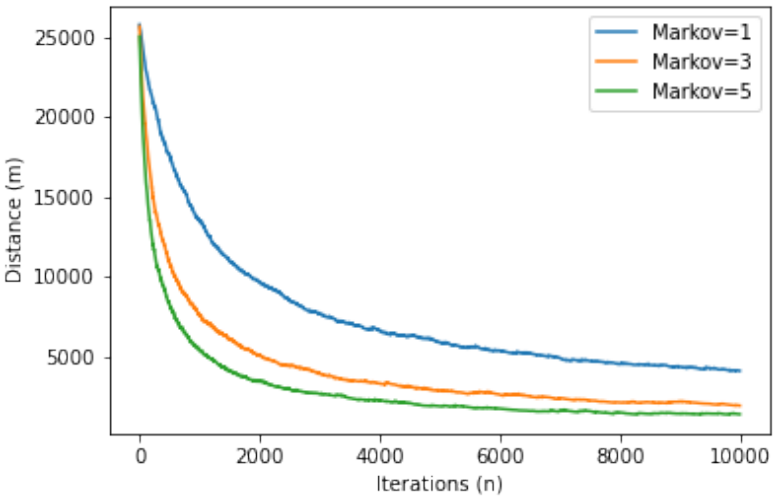

Figure 4: Visualisation of the exponential functions for Markov chain length.

3.4 Markov Chain Linear

The linear cooling function also shows similar behaviour as our earlier plots, being shown in 5. Each of the lines swinging around a certain value for most of the graph. For a Markov chain length of 1 we find a distance of about 20000m. If this size is increased to a length of 3 we find that the iterations converge to a total distance of about 14000m. Lastly, The Markov chain length of 5 also converges quickly to a value of 10000m.

3.5 Markov Chain Quadratic

The exponential cooling function has so far converged to the lowest degree of total distance, which is also apparent through its Markov chain lengths in figure 4. For a Markov chain length of 1, the graph once again converges to a value of 10000m. Increasing this chain length to 3 results in a total distance of about 1000m, followed by the chain length of 5 reaching to a distance

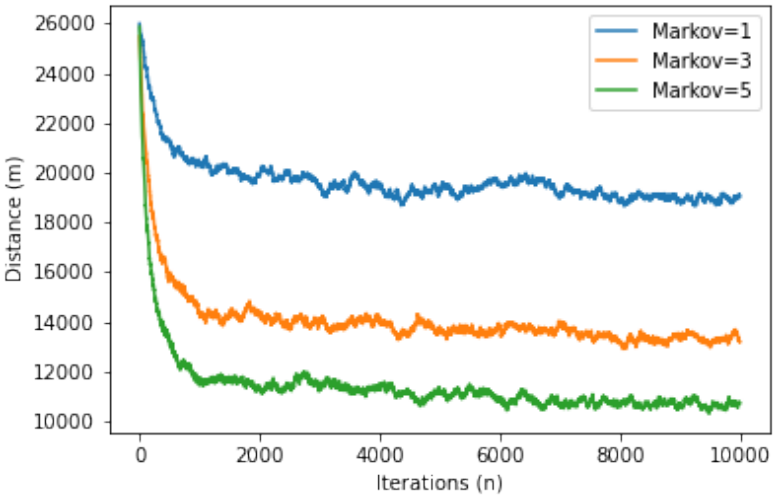

Figure 5: Visualisation of the linear functions for Markov chain length

of 800m.

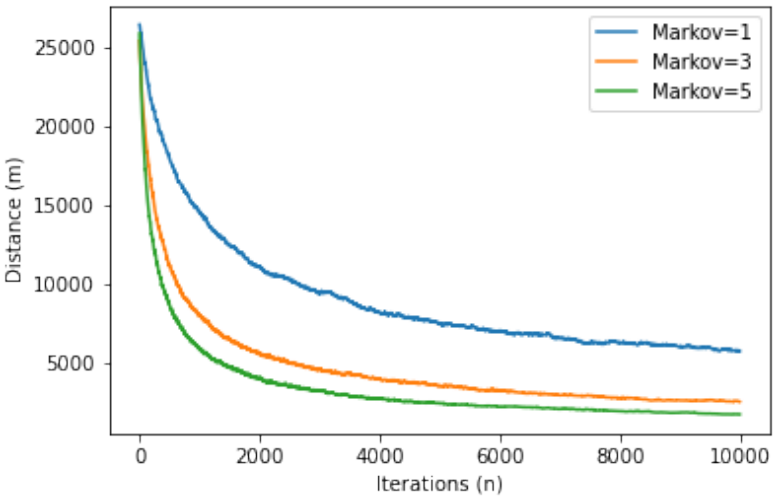

Figure 6: Visualisation of the quadratic functions for Markov chain length.

4 Discussion

Several interesting points can be concluded from the earlier shown results. It is clear that the linear cooling function yields the poorest result, with its distance getting stuck around a value of 20000m. This is likely due to the function finding a local minimum, and not being able to climb out of it. The cooling functions of the exponential and logarithmic don't seem to experience this problem, due to their different shape. One should always select carefully between the different cooling functions, since the behaviour of the functions is still very dependent on the values of T0 and α. Over all results, increasing the Markov chain length returns a lower total distance. For higher values of chain length the graphs also converge to these values significantly faster. It is therefore advised to adjust these chain lengths to higher values if the longer computational time

is available. For direct improvements on this study it is suggested to control parameter i as a floating parameter instead of an integer. In this study parameter i was increased with 1 every 50 iterations while increasing i with 0.02 every iteration results in a more fluent cooling function. It is believed that the effect of this minor mistake on the results is minimal and therefore the results are still valid to use for this study.

5 Conclusion

In conclusion, it is found that the exponential cooling function yields the best results over all other functions. Increasing the Markov chain length ensures that the function converges to a total distance over lower iterations. One must be wary however, since this costs significantly more computational time.

References

Hoffman, K. L., & Padberg, M. (2001). Traveling salesman problem (tsp)traveling salesman problem. In S. I. Gass & C. M. Harris (Eds.), *Encyclopedia of operations research and management science* (pp. 849–853). Springer US. https://doi.org/10.1007/1-4020-0611-X_1068

Bertsimas, D., & Tsitsiklis, J. (1993). Simulated Annealing. *Statistical Science*, *8*(1), 10–15. https://doi.org/10.1214/ss/1177011077