Assignment 3: Simulated Annealing

November 5, 2023

Group:
1

Students:
Lotte Felius

Amber den Hollander

Lecturer:
Gábor Závodszky

Course:
Stochastic Simulation

Abstract

The aim of this report is to find the shortest path of a 280 dimensional traveling salesman problem. To find this global optima we use Simulated Annealing and tested various sampling methods, cooling schemes, maximum temperatures and Markov Chain length in order to find the appropriate ones for the problem at hand. The sampling methods included swap, invert, insert and a hybrid approach combining all three previously mentioned approaches, taking the best outcome of these three. The cooling schemes implemented were a quick cooling, geometric and Lundy & Mees cooling scheme. We found that the hybrid sampling approach, the Lundy & Mees cooling scheme, a maximum temperature of 1000 and a Markov Chain length as high as possible (in our case 10000). Running a simulation based on the previous we found an optimal route with path length: 2600.4. Based on provided data we can conclude that this is not the shortest route. For future research, in order to find this global optimum, we therefore suggest to consider other cooling schemes and sampling methods. Lastly, in order to increase computing power we advocate that it would be highly beneficial to run this simulation on multiple machines.

1 Introduction

The traveling salesman problem (TSP) is a combinatorial optimization problem. The problem is as follows: A traveling salesman wants to visit a total of n cities, he starts at one of these cities, after visiting all other $n-1$ cities, he returns to the original starting city (Little et al., 1963). This leaves us with the question: In what order should the salesman visit the cities in order to minimize the total traveling distance? Solving the TSP problem leads to a shortest route such as depicted in figure 1.

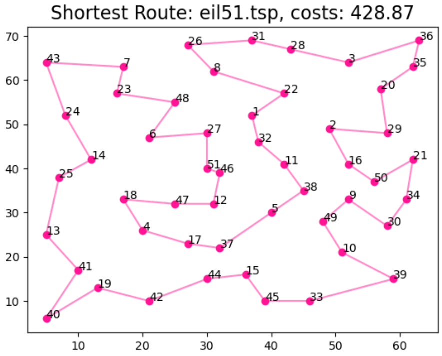

Figure 1: Optimal Route for eil51 (51-dimensional TSP)

To solve such combinatorial optimization problems Simulated Annealing (SA), a probabilistic method, is commonly used. SA allows us to find the global minimum of a cost function with or without multiple local minima, in this case the global minimum is the shortest route (Bertsimas, Tsitsiklis, et al., 1993).

In this report we aim to find the shortest path of a TSP with a dimension of 280, which is the amount of cities to be connected. In order to do this we use SA and experiment with varies values and methods in order to find the appropriate ones for this problem. Four sampling methods will be evaluated, i.e. the swap, inserting and inverting sampling methods in addition to a hybrid approach. We will also asses various cooling schemes including a quick cooling scheme, geometric cooling scheme and Lundy & Mees cooling scheme. Lastly, we will vary the maximum temperature and the Makov Chain length. The TSP used in the experiments is based on provided data. In the problem every city is connected to all other cities. The triangle inequality is applied, i.e. the path of a detour via another city is as long or longer than the direct route between the two cities. In addition the problem is symmetric, meaning that the paths from A tot B and B to A are equally long.

The report is structured as follows: In the methods section we discuss the Simulated Annealing method, we briefly go over the sampling methods and cooling schemes that were implemented and lastly we introduce the problem on which these methods are analyzed. In the results section we compare the results of the sampling methods, cooling schemes, maximum temperatures and Markov Chain length. Lastly we discuss the optimal route. In the discussion we describe the conclusions derived from the results and give suggestions for future research.

2 Methods

2.1 Simulated Annealing

Simulated Annealing (SA) is an optimization method that originated in statistical mechanics and is based on the heating of a substance and it's cooling process (Tayal & Singh, 2019). It is particularly useful for functions that contain numerous local minima, such as a TSP. The outline of the algorithm is briefly described in 1.

Algorithm 1 Simulated Annealing

```
Result: Identified Shortest path
1: Initialization of possible solution, T
2: Sample next possible state with sampling method
3: Generate a random number between 0 and 1
4: Compute $min(\frac{e^{-h(x_{i+1}/T)}}{e^{-h(x_i/T)}}, 1)$
5: Accept $x_{i+1}$ state if U < 1
6: Decrease T and go back to 2
```

Simulated Annealing relies on the Hastings-Metropolis algorithm, which incorporates a Monte Carlo Markov Chain (MCMC) method to simulate complex, distributions (Chib & Greenberg, 1995, pp. 1). In SA, the length of the Markov Chain is equivalent to the amount of iterations that is performed from step 2 to 5. This means that 1 'outer' iteration of the whole SA algorithm includes the length of the Markov's Chain 'inner' iterations. In addition, a stopping criteria must be set in order to let the algorithm finish at a certain point in time. In our case, the stopping criterion is a prefixed amount of 1000 outer loop iterations. This means that the total amount of iterations in one simulation consists of $1000 * MC_{lenght}$ iterations.

To create the algorithm, we use the Python programming language (Python Core Team, 2019) and several standard Python packages such as numpy, scipy and pandas. Within Simulated Annealing, myriad different sampling methods and cooling methods can be used. The latter refers to a method or rate in which the initial temperature decreases. The sampling- and cooling methods we considered include, inter alia, swapping, inserting, inverting and a hybrid method, and a quick cooling scheme, geometric cooling and a cooling scheme based on the Lundy & Mees cooling scheme (Peprah et al., 2017; Zhan et al., 2016; Wang et al., 2019). All methods are elaborated upon in section 2.2 and 2.3.

2.2 Sampling Methods

We experimented with four sampling methods with the aim to find the method with the best outcomes.

2.2.1 Swap

Swap sampling, also known as random sampling, works as follows:

Algorithm 2 Swapping

```
1: Randomly generate 2 different entries
2: Swap places of both entries within the solution
```

(Zhan et al., 2016, p.3).

2.2.2 Inserting

The insert sampling method consists of the following algorithm:

Algorithm 3 Insert

```
Randomly generate 2 different entries
Place the second generated entry right after the first generated entry
```

(Zhan et al., 2016, p.3).

2.2.3 Inverting

The invert sampling method, is based on the algorithm below.

Algorithm 4 Invert

```
Randomly generate 2 different entries
Inverse the cities between the entries
```

(Zhan et al., 2016, p.3).

2.2.4 Hybrid Approach (best of 3)

The fourth and last sampling method we examine is the hybrid sampling approach. This method is based on the paper of Zhan et al. (2016). In essence this method includes all three methods explained in 2.2.1, 2.2.2, 2.2.3. With evaluating the outcomes of all 3 methods, it takes the best performing solution by evaluating the minimum of the costs of all generated solutions. Therefore, instead of solely applying 1 sampling method, we take the best generated solution of three different sampling methods.

2.3 Cooling schemes

Applying an appropriate cooling scheme is fundamental for a SA algorithm to work properly. Preferably, a cooling scheme is relatively slow as this avoids early convergence to local minima (Peprah et al., 2017, pp. 1200). However, different sizes of problems perform better with different cooling temperatures and different initial settings of the temperature. Namely, for e.g. a small problem the temperature can be set relatively low since a small problem tends to converge faster to a (local) minimum or maximum. Inspired by Tayal and Singh (2019), Peprah et al. (2017), and Zhan et al. (2016) we apply various cooling schedules, some slightly adapted.

2.3.1 Cooling cooling scheme

The first cooling scheme is which we call 'cooling' (1). It provides a cooling schedule such that the temperature decreases relatively fast. This cooling schedule is expected to be particularly suitable for really small TSP problems.

$$T_{n+1} = T_n - ln(i+2) \tag{1}$$

Where i represents the current 'outer' iteration the algorithm is in.

2.3.2 Geometric cooling scheme

The geometric cooling schedule is widely used (Tayal & Singh, 2019; Peprah et al., 2017; Zhan et al., 2016) and tends to perform well for different sizes of problems. The mathemathical representation of this cooling scheme is in 2

$$T_{n+1} = T_n * factor \tag{2}$$

Where factor represents a rate between 0.80 en 0.999 typically.

2.3.3 Lundy & Mees variation cooling scheme

Similarly as the geometric cooling schedule, The Lundy and Mees (L&M) algorithm (Lundy & Mees, 1986) tends to perform well for large problems, as observed in Tayal and Singh (2019). Inspired by this algorithm, we constructed another algorithm that converges even slower compared to the regular L&M algorithm (3).

$$T_{n+1} = \frac{T_n}{1+\beta} \tag{3}$$

Where β is defined as 4

$$\beta = \frac{T_{max} - T_{min}}{(i+1) \cdot T_{max} \cdot T_{min}} \tag{4}$$

and i represents the i'th outer iteration the algorithm is currently in. The minimum temperature, T_{min} is in our experiment always set to 1.

2.4 The problem set

The results of the different sampling and cooling methods will be analyzed upon a problem set that consists of 280 cities, where each city is represented by an integer from 1-280. This problem set is made available to us by the University of Amsterdam. As briefly mentioned, we assume that the problem set is completely symmetric which means that distances to different cities (denoted as integers) are exactly equal from city 1 to city 2 and vice versa. The triangle inequality also holds, meaning that when taking a detour via another city, this detour is equally long or longer than the direct route between two cities.

3 Results

In order to find the shortest route in a 280-dimensional TSP, we need to derive optimal parameter settings that allow us to find a global minimum. We aim to find these optimal settings by evaluating different sampling methods, different cooling methods and determining an appropriate maximum for the temperature as and an appropriate length of the Markov Chain.

3.1 Sampling Methods

The evaluation of four different sampling methods is depicted in table 1. The evaluated sampling methods consist of swapping, inserting, inverting and a hybrid method, also referred to as "best of 3" as it takes into account the method that performed the best in a particular iteration in the Markov Chain.

Table 1: Simulation results of 4 sampling methods, for Simulations = 10, MKC = 250, Outer loop = 500, Tmin = 1, Tmax= 400, cooling scheme = Lundy & Mees variation

	Mean (μ)	St.dev (σ)	Confidence Interval (p = 0.95)
Swap	34003.90	604.16	$33629.44 \leq X \leq 34378.36$
Insert	34155.50	904.72	$33594.76 \leq X \leq 34716.24$
Invert	34148.70	631.69	$33757.18 \leq X \leq 34540.22$
Hybrid	2854.26	74.12	$2808.33 \leq X \leq 2900.20$

Table 1 shows that hybrid sampling outperforms the other individual sampling methods. A downfall however corresponding to hybrid sampling is that it takes relatively long as it needs to evaluate all three other sampling methods and subsequently take a minimum. However, hybrid sampling performs so much better compared to other sampling methods that it is convenient to use hybrid sampling for further evaluation.

3.2 Cooling Schemes

Cooling schemes are a key factor in determining whether a SA algorithm performs well. Moreover, problems with different dimensions often require different cooling schemes. Figure 2 shows how fast three cooling schemes, 'cooling', 'geometric cooling' and 'Lundy & Mees variation cooling' (Tayal & Singh, 2019), decrease. As one can derive from this graph, the Lundy & Mees variation cooling method decreases very fast in the beginning, but very slow after a short while.

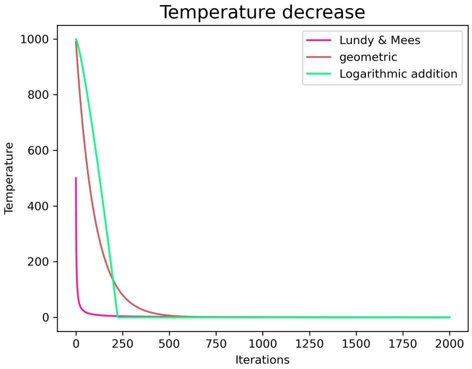

Figure 2: Temperature decrease, MK Chain = 500

After simulating all different cooling schemes, we derived the data depicted in table 2.

Table 2: Simulation results of the three cooling schemes, for simulations = 10, MKC = 500, Outer loop = 1000, Tmin = 1, Tmax= 400, Sampling method = Hybrid Sampling

	Mean (μ)	St.dev (σ)	Confidence Interval (p = 0.95)
Cooling	8210.49	296.77	$7936.55 \leq X \leq 8304.43$
L&M Variation	2753.12	24.46	$2737.96 \leq X \leq 2768.28$
Geometric	2792.98	67.32	$2751.25 \leq X \leq 2834.70$

The results show that the Lundy & Mees variation cooling scheme performs best; it not only leads to the lowest average, but also the lowest standard deviation.

3.2.1 Alternating Starting Temperatures

In order to find the appropriate maximum temperature for the problem at hand, we simulated with various maximum temperatures, of which the results are displayed in table 3.

Table 3: Simulation results for Simulations=10, MKC = 500, Outer loop = 1000, Tmin = 1, Cooling Scheme = Lundy & Mees variaton

Tmax	**Mean** (μ)	**St.dev** (σ)	**Confidence Interval** (p = 0.95)
100	2819.59	67.93	$2752.72 \leq X \leq 2790.06$
250	2776.61	25.06	$2761.08 \leq X \leq 2792.14$
400	2771.39	30.11	$2752.73 \leq X \leq 2790.06$
500	2745.35	35.02	$2723.64 \leq X \leq 2767.05$
1000	2723.76	43.87	$2696.57 \leq X \leq 2750.95$
2000	2770.43	46.18	$2741.80 \leq X \leq 2799.05$

In table 3 we can see that with the increase of the maximum temperature the mean decreases. Although the mean increases again when reaching the max temperature of 2000. This is likely because as previously mentioned the appropriate temperature is problem dependant, smaller problems converge faster with lower temperatures. We therefore use a max temperature of 1000 for the remaining simulations.

3.3 Markov Chain Length

The length of the Markov Chain determines how many times new solutions will be randomly generated. This means that by definition, the bigger the chain, the more likely it is that a 'better' solution will be randomly generated. This is underlined by the results depicted in table 4.

Table 4: Simulation results for Simulations=10, Outer loop=1000, Tmin = 1, Tmax=1000 Cooling Scheme = Lundy & Mees variation

MC length	Mean (μ)	St.dev (σ)	Confidence Interval (p = 0.95)
100	2819.59	67.93	$2752.72 \leq X \leq 2790.06$
500	2723.76	43.87	$2696.57 \leq X \leq 2750.95$
750	2723.28	32.97	$2702.85 \leq X \leq 2743.72$
1000	2686.0	44.85	$2658.20 \leq X \leq 2713.80$
3000	2646.04	30.93	$2626.87 \leq X \leq 2665.21$

It is evident that a large size of the Markov Chain leads to better performance in the sense of finding a (global) minimum. The results depicted in table 4 confirm this by showing that with the size of the chain set to 3000 iterations the resulting average is the lowest as well as the corresponding standard deviation and confidence interval. However, using a large Markov Chain is rather expensive in terms of computation power and takes very long to simulate. Therefore, as in most simulation problems, a trade-off has to be made between available computation power and the quality of the result.

3.4 Optimal Route

Using the knowledge we derived from testing optimal parameters settings, we set the maximum temperature at 1000 and the Markov Chain length as high as possible feasible with computing power (10000 iterations) while using hybrid sampling and the L&M variation cooling scheme. Due the high amount of computations needed to execute this simulation, we only simulated it four times. However, it gave us the best result we derived until now; 2600.4. The optimal route we found is depicted in figure 3 and its corresponding costs (i.e. path length) is 2600.4.

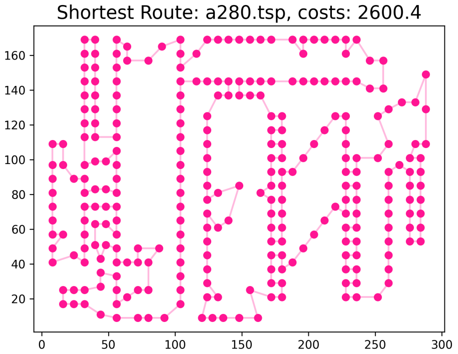

Figure 3: Optimal Route for a280, a 280 dimensional problem

4 Discussion

There does not exists an algorithm that always gives the optimal path for a TSP, if not using a brute-force approach. The latter is often not possible since for a TSP the problem space is often huge, which leads to an infinite amount of possible combinations to evaluate. Hence, aiming to solve a 280 dimensional TSP, we used Simulated Annealing and herewith tried various sampling methods and cooling schemes that possibly improve our solutions. The sampling methods consisted of swapping, inserting, inverting and hybrid sampling. The latter sampling method easily outperforms the other sampling methods, although in return it takes most computing power as it evaluates the three other sampling methods at once. In addition, the L&M variation cooling scheme tends to outperform geometric cooling, although with L&M the temperature decreases much faster at the start. It seems to decrease so fast, that one should consider making a cooling scheme that is able to explore the initial problem space slightly longer such that the algorithm eventually does not get stuck in a local minimum.

Moreover, the optimal initial temperature (T_{max}) is set to 1000 which seems relatively high. However, for this particular 280 dimensional problem accompanied by the L&M variation cooling scheme this seems to work. It is very likely that alternating cooling schemes perform much better on a different starting temperature, possibly much lower than 1000. In addition, the optimal Markov Chain length is the biggest possible chain length corresponding to the available computing power. This is, as in most simulation problems, a crucial trade-off as larger Markov Chain lengths lead to more accurate results. Due to limited available computing power we were only able to do 10 simulations for testing each problem, which explains the large standard deviations and confidence intervals. Since this problem aims to find a global minimum however, the focus should primarily be on finding the path with the lowest possible 'costs', i.e. length, and the statistics are mainly used to derive which schemes and methods lead to a good solution.

The global minimum is not found for this 280 dimensional TSP. Therefore, other cooling schedules such as list-based cooling (Zhan et al., 2016) could be applied as this possibly leads to a better performance of the SA algorithm. Moreover, a myriad of better performing sampling methods could be implemented, which should be further explored. In addition, for solving TSP it can be very beneficial to run the problem on multiple machines in order to increase the runtime of a simulation as this truly is a bottleneck in TSP's.

References

Little, J. D., Murty, K. G., Sweeney, D. W., & Karel, C. (1963). An algorithm for the traveling salesman problem. *Operations research*, *11*(6), 972–989.

Bertsimas, D., Tsitsiklis, J. et al. (1993). Simulated annealing. *Statistical science*, *8*(1), 10–15.

Tayal, A., & Singh, S. P. (2019). Analysis of simulated annealing cooling schemas for design of optimal flexible layout under uncertain dynamic product demand. *International Journal of Operational Research*, *34*(1), 85–103.

Chib, S., & Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *The american statistician*, *49*(4), 327–335.

Python Core Team. (2019). *Python: A dynamic, open source programming language* [Python version 3.8]. Python Software Foundation. https://www.python.org/

Peprah, A. K., Appiah, S. K., Amponsah, S. K., et al. (2017). An optimal cooling schedule using a simulated annealing based approach. *Applied Mathematics*, *8*(08), 1195.

Zhan, S.-h., Lin, J., Zhang, Z.-j., & Zhong, Y.-w. (2016). List-based simulated annealing algorithm for traveling salesman problem. *Computational intelligence and neuroscience*, *2016*.

Wang, L., Cai, R., Lin, M., & Zhong, Y. (2019). Enhanced list-based simulated annealing algorithm for large-scale traveling salesman problem. *IEEE Access*, *7*, 144366–144380.

Lundy, M., & Mees, A. (1986). Convergence of an annealing algorithm. *Mathematical programming*, *34*(1), 111–124.