UNIVERSITY OF AMSTERDAM

# Travelling Salesman Problem

STOCHASTIC SIMULATION - ASSIGNMENT 3


*Authors:*
Nadav Levi, 11806990,
nadav.levi@student.uva.nl
Yunxiang Li, 13189387,
yunxiang.li@student.uva.nl



**Abstract**

The Travelling Salesman Problem is a problem in mathematical optimization that attempts to find the shortest route between a configuration of cities and has been studied intensively in the last few centuries. While simple to state, finding a solution for a large number of cities is far from trivial. A number of Heuristics have been historically used. The approach taken in the following paper is Simulated Annealing; A method that draws inspiration from the formation of crystals in nature. A total of 3 configurations of length 51, 280 and 442 cities on a 2D grid have been used in order to find the optimal(shortest) route between all cities using simulated annealing. The influence of cooling schedules and Markov Chain's length is investigated. The geometric schedule converges fast and has best route found. The larger Markov Chain's length leads to a faster convergence and a better minimum distance.


December 26, 2021

# 1 Introduction

The *Travelling Salesman Problem*, or in short *TSP*, is a widely known problem in mathematical optimization. Given a set of nodes, usually representing cities, along with edges connecting them representing the distance between them, the travelling salesman problem attempts to find the shortest distance of a route(referred to as *tour*) that starts at an origin and visits every node once before returning to the origin. While the travelling salesman problem is very easy to formulate, finding an optimal solution is far from easy - and is surprisingly very difficult to find. The problem has been intensely researched in computational mathematics and has many other applications, such as in logistics, genetics, manufacturing and telecommunications to name a few[1].

The travelling salesman problem can be formulated in the following way: For $n$ cities, and an $n \times n$ distance matrix $D_{i,j}$ that denotes the distance between city $i$ to $j$, our goal is to find from the set of all possible permutations $\pi$ a closed tour such that each city is visited once and the total length of the tour, given by

$$f(\pi) = \sum_{i=1}^{n-1} \pi_i \tag{1}$$

is minimized.

Throughout the years, researchers have come up with various ways of solving the problem using heuristics. For instance, Appelgate et al manged to solve the TSP with $n = 13506$ using a branch and bound algorithm[1]. Furthermore the problem can also be approached by using evolutionary algorithms to obtain the optimal route, or by population based models. Neural networks have also been used to solve the problem. For a comprehensive overview of some of these methods, refer to [2].

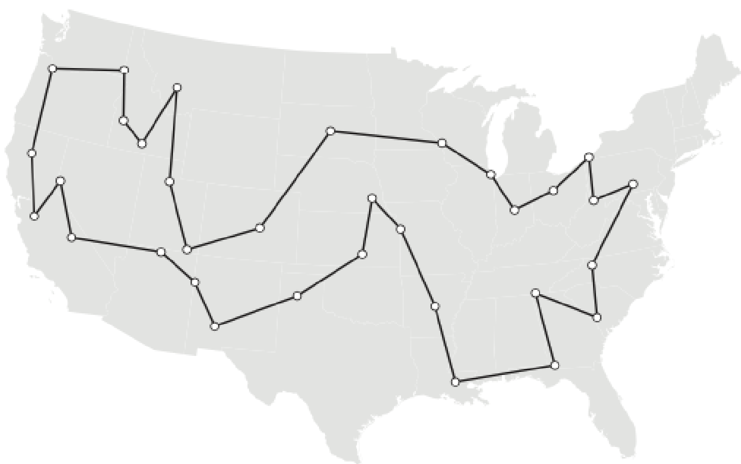

**Figure 1: Optimal 33 cities tour by Appelgate et al[1]**

In mathematical optimization, one is often interested in minimizing a cost function. In the case of the TSP, this is the distance travelled. By minimizing it, we attempt to obtain the shortest possible tour. The approach that we will be investigating in this paper is known as *Simulated Annealing* - A method to search for global minimum/maximum, which was inspired by mimicking the natural process of crystal formulation[3].

The method was initially presented in [4, 5] as a search algorithm for combinatorial optimization problem[6]. Typically, cost functions will have many local maxima and minima points. However, we are simply interested in finding the global minimum or maximum of the cost function without being "stuck" at a local maximum/minimum point[3]. This is the case, for instance by using a simple Greedy algorithm that iterates between all possible permutations until it finds a minima. However, some of the main drawbacks of the method is the number of parameters that need to be tuned. To address that, the parameters can be dynamically changed throughout the execution of the method.

In the following sections, we will describe the theoretical background of the methods used, followed by a description of the approach which we have taken as well as the results obtained.

# 2 Theoretical background

Simulated annealing is a meta-heuristic method that takes inspiration from nature, and more specifically from solid-state physics and the formation of crystals. The properties of convergence towards high quality solutions in problems of combinatorial optimization using simulated annealing are widely known, albeit with a high computational cost. The basic idea which serves as an inspiration is in crystal formation: The material, which has a lattice structure is heated to a high temperature, which in turn allows the atoms to explore different configurations until reaching a state of equilibrium. In statistical mechanics, we know that the probability of a crystal being in a state with energy $E$ follows the Boltzman distribution with the following probability density function:

$$\mathbb{P}[\text{Energy} = E] = \frac{1}{Z(T)} \exp -\frac{E}{kT}$$

where $T$ denotes the temperature, $k$ denotes the Boltzman constant and $Z(T)$ is a normalization function.

Now, how does one translate the ideas described above for using simulated annealing in solving optimization problems? We simply replace the energy function by the cost function $C$ and the temperature is replaced by a control variable $c$ such that we have:

$$\mathbb{P}[\text{Configuration} = i] = \frac{1}{Z(c)} \exp -\frac{C(i)}{c}$$

For a complete overview of the topic, refer to [3].

In essence, the idea of simulated annealing is to accept moves which result in solutions of worse quality than the current solution, which in turn would help escape from local minima[6]. Following this approach, the probability of accepting such a move will decreased when executing the search through the temperature/control parameter.

Then, with the Metropolis acceptance criterion, a candidate solution $y$ is accepted over the current solution $x$ based on the following rule:

$$\begin{cases} 1 \quad \text{if} \quad f(y) < f(x) \\ \exp\left(-\frac{f(y)-f(x}{T}\right) \quad \text{otherwise} \end{cases} \tag{2}$$

With this process, we generate a Markov Chain, whose stationary distribution depends on the control variable $c$ representing the temperature[3]. As $c$ tends to 0, the stationary limit of the Markov chain will have a finite probability for an optimal configuration.

## 3 Methods

### 3.1 Problem formulation

The problem that we will be attempting to solve is finding a global optimum for the TSP using simulated annealing. Throughout our experimentation, we will make the following assumptions:

- There are possible paths between every single city to another on our plane
- The triangle inequality holds: A detour will always result in a longer path. Mathematically speaking, the following relation holds:

$$||z|| \leq ||x|| + ||y||$$

  Where the norm denoted the euclidean distance between the cities.
- Path from city $A$ to city $B$ has the same distance as the reverse path from $B$ to $A$

### 3.2 Operations

We will use a number of operations and helper functions for the simulated annealing:

- Reverse$(i, j)$, often denoted as 2-opt. For example, for a tour starting with a chronological order, applying reverse$(2, 5)$ results in:

$$1, 2, 3, 4, 5, 6, 7, \cdots \implies 1, 5, 4, 3, 2, 6, 7, \ldots$$

- Insert$(i, j)$: Inserting city $j$ in position $i$ of the tour. For example, applying insert $(2, 5)$ yields:

$$1, 2, 3, 4, 5, 6, 7, \cdots \implies 1, 5, 2, 3, 4, 6, 7 \ldots$$

- Swap$(i, j)$: Swapping positions of city $i$ with city $j$. For example, applying swap$(2, 5)$ yields:

$$1, 2, 3, 4, 5, 6, 7, \cdots \implies 1, 5, 3, 4, 2, 6, 7 \ldots$$

Furthermore, we wrote short scripts in order to:

- Calculate an $n \times n$ matrix distance
- Calculate the total length of a given tour

### 3.3 City configurations

In our experimentation, we will be using a total of three cities configurations:

- eil51: A configuration of 51 cities which will serve as our basic configuration to experiment with.
- a280: A configuration of 280 cities, which we can use to further test our algorithm.
- Lastly, pcb442: The largest configuration on which we will use our algorithm.

Note that for all the configurations above, an optimum tour has been provided, which will be used to evaluate to goodness of our solution. The shortest routes for eil51, a280 and pcb442 are approximately 429.98, 2586.77 and 50578.55 respectively.

## 3.4 Cooling schedules

Implementing the method of simulated annealing requires generating a finite sequence of decreasing values for the temperature variable. A frequently used cooling schedule was proposed by Kirkpatrick et al in [5] and consists of:

- Initial temperature $T_0$ which must be sufficiently high.
- Temperature decrease function; usually an exponential decrease function.
- Number of state transitions, for each temperature value.

In our implementation, we have used a total of three different cooling schedules, namely:

- Linear $T_{k+1} = T_k - \frac{T_0}{N}$
- Geometric $T_{k+1} = T_k \times 0.99$
- Logarithmic $T_{k+1} = \frac{T_0}{1+\alpha log(1+k)}$.

# 4 Results

In order to investigate how the different cooling schedules and Markov chain's length affect the convergence, we took a short explore on the initial setting of parameters. We first tested the algorithm on the given TSP of 51 cities and then carried on our study on the one of 280 cities. Best routes acquired for both problems are visualized in the end. [6]

## 4.1 Testing Convergence

The stopping condition for the outer loop is set to 1000 initially and can be further adjusted in further steps according to the convergence rate. In terms of the initial temperature value, 10 simulations was carried on for each $T_0 = 10, 50, 100, 500, 1000$ respectively with the geometric cooling schedule($\alpha = 0.99$) and the Markov Chain's length equals to 100. A similar investigation was taken with variate permutation methods, (insert, swap and inverse). As the box plots in Figures 2 and 3 show, the distribution of 10 best routes found in different initial temperatures and permutations. Swapping two nodes is not a effective mutation as expected. We will take $T0 = 50$ and insert mutation for the following experiments.

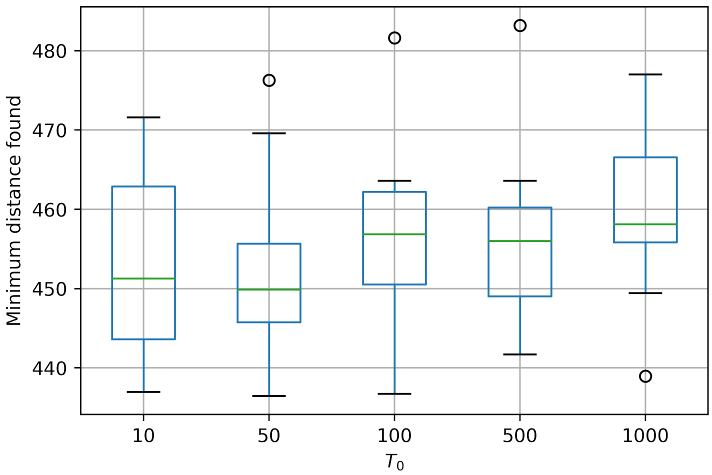


**Figure 2: Investigating $T_0 = 10, 50, 100, 500, 1000$, 10 simulations with the geometric cooling schedule($\alpha = 0.99$) and the Markov Chain's length equals to 100.**

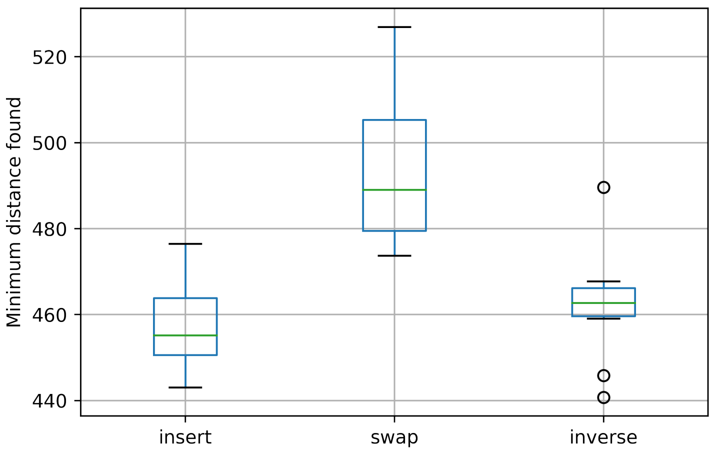


**Figure 3: Investigating three permutation methods, 10 simulations with the geometric cooling schedule($\alpha = 0.99$) and the Markov Chain's length equals to 100.**

## 4.2 Different cooling schedules

Three different cooling schedules, linear, geometric and logarithmic, were implemented and compared for TSP of 280 cities. The Markov Chain's length equals to 100 as before. Figure 4 shows the mean current best route and its 95% confidence interval in each iterations for three cooling schedules. The geometric schedule converges rapidly and obtain best routes in the end at the same time. The typical linear additive schedule behaves normal and linearly converge. The logarithmic schedule was trapped at local minimum shortly. It may due to the bad choice of parameter $\alpha = 2$, $T = \frac{T_0}{1+\alpha log(1+k)}$.

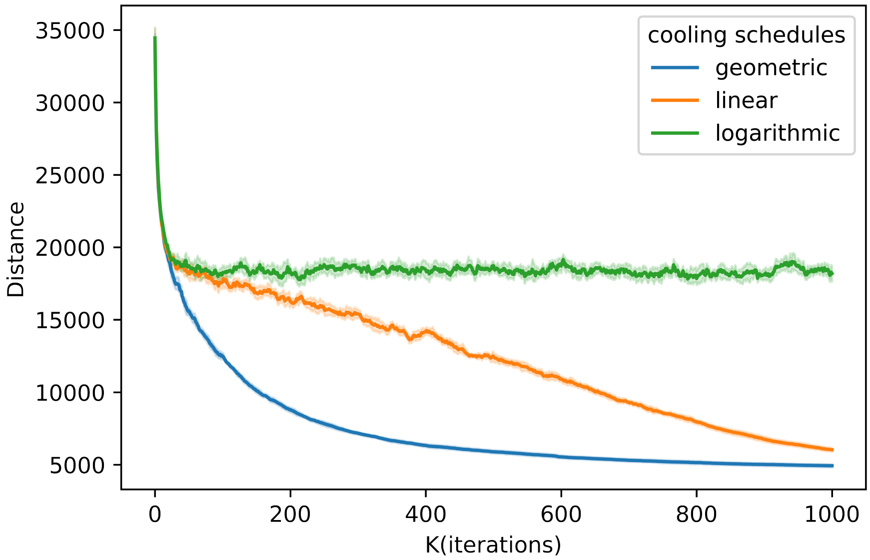


**Figure 4: Investigating three cooling schedules, 10 simulations with $T_0 = 50$, insert permutation and the Markov Chain's length equals to 100.**

### 4.3 Effects of Markov Chain's length

Due to the limited computational power, it's not piratical to conduct a long series of investigating on Markov Chain's length. We performed 10 simulations for each $length = 10, 25, 50, 100, 200$ with geometric schedule. Figure 5 shows the mean current best route and its 95% confidence interval in each iterations for different chain's length. It clearly indicates that a larger chain's length the faster convergence and better routes acquired. The improvement is slowing down as the length increases.

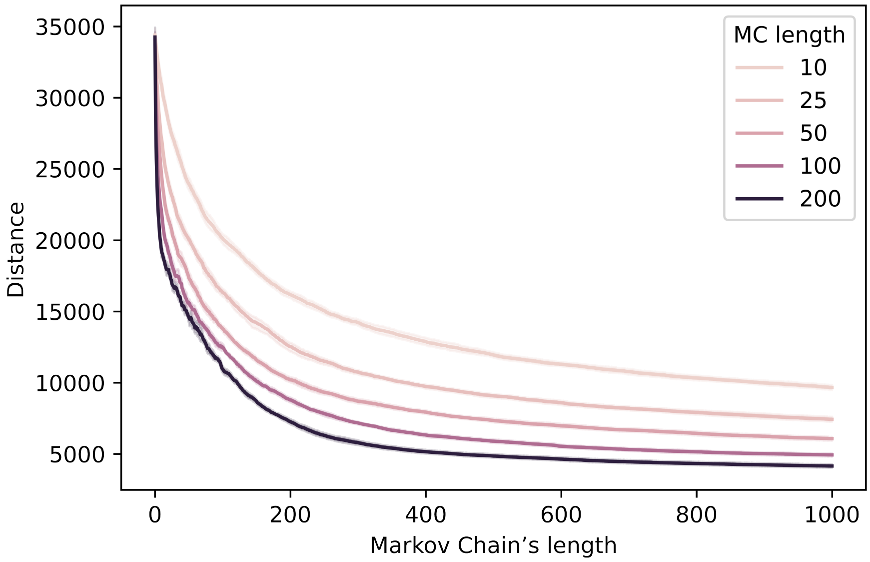


**Figure 5: Investigating length of Markov Chain, 10 simulations with $T_0 = 50$, insert permutation and the geometric cooling schedule.**

## 4.4 Solving TSP

The minimum distance of best route obtained for TSP of 280 cities so far is around 4000. In order to get a minimum closed to the given global optimum, run another simulation with Markov Chain's length equals to 2000. The local minimum is 3108.17 whereas the global minimum is 2586.77.

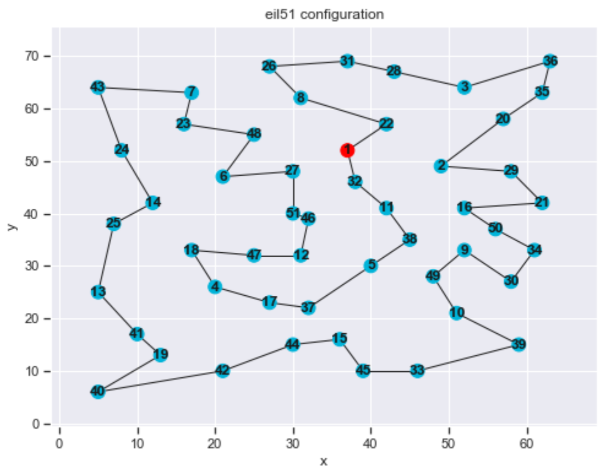


**Figure 6: Global minimum route obtained for eil51.**

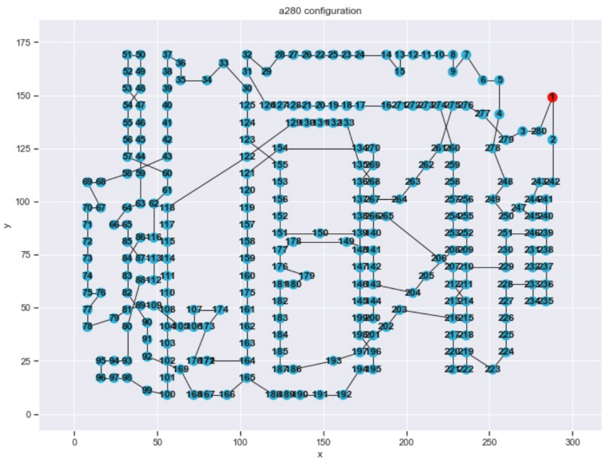


**Figure 7: Local minimum route obtained for a280.**

## 5 Discussion

As discussed above, we have investigated the initial temperature, the permutation methods and how the cooling schedules and Markov Chain's length influence the behavior of convergence. During our experiments, we found classic geometric cooling schedule performs better than the linear one as it converges fast and finds better routes. Additional tests of different parameters should be implemented to conclude whether the logarithmic schedule is a good choice. The larger Markov Chain's length is found to be beneficial to converge at better minimum routes. However, considering the scale of the TSP treated and the computational power costs, it's unclear whether the positive effect still holds for even larger Markov Chain's length enough to converge at global minimum.

Since the minimum obtained for 280 cities is still distance from given optimum, based on our experiments and study, further researches could focus on adaptive cooling schedules and even larger stopping criteria and Markov Chain's length.

## References

[1] D. L. Applegate, ed., *The traveling salesman problem: a computational study.* Princeton series in applied mathematics, Princeton: Princeton University Press, 2006. OCLC: ocm83853510.

[2] F. Greco, ed., *Traveling Salesman Problem.* InTech, Sept. 2008.

[3] *Natural solvers, Simulated Annealing.* Unpublished course material, 2021-12-26.

[4] V. Černỳ, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of optimization theory and applications*, vol. 45, no. 1, pp. 41–51, 1985.

[5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[6] S.-h. Zhan, J. Lin, Z.-j. Zhang, and Y.-w. Zhong, "List-based simulated annealing algorithm for traveling salesman problem," *Computational Intelligence and Neuroscience*, vol. 2016, p. 1–12, 2016.