The Travelling Salesman Problem in the context of Natural Solvers

Andrea Maestri, 14055031, andreamaestrimaestri@gmail.com
Aron Golombek, 14064103, aron.golombek@student.uva.nl

December 2021

Abstract

Natural Solvers are mathematical methods inspired by natural processes, simulated annealing in particular is inspired by the formation of perfect crystals during a gradual cooling process called annealing. This report investigates how three distinct factors, namely the mutation method, Markov chain length and cooling schedule affect the solution obtained for the Travelling Salesman problem using the Simulated Annealing algorithm. A central result is that the best mutation style is an adaptive mutation, a hybrid between the *2-opt* mutation and the *Insert* mutation. Furthermore, it is shown that increasing the length of the Markov Chain increases the probability of finding a global minimum. Lastly, it is found that from the cooling schedules investigated, the one with a geometrically decremented temperature performs best. Finally, the optimized algorithm is applied to three distinct travelling salesman problems with increasing number of nodes. The best results found by the algorithm are 426, 2660, 53257 compared to the benchmark solutions of 426, 2579 and 50778 respectively.

Introduction

The Travelling Salesman Problem is a well known and well documented computational problem which aims to find the minimum distance between a certain number of cities, visiting each city once. Solving this by trying all combinations using a computer may seem intuitive, however with an increasing number of cities n_{nodes} the possible combinations of trips increases rapidly, specifically with $(n_{nodes} - 1)!$. Even for modern computers, solving this problem, trying all of the possible configurations, can have an immense computational cost for a large n_{nodes}. This raises the question of whether more efficient methods for solving such a problem exist [1].

This paper will specifically explore how natural solvers, namely Simulated Annealing can be used as an optimisation strategy to find the global minimum of the Travelling Salesman Problem. In this case, the problems explored are the eil51, the a280 and the pcb442[1]. Natural Solvers are a group of methods inspired by natures ability to find optimal solutions for highly dimensional and thus complex problems. Common examples for this are protein folding, slow cooling into perfect crystals as well as evolution itself [1]. Simulated Annealing is an iterative improvement algorithm, where, from a starting configuration, other possibilities are checked systematically. While this may drastically decrease the computational effort, for highly complex and large problems where not all possible configurations can be evaluated, it is not possible to determine whether a global or merely a local minimum has been found.

Introductory theory concerning the travelling salesman problem and the implementation of methods to solve it can be found in the following literature [2], [3], [4]. While additional important theoretical results are documented in [5]. Moreover, the notation used in this paper is consistent with [1].

This report consists of four sections. Section 1 will introduce some general theory on the topic. Subsequently, section 2 will elaborate on how this can be implemented to preform experiments. Section 3 will detail the experiments and their outcomes followed by section 4 where some general conclusions will be drawn.

[1]These problems simply consist in a different number of cities to visit. Respectively 51, 280 and 442.

1 Theory

This section will introduce some fundamental mathematical theory required to formulate an approach to solving the travelling salesman problem using natural solvers, namely simulated annealing.

1.1 Iterative Improvement Algorithms

An iterative improvement algorithm starts with a given configuration and subsequently proceeds to test neighbouring configurations (i.e. by applying slight perturbations to the original configuration) until a critical point of the desired type has been found. A classic example of this is the hill-climbing algorithm which is schematically depicted in fig. 1. From the starting point a step in the direction of the gradient necessary, depending on whether a minimum or maximum is sought, is taken repeatedly until the algorithm converges to a specific configuration.

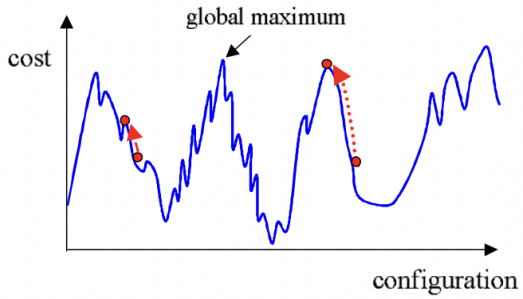


Figure 1: Hill climbing algorithm, the figure is taken from [1].

Looking at fig. 1 it becomes evident that while this may work effectively for smooth functions with no local minima or maxima, once a more complex function is introduced, the algorithm will most likely merely converge to a local critical point rather than the global one which is sought. While simply running the algorithm repeatedly for different initial configurations may seem like an intuitive way to approach this problem, it will not be efficient for complex problems as it does not address the underlying issue. However, there are several approaches to improve this algorithm to avoid the stalling at local critical points two of which are listed below.

1. Using information from previous runs or introducing generation algorithms with higher complexity allowing to move on from local critical points.

2. Accepting configurations going in the 'wrong' direction under specific conditions.

The first method in essence is how a specific type of natural solvers namely genetic algorithms work. However, this report will focus on the second method which describes the idea behind simulated annealing, a different natural solver method.

1.2 Simulated Annealing

In solid-state physics, it can be desirable to achieve perfect crystals. To do this, one may use an experimental technique called annealing. The key idea is to start from a high temperature and decrease it slowly. The crystal will move into new configurations, of which the probability is given by the Boltzmann distribution[1]:

$$P(Energy = E) = \frac{1}{Z(T)} exp\left(-\frac{E}{k_B T}\right). \tag{1}$$

Here T is the temperature, k_B the Boltzmann constant and $Z(T)$ is a normalization function. For a high temperature, every configuration has the same probability. However, when T goes to zero, only the configurations with the smallest energy are possible. This process often leads to an almost perfect crystal if the temperature is decreased slowly enough as it allows sub-optimal configurations to be altered again. This is analogous to the solution of the travelling salesman problem jumping out of a local minimum.

To allow this process to solve other optimization problems, one must first translate the Boltzmann distribution, then each step in the annealing should be linked to an equivalent process in the optimization domain. The translated Boltzmann equation is:

$$P(configuration = i) = \frac{1}{Z(T)} exp\left(-\frac{C(i)}{c}\right). \tag{2}$$

Here $C(i)$ is the cost of the configuration i and c is the control parameter[2]. The difficulty of this approach is to sample a configuration according to eq. (2). In order to do this, the Metropolis Algorithm is used.

Before explaining the Metropolis Algorithm, it is essential to introduce the Markov Chain. A Markov chain is a collection of status $\{X_n, n > 0\}$ where X_n are random variables (random paths in this case) with a transition probability between i and j of $P_{i,j}$[5] with

$$\sum_{j=1}^{N} P_{i,j} = 1, \quad i = 1, ..., N.$$

Moreover, a Markov chain is irreducible if for each pair of i and j there is a positive probability, starting from i, that the process will never enter the state j. It can be proven that if an irreducible Markov chain is long enough and

[2]In this report, the word *temperature* and *control parameter*. will refer to the same parameter c

aperiodic[3], the probability that the system will be in state j is $P\{X_n = j\} = \pi_j$ if the initial state is distributed accordingly to the $\{\pi_j,\ j = 1, ..., N\}$. The set $\{\pi_j,\ j = 1, ..., N\}$ are called stationary probabilities, and it can be shown that they are unique. In the next section, a way to construct a Markov chain whose limiting probabilities are π_j will be shown.

1.2.1 Metropolis Algorithm

Given a random walk with transition probability $P_{i,j}$ (the product of the probability of generating the state j from i times the probability of accepting it $A_{i,j}$). If $P_{i,j}$ define an irreducible, aperiodic Markov chain, then if

$$A_{i,j} = min\{1, P(j)/P(i)\},$$

it can be proven that the system will converge to the set of the stationary probabilities

$$\pi_j = \frac{1}{Z(T)} exp\left(-\frac{C(j)}{c}\right), \tag{3}$$

if the following procedure is followed.

Algorithm 1 Metropolis Algorithm

for $i \leq$ Length of the Markov Chain **do**
  From a state i, propose a new state $i + 1$.
  Calculate cost difference $\Delta C = C_{i+1} - C_i$.
  Calculate $A_{i,i+1} = min[1, e^{-\Delta C/c}]$.
  if $A_{i,i+1} = 1$ **then**
    accept j as the new state.
  else Generate a random number U between 0 and 1
    if $U \leq A_{i,i+1}$ **then**
      accept $i + 1$ as the new state
    else
      keep i as the new state
    end if
  end if

One of the benefits of this method is that a normalisation function is not required.

Now that the Metropolis algorithm has been explained, is it possible to explain how this method is useful to reach a configuration with a small configuration's cost. The key idea is that, reducing the temperature gradually after each Markov chain will lead to the system slowly converging to the optimal value (i.e. the one with the smallest configuration's cost), since

$$\lim_{c \to 0} \pi_j = \delta(j), \tag{4}$$

[3]See [5] for a theoretical insight on this.

where j is the configuration with the smallest cost. Such an algorithm is called *Simulated Annealing* and it will be described in detail in section 2.

1.3 Theoretical background for parameters

Even though the majority of parameter tuning will be discussed in section 3, it is nonetheless important to underline that certain parameters can not be outside of certain boundaries due to the mathematical properties of the Markov Chain. Therefore it is not worth exploring them outside this range. Specifically, literature [2, 1] shows that the initial acceptance probability of the simulated annealing should be over 50%, and that the temperature (i.e. the control parameter) must be decreased very slowly to not move too fast from the asymptotic behaviour of the chain. Moreover, different methods to find the optimal initial temperature can be used [3]. However, these methods are outside the scope of this report.

2 Methods

This section will explain how a solution to the Travelling Salesman Problem can be found using a python implementation of the theory discussed in the previous section. Moreover, the Welch-test which will later be used to compare the outcomes of different experiments will be introduced briefly.

2.1 Simulated Annealing in Python

As previously mentioned in section 1, the simulated annealing algorithm could slightly change due to different approaches. This report will analyse two cooling schedules and three types of mutations for the travelling salesman problem. Using the Metropolis algorithm described in section 1, it is possible to describe the implementation of simulated annealing using a homogeneous Markov chain[1]. This is described by the following algorithm:

Algorithm 2 Simulated Annealing

```
Put the system in a random state.
Start with a high temperature.
while stopping criterion is not met do
    Make a Markov chain with the Metropolis algorithm.
    Lower the temperature.
    Check stop criterion.
end while
```

In order to increase the confidence in the mean value, this process is preformed several times for each experiment. The optimal number for this will be investigated and the result can be found in section 3. The last path's length of each simulation is saved in an array, using this a 95% confidence interval is constructed. The last argument not treated so far is how a new path configuration can be constructed from the previous one. In order to do this, it is possible to use a mutation algorithm. Which algorithm to use is problem-dependent. In this report, four mutation methods are explored.

2.1.1 2-opt Mutation

A commonly used mutation for this type of problem is the 2-opt mutation which is described by the following pseudo-code:

Algorithm 3 2-opt(route, i, k)

```
Randomly pick two nodes i and k in the array of the route.
All nodes before node i and after node k remain in the same order.
The nodes between and including i and k are reversed in order.
```

The central idea behind this mechanism is to take a route that crosses over itself and reorder it in a way such that it avoids crossovers.

2.1.2 Inserting Mutation

A different way to generate a mutation is the so-called *inserting* mutation. In this mutation, one random node of the path is simply moved to a different random position.

2.1.3 Adaptive Mutation

The first 4/5 of the total steps in the simulated annealing are done using the *2-opt* mutation, while the rest are done using the *inserting* mutation.

2.1.4 Swapping Mutation

The final mutation treated in this report is the *swapping* mutation. With this mutation, two nodes are randomly chosen and swapped.

2.1.5 Cooling Schedules

Besides investigating different methods for creating mutations, the report will analyse the effect of using different *cooling schedules*. These will be discussed in this section.

As mentioned before, the term *cooling schedule* refers to a process decreasing the control variable in a systematic manner. The following information is required:

- Initial values of the parameters.
- Final value of the temperature i.e. the stopping criterion.
- Length of the Markov chain.
- Rule to decrease the control variable.

The two different cooling schedules used in this report decrease the temperature according to eq. (5) and eq. (6) respectively.

$$T^* = \alpha T, \tag{5}$$

$$T^* = T\left(1 + \frac{ln(1+\delta)T}{3\sigma(T)}\right)^{-1}, \tag{6}$$

where $\sigma(T)$ is the standard deviation in the cost function of the current chain and δ controls how much the probability functions may differ between two chains. Moreover, in this paper, the temperature is decreased 300 times before stopping the algorithm.

2.2 Parameter tuning

As previously mentioned, after the mathematical analysis of the system, a problem tailored parameter tuning is required in order to obtain the optimal results. All of the tuning done, from now on, will be for the a280 problem. The other two problems, eil51 and pcb442 will be discussed in section 3.

2.2.1 Starting Temperature

In order to avoid the convergence of the path into a local minimum, the starting temperature must be high enough. In the Travelling Salesman Problem, this starting point should guarantee that the acceptance probability of a new configuration will be above 50%. Starting from $T = \sqrt{n_{nodes}} \cdot K$, different values of K were tested. After different trials, in order to balance the computational cost and the convergence of the algorithm to a good local minimum, the starting T was chosen to be $T = \sqrt{n_{nodes}} \cdot 5$ for the a280 problem. With this value, for all the following cases, the acceptance probability will be over 55%.

2.3 Hypothesis testing

Assuming the Markov chain is long enough to consider the system in a state i which probability is a stationary probability π_j, the confidence interval will be constructed after multiple runs. In order to compare the simulated values from different simulations (i.e. using different mutations and cooling schedules), is it possible to use the Welch-test. The Welch-test require to compare the actual value:

$$A_{val} = \frac{\bar{L_1} - \bar{L_2}}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \ , \tag{7}$$

to the critical value corresponding to a confidence level of 95% from a T-distribution. Using this value, the hypotheses that there is not a statistically significant difference between the two means will be accepted if $|A_{val}| < |C_{val}|$.

3 Results

3.1 Exploring different mutation algorithms

Using the initial temperature for the a280 problem from section 2, the focus of the report can move to how different mutation methods affect the convergence of the simulation towards the global optimum. For each of these experiments the cooling schedule used is the one given in eq. (5), with $\alpha = 0.975$. The simulation was terminated when the temperature had been decreased 300 times. Moreover, the length of the Markov Chain has been set to 2000. The effect of Markov Chain length on convergence will be explored later in this section.

Mutation	**Best Length**	**Simulations**	**95 % C.I.**	**Comp. time**
Swapping	4315	40	[4667, 4817]	6h 40'
Inserting	3153	40	[3379, 3476]	6h 40'
2-opt	2746	40	[2847, 2881]	6h 40'
Adaptive	2716	40	[2795, 2825]	6h 40'

Table 1: Confidence interval and computational time for different mutations and for $N = 2000$

From table 1 it is evident that the *2-opt* and the *adaptive mutation* perform better compared to the others, with the latter providing slightly better results than the more common *2-opt*.

3.2 Exploring different Markov chain lengths

In this paper Markov chain length, using the notation of [1], refers to the length of a Markov chain for a fixed value of temperature. Therefore, the total length of the chain in the simulated annealing process will be the Markov Chain length times the number of temperature decrements. The Markov chain's length is a crucial parameter value in the simulated annealing. If this value is too small, especially for large n_{nodes} is very big, it is expected that the algorithm will not converge to a good minimum. However, conversely, this value can not be set arbitrarily big as this would drastically increase the computational expense. The final decision for the Markov chain's length is made based on the following factors:

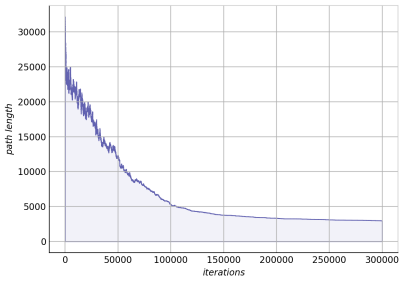
- Computational cost of the simulation.
- Convergence to a good local or global minimum.
- Good statistical relevance of the simulated value.

In order to understand what is the optimal value, different experiments were conducted. For all of them, the cooling schedule used is given by eq. (5), with $\alpha = 0.975$. The simulation was run until the temperature had been decreased

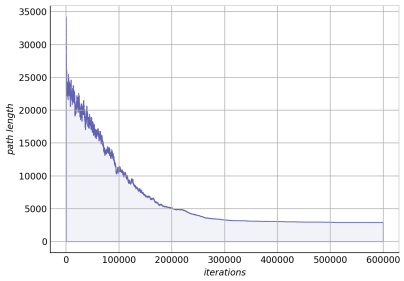
300 times. Furthermore, the 2-opt method for mutations was used. The experiment was preformed for the a280 problem, with the values of N tested being $1000, 2000, 3000, 4500$ and 6000. For the first three experiments, the number of simulations was set to 40. For $N = 4500$ and $N = 6000$ the number of simulations was set to 30 and 20 respectively to avoid the computational time exceeding 13 hours[4].

N	**Best Length**	**Simulations**	**95 % C.I.**	**Comp. time**
1000	2841	40	[2983, 3031]	3h 20'
2000	2746	40	[2847, 2881]	6h 40'
3000	2741	40	[2805, 2831]	10h
4500	2687	30	[2760, 2788]	12h 20'
6000	2649	20	[2731, 2784]	10h 40'

Table 2: Confidence interval and computational time for different values of the Markov Chain's length.



(a) Length progression for the a280 problem and $N = 1000$.

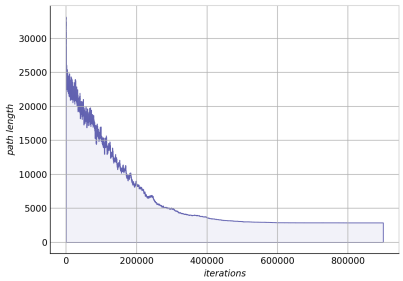


(b) Length progression for the a280 problem and $N = 2000$.

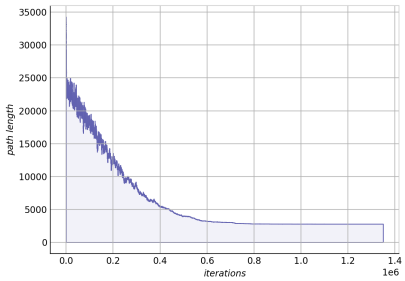
Figure 2: a280 problem with initial control parameter $T = \sqrt{n_{cities}} \cdot 5$ and 300 decrements according to eq. (5).

From figures 2,3 and 4, it becomes evident that increasing the length of the Markov chain results in a slower rate of convergence. This is due to the fact that more steps are preformed by the algorithm before the temperature is decreased. Moreover, the variance at the beginning of the simulation is higher for larger values of N. However, as seen in table 2, both the confidence interval and the best solution become better when increasing the length of the Markov Chain.

[4]This time may vary due to different computers being used for the simulations.



(a) Length progression for the a280 problem and $N = 3000$.



(b) Length progression for the a280 problem and $N = 4500$.

Figure 3: a280 problem with initial control parameter $T = \sqrt{n_{cities}} \cdot 5$ and 300 decrements according to eq. (5).

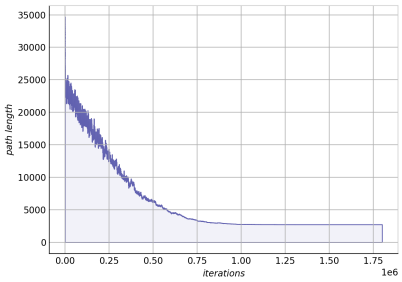


Figure 4: Length progression for the a280 problem and $N = 6000$ and starting control parameter $T = \sqrt{n_{cities}} \cdot 5$ and 300 decrements according to eq. (5).

Pairwise Welch test: p-values

	N = 1000	**N = 2000**	**N = 3000**	**N = 4500**	**N = 6000**
N = 1000	-				
N = 2000	$\ll 0.05$	-			
N = 3000	$\ll 0.05$	$\ll 0.05$	-		
N = 4500	$\ll 0.05$	$\ll 0.05$	$\ll 0.05$	-	
N = 6000	$\ll 0.05$	$\ll 0.05$	$\ll 0.05$	0.12	-

Table 3: p-values from Welch's test for different values of the Markov chain's length.

Using the Welch-test for all the combinations of N, except for the couple $N = 4500, 6000$, the hypothesis that the difference between the simulated values is due to the stochasticity of the simulation should be rejected. On the other hand, for $N = 4500, 6000$:

- $A_{val} = 1.2, \quad df = 29, \quad C_{val} = 2.04.$

Therefore, the hypothesis that the difference between $N = 4500, 6000$ is not statistically significant should be accepted with a confidence level of 95%. Finally, after considering the computational cost of each simulation, the value of N for the $a280$ problem is set to 4500 for the remainder of this report in order to balance computational efficiency and the accuracy of the results.

3.3 Cooling schedules comparison

The final factor which will be explored in this report are different cooling schedules, i.e. in which manner the temperature is decreased. The two cooling schedules given by eq. (5) and eq. (6) were tested for the a280 problem. For eq. (6) $ln(1+\delta)$ was set to 1. The N was set to 4500, the 2-opt mutation method was used and both the starting temperature as well as the number of temperature decrements remain the same as for the previous experiments.

Cooling schedule	Best Length	Simulations	95 % C.I.	Comp. time
Cooling 1	2687	30	[2760, 2788]	12h 20'
Cooling 2	2746	30	[2822, 2863]	14h

Table 4: Confidence interval and computational time for different mutations and for $N = 4500$.

Using the Welch-test, the resulting p-value is $\ll 0.05$. Therefore, the hypotheses that the result coming from the different cooling schedules are compatible can be rejected with a 95% confidence level. Furthermore, it is evident that the geometrically decremented temperature (i.e. eq. (5)) performs better given the choice of the parameters.

3.4 Tests on different problems

In the previous subsections the best configuration for the algorithm in terms of Markov chain length, cooling schedule and mutation has been identified. The next step is to apply the algorithm to different travelling salesman problems namely, eil51, a280 and pcb442 and evaluate whether the result obtained is acceptable.

Problem	Starting T	N	Best Length	Benchmark	Simulations	95 % C.I.	Comp. time
eil51	$3\sqrt{51}$	1000	426	426	30	[429, 433]	40'
a280	$5\sqrt{280}$	4500	2660	2579	30	[2722, 2752]	12h 40'
pcb442	$70\sqrt{442}$	6000	53257	50778	30	[54291, 54746]	20h 30'

Table 5: Results with optimal configurations for the three different problems.

Using the cooling schedule given by eq. (5) with 300 decrements and the adaptive mutation, it can be seen in table 5 that the best result obtained for all

problems is relatively close (or equal) to the benchmark solution. Furthermore, the average of the runs is also close to the global optimum.

In order to see where the simulated solution differs from the benchmark solution for each of the problems, the paths have been plotted below.

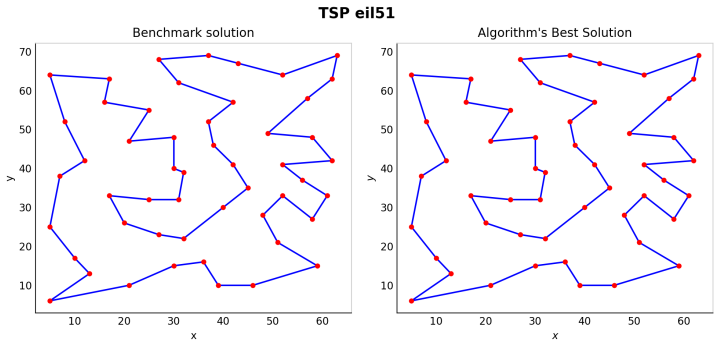


Figure 5: Benchmark vs Best solution for the eil51 problem.

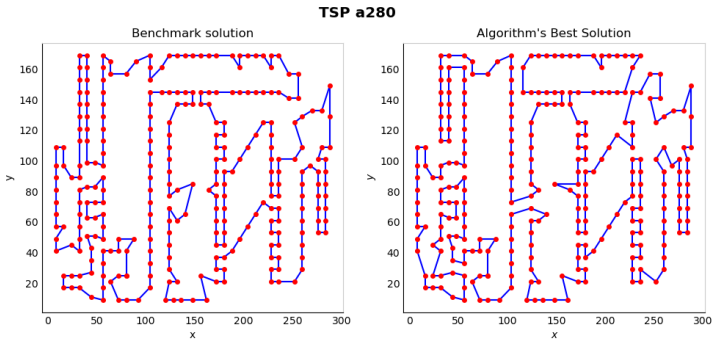


Figure 6: Benchmark vs Best solution for the a280 problem.

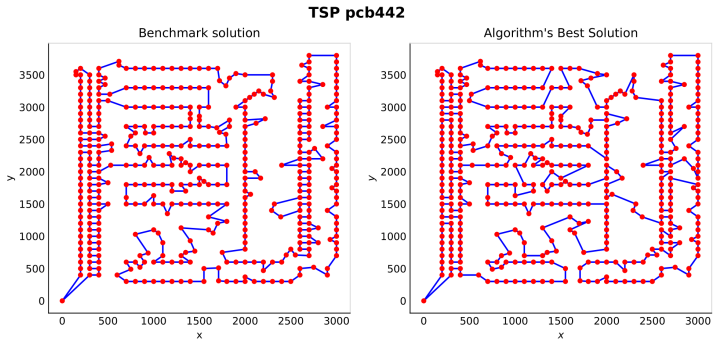


Figure 7: Benchmark vs Best solution for the pcb442 problem.

4 Conclusion

As explored in this report, studying a conceptually easy problem like the travelling salesman problem, can grant insight into how the Metropolis algorithm and Markov chain can be powerful tools when generating random variables distributed accordingly to a probability function. Despite the fact that each parameter value used is strongly problem-related, it is nonetheless possible to draw some general conclusions.

Firstly, using the mathematical background of simulated annealing, it was clear that a slowly decremented temperature leads to a convergence towards the global optimum. This is due to the fact that the current state, will only move slightly from the asymptotic behaviour. This has been confirmed in the experiments.

Moreover, different mutation methods were explored. The 2-opt and the adaptive mutation gave the best results. A possible explanation for this is that, after each iteration, these methods generate a configuration that is closer to the previous one compared to, for example, the swapping mutation. This similarity is an important requirement for the correct generation of a Markov chain.

In order to obtain a result that is closer to the optimal value (i.e the Markov chain converges to the stationary probability), the length of the Markov chain should be ideally infinite. Since this is not possible, the length of the Markov chain is a parameter that should be tuned before attempting to find a solution. The results clearly show that this parameter strongly influences the convergence of the simulation to the global optimum, with better paths obtained for the larger values of N. However, increasing N increases computational cost and thus a trade-off needs to be made as with a lower N the experiment can be repeated more often leading to a higher statistical relevance of the simulated value.

Furthermore, two different cooling schedules were investigated. The best results were achieved using simple geometrical decrements of the temperature,

with a fixed decrement ratio. The second investigated schedule, besides being more involved and requiring more computational operations, also returned worse results.

After having explored the different configurations, the most promising one was applied to all three problems. As expected, for the eil51 problem, the global optimum was reached even with a relatively short Markov chain. However, for the two remaining problems, only local minima were found. A possible explanation for this is can be seen in the fact that for these problems the number of possible paths is enormous, specifically 279! and 441!. Therefore, in order to converge to the global optimum, a longer Markov chain is required.

A proposition for future research would be to investigate the minimum length of the Markov chain required to converge to a better local minimum for the a280 and pcb442 problems. Moreover, due to the impact of the mutation which has been demonstrated in this report, it comes naturally to investigate a more involved method of creating mutations at different stages depending on the state of the chain. Additionally, analysing the effect of using different cooling schedules may lead to a more effective algorithm. A suggestion for this would be using logarithmic decrements of the temperature after each step in the Markov chain (i.e. in-homogeneous Markov chain), and not merely after an entire chain is generated with a fixed temperature.

References

[1] "Unpublished course reading material, natural solvers," [Online; accessed July-2020].

[2] H. Bayram and R. Şahin, "A new simulated annealing approach for the traveling salesman problem," *Mathematical and Computational Applications*, vol. 18, pp. 313–322, 12 2013.

[3] W. Ben-Ameur, "Computing the initial temperature of simulated annealing," *Computational Optimization and Applications*, vol. 29, pp. 369–385, 12 2004.

[4] D. Weyland, "Simulated annealing, its parameter settings and the longest common subsequence problem," *GECCO'08: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation 2008*, pp. 803–810, 01 2008.

[5] S. M. Ross, *Simulation*. Knovel Library., Amsterdam: Academic Press, fifth edition. ed., 2013.