UNIVERSITY OF AMSTERDAM

Report

Solving Traveling Salesman Problem using Simulated Annealing

December 20, 2024

Student 1:
First Name: Francesco
Last Name: Tiepolo
Student Number: 13319817
Email: francesco.tiepolo@student.uva.nl
Student 2:
First Name: Andrew
Last Name: Crossley
Student Number: 15631265
Email: andrew.crossley@student.uva.nl
Student 3:
First Name: Michelangelo
Last Name: Grasso
Student Number: 15840417
Email: michelangelo.grasso@student.uva.nl

Lecturers:
Andrea Tabi, Giulia Pederzani

Course:
Stochastic Simulation

Course code:
5284STSI6Y

Abstract

The paper gives a basis for the understanding of Traveling Salesman Problems (TSP) and how to solve such problems through simulated annealing. Subsequently, the efficiency of said method is tested on TSPs of different sizes. In particular, an analysis of the effect of changing the number of nodes involved, the length of Markov Chain used, and cooling rates chosen on rate of convergence of the problem to the optimal solution is undertaken through visual representations of simulations. Furthermore, the temperature evolution as cooling rates are changed is also inspected, with the purpose of gaining insights on the relationship between these parameters.

Contents

1 Introduction

Combinatorial optimization entails a large class of mathematical programming problems of discrete nature. Problems of this type are popular in scientific literature because of their many practical applications. However, many of these problems are NP-hard, making it unlikely for an algorithm solving such problems in polynomial time to exist. Nevertheless, continuous research towards increasingly efficient approximate solution methods is made. These tend to take advantage of special structure characteristics of the particular problem being solved, as all combinatorial problems possess a common structure as large multivariate discrete systems with many degrees of freedom [1].

An interesting method for solving combinatorial optimization problems exploiting their structure characteristics is simulated annealing. In particular, the report focuses on the application of simulated annealing on the famous Traveling Salesman Problem (TSP), which is part of the NP-hard class mentioned previously. The paper delves into the assumptions and formulation of TSPs and simulated annealing, as well as assessing its performance on problems of different sizes (based on the number of cities considered). We start by presenting the problem at hand and the algorithm applied to find its solution; we then show the methodology used behind our experiments and explain the results we observe by manipulating the parameters.

2 Theory and Methodology

2.1 Traveling Salesman Problems

The Traveling Salesman Problem (TSP) is a classic combinatorial optimization problem. Given a set of cities and their respective intercity distances, the goal is to determine the shortest closed path that passes through all cities for the salesman to take. The path must start and end in the same city, and no city can be visited more than once. Furthermore, the problem is symmetric, meaning the path from A to B has the same length as that from B to A. TSP's are famously nondeterministic polynomial complete (NP complete), which are problems that cannot be solved exactly in polynomial time [2].

However, multiple heuristic techniques have been developed to solve TSP's. Notable examples are the greedy method, simulated annealing, and tabu search. The difficulty lies in the high computational cost of such techniques when applied to TSP's with many cities [3]. In particular, this paper focuses on simulated annealing in the context of TSP's.

2.2 Assumptions

Generally, TSP is formulated as follows [4]:

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}$$

subject to:

$$\sum_{i=1}^{n} X_{ij} = 1, \quad j = 1, 2, \ldots, n, (1)$$

$$\sum_{j=1}^{n} X_{ij} = 1, \quad i = 1, 2, \ldots, n, (2)$$

$$U_i - U_j + nX_{ij} \leq n-1, \quad i,j = 2,\ldots,n,\ i \neq j, (3)$$

$$X_{ij} \in \{0,1\}, \quad \forall i,j. (4)$$

In this framework, X_{ij} and U_i with $i,j = 1,2,\ldots,n$ that minimize z are to be determined. X_{ij} represents the path from city i to city j, and is binary, meaning if the path goes directly from node i to node j it is equal to 1, otherwise it is 0. C_{ij} represents the distance of the path between city i and city j out of n given cities. U_i/U_j are auxiliary variables, which are used to eliminate sub-tours in the solution. Sub-tours are short paths since they do not pass through all nodes. Constraint (3) prevents the solution from being multi-disjointed cycles, ensuring the assumption that all cities must be visited on the optimum path. On the other hand, constraints (1) and (2) ensure, respectively, that every node is entered and exited exactly once in the tour [5].

In addition to the basic configuration of TSP's given above, the greedy heuristic 2-opt is added. Greedy heuristics select at each iteration the feasible move leading to the greatest improvement in objective function value. It involves removing a pair of edges in the tour in each iteration and reconnecting their endpoints in the only other way. This step is repeated until the length of the tour stops decreasing [6].

2.3 Solving through Simulated Annealing

All combinatorial problems can be said to have a common overarching structure. They are large multivariate discrete systems with many degrees of freedom [1]. A particularly efficient method in finding global minima and maxima in a system can be found by taking inspiration from nature by observing how crystals form. The crystal structure represents the global energy minimum of the material lattice, and the formation method itself, consisting of slow and steady cooling called annealing, is applicable to combinatorial problems.

In seeking the global minimum of a function $h(X)$ with X being a random variable, a suitable density function for the steady approximation towards the global minimum as the system slowly cools is necessary. The best candidate for this purpose is the Boltzmann distribution, as it yields the probability that the system is in a certain state as a function of the energy (E) associated with the state and the temperature (T) of the overall system. Explicitly, it follows $P(E) = \frac{e^{\frac{-E}{kT}}}{Z(T)}$. If rewritten to minimize E using T as a control variable, it becomes $p_j \approx e^{\frac{-h(X_i)}{T}}$, which has two useful properties. As $T \to \infty$, $p(X) \to$ uniform density, while as $T \to 0$, $p(X) \to \pi(X)$, which is the Dirac-Delta around the minimum point. It is a stationary density with a point mass only at the global minimum.

The process begins with a high T value, as the initial sample can be anywhere, which is often a close to zero probability region. Starting from a uniform distribution (high T value), the Markov Chain explores without then risking of getting stuck in a local extrema. As T is decreased, the more extreme peaks will be emphasized until only the global extrema remains as T goes to zero.

The sampling procedure encompasses the following steps: (1) a symmetric proposal density $Q(X_{i+1}|X_i)$ is chosen to pick randomly which neighboring point to visit next; (2) the next possible state $X_{i+1} = q(X|x_i)$ is sampled; (3) U is sampled from a uniform distribution with range $[0,1]$; (4) $\alpha(x_{i+1}|x_i)$ is computed as $min(e^{\frac{-(h(x_{i+1})-h(x_i)}{T}}, 1)$; (5) accept state x_{i+1} if $U \leq \alpha(x_{i+1}|x_i)$; (6) decrease T.

For the TSP in particular, a decrease in the tour length is always accepted and the method is characterized by an overall slow descent. The procedure is applied to multiple problems of varying size, and their solutions are compared with the optimal solutions [1].

Given a high starting T value, a maximum number of iterations n, and a cooling rate leading to the sequence of temperatures obeying $T_1 > T_2 > ... > T_n$, the algorithm goes as follows. (1) Generate a random tour and set $i = 1$. (2) Loop through a Markov Chain where one step of the 2-OPT algorithm is executed at each iteration and the change in the tour length Δd is evaluated. (3) Generate a random variable $U \sim U(0, 1)$. If the new tour length is lower than the previous one or the generated U is smaller than the minimum between 1 and $e^{\frac{\Delta d}{T}}$, accept the 2-OPT exchange and update the values of the tour length and the optimal tour path. (4) After having looped through the Markov Chain, update the T value according to the cooling rate chosen, and set i as i + 1, repeating the process from step 2 unless $i > n$ (maximum number of iterations chosen) [1].

2.4 Methodology

To study the issue presented above, we program a simulated annealing algorithm and apply it to some city configurations which vary in size. Firstly, to build and test our code we focused on a 51-node problem; once we reached a close enough solution to this data set, we studied a larger problem including 280 cities. For additional robustness, we also briefly present the observations drawn from the application of our program to a 442-node specification. Throughout, the main analysis of the mechanisms involved in solving the TSP through simulated annealing is focused on the problem involving 280 cities, as a good compromise between complexity and necessary computational power.

Approaching any of the problems just cited, we start by computing a $n \times n$ distance matrix D of which entries $d_{i,j}$ represent the distances (assumed to be straight lines) between city i and city j (where n is the total number of nodes through which the salesman has to travel); thus, to calculate the length of a specific tour, we refer to D, summing its entries, in accordance with the order of the cities in the tour. For instance, if we take a sample of 4 cities and generate the tour [3, 2, 4, 1], its length will be given by $d_{3,2} + d_{2,4} + d_{4,1} + d_{1,3}$ (given that the salesman must return to the initial city).

Once D has been computed, we start iterating through the simulated annealing process: we begin with a randomly ordered tour of all the cities, calculate its total length and proceed to engage in the first Markov Chain. For each of these, at each step of the chain, we take the shortest-so-far tour and apply a 2-opt swap by deleting 2 non-adjacent edges and reconnecting the four cities as to generate a new circuit; the obtained tour is then accepted/rejected following the terms described above. Once the algorithm iterates through the Marvok Chain length desired, the temperature is lowered according to the set cooling rate, and a new iteration is performed starting from the shortest-so-far tour found. The program is stopped either when the temperature T satisfies $T < 1 \times 10^{-12}$ or when a set maximum number of iterations is reached.

We focus our study on the convergence of our algorithm (simulated annealing) to the true solution of each of the three problems described above, in the form of an ordered list (tour) representing the shortest possible option. This is mainly done by looking at the total distance of the shortest-so-far tour at each iteration and computing the difference with the total length of the true-optimal tour (which for the data sets used here is known). Every conclusion we draw is based on the mean results of 30 simulations run for every parameter combination we study; the graphs also report the 95 % confidence intervals around these means. Our interest will be mainly devoted to the role of cooling schedules and Markov Chain's length on this convergence. In the next section, we turn to the analysis of the results regarding these topics.

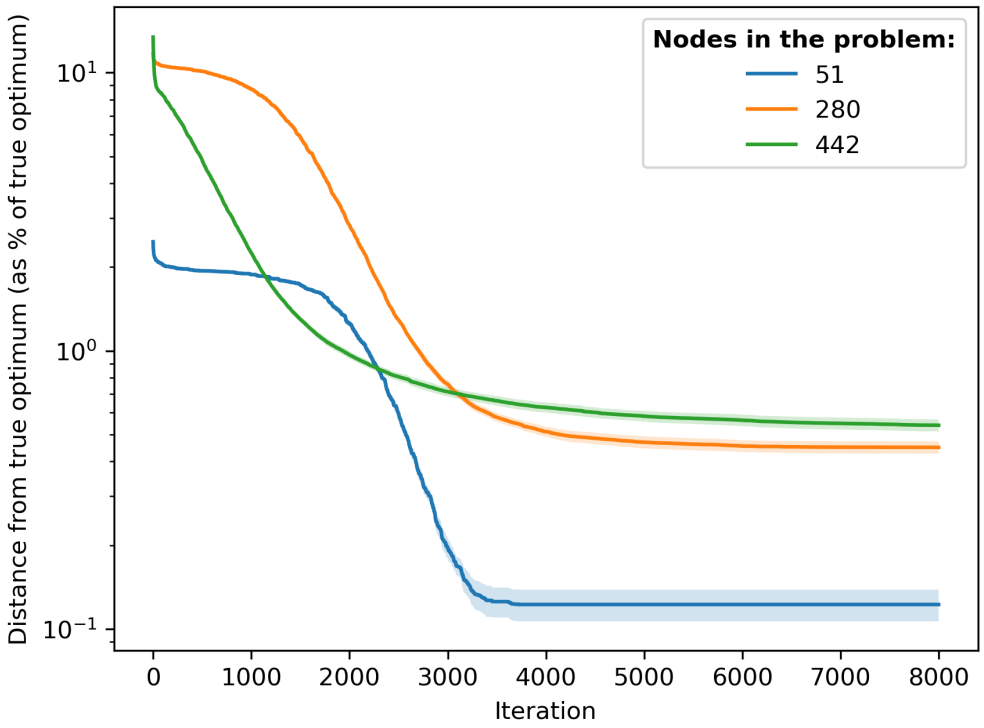


Figure 1: **Convergence of the simulated annealing algorithm for the Traveling Salesman Problem (TSP) with 51, 280, and 442 nodes**. The y-axis shows the distance from the true optimum as a percentage, while the x-axis shows the iteration count. The shaded regions around the curves represent the 95% confidence intervals, indicating variability in the solutions across multiple runs of the algorithm. Parameters used: $T_0 = 1000$, cooling rate = 0.998, MC length = 100, and max iterations = 8000.

3 Results and Discussion

We focus our study on the convergence of our algorithm (simulated annealing) to the true solution of each of the three problems described above, in the form of an ordered list (tour) representing the shortest possible option. This is mainly done by looking at the total distance of the shortest-so-far tour at each iteration and computing the difference with the total length of the true-optimal tour (which for the data sets used here is known). Every conclusion we draw is based on the mean results of 30 simulations run for every parameter combination we study; the graphs also report the 95 % confidence intervals around these means. Our interest will be mainly devoted to the role of cooling schedules and Markov chain's length on this convergence. In the next section, we turn to the analysis of the results regarding these topics.

Our initial runs of the Simulated Annealing algorithms on the different node sizes demonstrated the expected behavior of the methods efficiency and relative error given the size of the problem set introduced. Figure 1 displays the convergence of the algorithm towards the true optimum by graphing the relative distance from the true optimum on the y-axis and iteration count on the x-axis for each problem size. It is important to note that the convergence rate is purely in terms of iterations, neither time nor computational intensity are studied.

The results of all three trials show a relatively quick ($< 3000 iteration$) drop-off of the estimated optimal distance toward the real optimal result. After 4000 iterations, the trajectory of all the problem sizes plateaus, with the 51-node problem finding the result closest to the true optimum and the other two being similarly

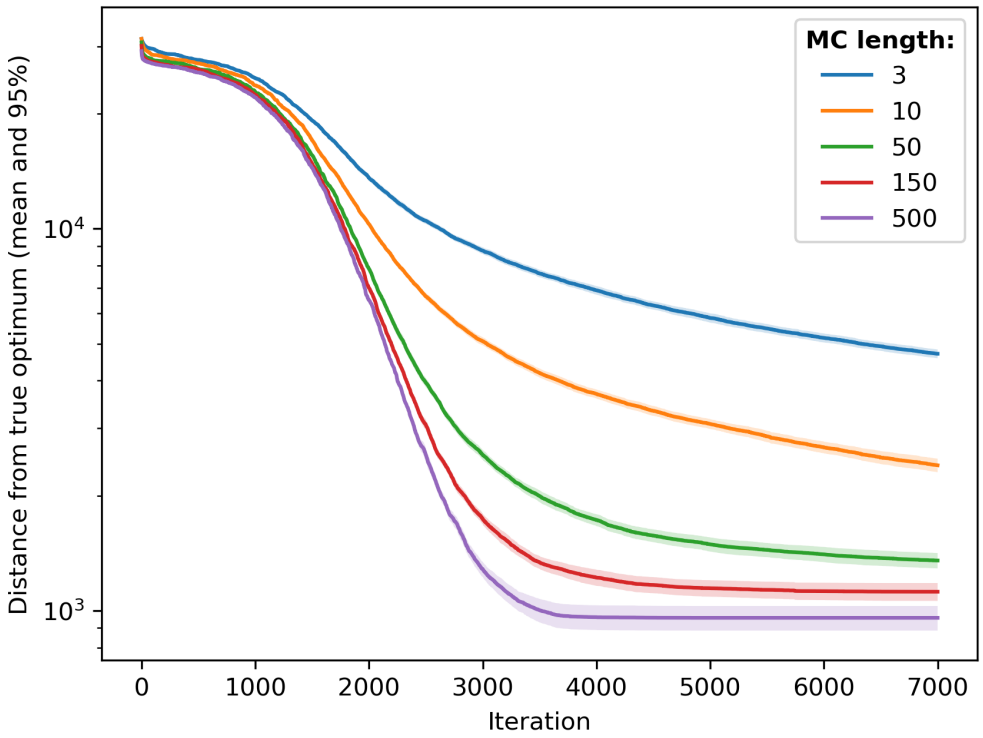


Figure 2: **Effect of Markov chain (MC) length on the performance of the simulated annealing algorithm for the Traveling Salesman Problem (TSP) with 280 nodes**. The y-axis shows the distance from the true optimum on a logarithmic scale, while the x-axis shows the iteration count. The algorithm was initialized with a starting temperature of $T_0 = 1000$, a cooling rate of 0.998 and a maximum of 7000 iterations. Results are shown for different MC lengths: 3, 10, 50, 150, and 500. The shaded regions represent the 95% confidence intervals over multiple runs. Longer MC lengths demonstrate improved convergence to the true optimum due to more extensive exploration at each temperature.

grouped but relatively farther away.

Before moving to the results regarding the Markov chain length and the cooling rate, we highlight that all the following experiments were executed on solutions of the 280-node problem, the general behavior of which is demonstrated relative to other problem sizes in Figure 1.

The Markov chain length parameter refers to the number of transitions that occur at the same temperature within the algorithm calculation cycle. Put simply, the parameter changes the amount of "exploration" the algorithm does before choosing an optimal value and lowering the temperature ("cooling"). As demonstrated in Figure 2, the observations for the MC length trials show lower MC lengths resulting in less optimal approximated solutions and an earlier result plateau. Less exploration of the solution space consistently leads to worse results. The cooling effect also appears to be marginally lower between the $mc = 150$ and $mc = 500$ runs compared to the $mc = 3$ and $mc = 50$, implying diminishing returns for longer MC lengths. The results show a higher propensity for getting "stuck" in local minima for annealing processes simulated with a shorter MC length.

Isolating the cooling effect had much less immediately legible results, presented in Figure 3. Lower cooling rates (≤ 0.95) led to much faster slopes of descent toward optimum, but tapered off after significantly fewer iterations. The low cooling-rate trials ended at suboptimal local results and despite requiring fewer iterations to reach a solution, were far less precise (farther from true optimum). Each increase in cooling rate decreased

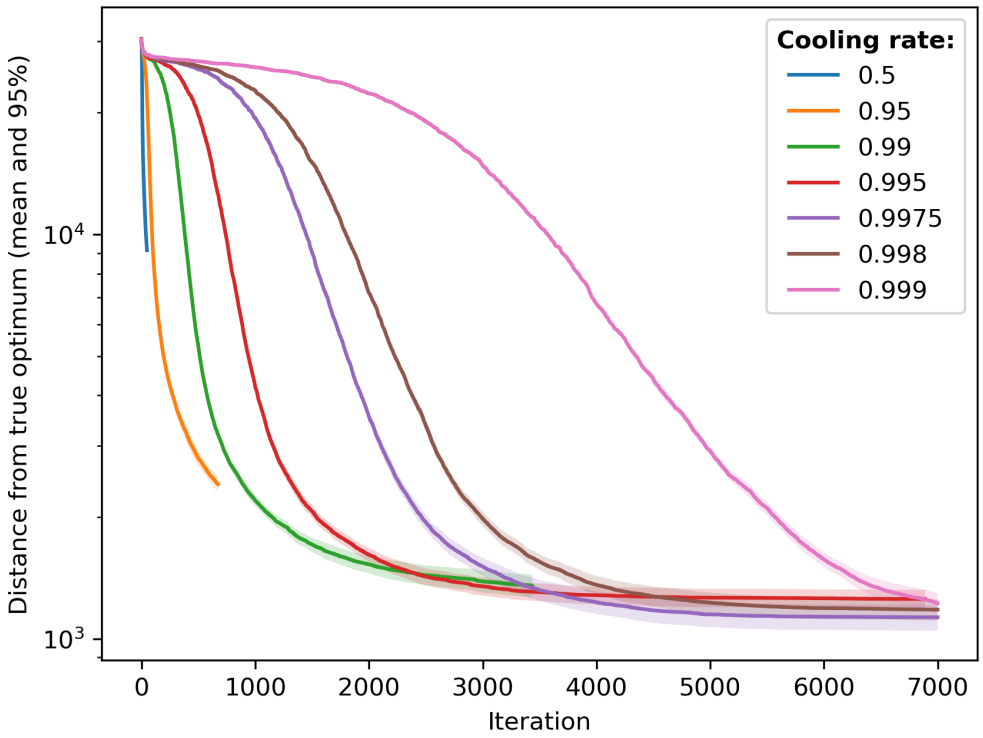


Figure 3: **Effect of cooling rates on the performance of the simulated annealing algorithm for the Traveling Salesman Problem (TSP) with 280 nodes**. The y-axis shows the distance from the true optimum on a logarithmic scale (mean and 95% confidence intervals), while the x-axis represents the iteration count. The algorithm was run with a starting temperature of $T_0 = 1000$, a Markov chain length of 100, and a maximum of 7000 iterations.

the rate of convergence, as best evidenced by a comparison of the slopes of simulations for cooling rates 0.999 and 0.998. Given the total maximum iteration count (7000), the cooling rate that led to the best approximation was 0.9975. However, observation of the rate of descent of the 0.999 cooling rate even at 6000+ iterations, suggests that the accuracy of the results may have improved if the trials had been run for longer.

The decreases in temperature during these simulations are shown in Figure 4. The chart shows the log-scaled temperature of each cooling rate. The graph makes most obvious the effective difference explored initially in Figure 3, with the decrease in temperature occurring slower - across more iterations - the higher the cooling rate.

The figure makes it clear that while higher cooling rates allow for better exploration of the solution space, they require more iterations to reduce the temperature significantly, resulting in slower convergence. On the other hand, lower cooling rates achieve a rapid reduction in temperature, making optimization faster, but potentially less useful for more complex optimization landscapes. In the latter case, the temperatures reach the cutoff value of 10^{-12} earlier, given the faster cooling schedule, which stops the algorithm after fewer iterations (as seen by the broken lines in Figure 3). Thus, a lower cooling rate increases convergence speed, but prevents the algorithm from running long enough to find a better solution.

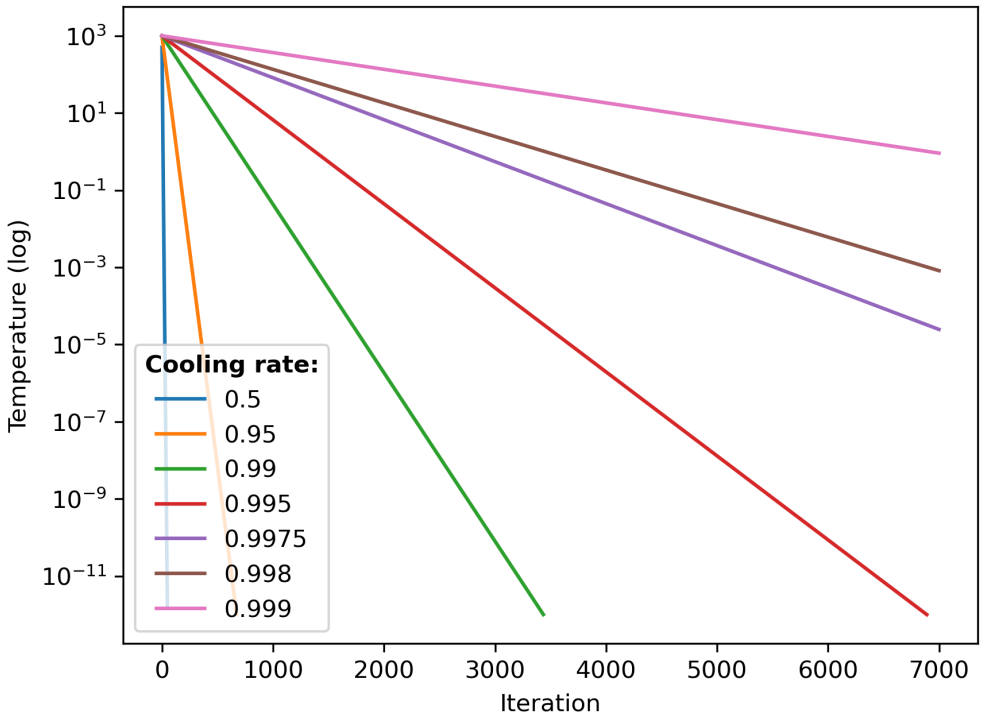


Figure 4: **Temperature evolution during simulated annealing optimization for various cooling rates.** The y-axis shows the temperature on a logarithmic scale, while the x-axis represents the iteration count. Different colored lines correspond to different cooling rate values, ranging from 0.5 to 0.999, which control the rate of temperature decrease during the optimization process. The starting temperature T_0 was set to 1000, and the maximum number of iterations was 7000.

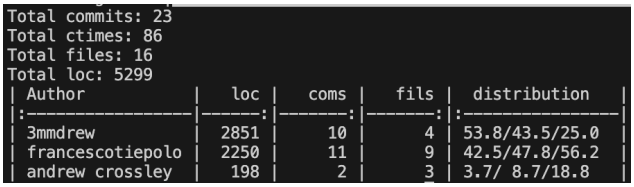
4 Conclusion

With the present report, we intended to study the application of the simulated annealing algorithm to solving the traveling salesman problem. By applying the method to different cities configurations and running our program multiple times, we obtained robust results regarding the subject. Firstly, larger problems suffer from higher computational requirements in order to converge to a better solution, achieving better results after a smaller number of iterations; secondly, the Markov Chain length significantly impacted convergence, increasing performance as this parameter is increased; lastly, a faster schooling schedule results in faster convergence, but at the same time presenting a final solution further away from the true optimum. In conclusion, the parameters studied here should be carefully chosen in accordance with the problem at hand in order to best fit convergence speed, accuracy in the solutions and available computational power.

5 Task Distribution

The work for the assignment was divided up as follows: (1) Michelangelo: Abstract, Introduction and Theory; (2) Francesco: code for simulated annealing and part of parameters study, Methodology and Conclusion sections of the report; (3) Andrew: Discussion of results, part of code for parameters study, general git management.

We show the Git Fame report in Figure 5. All code and documentation can be found in the following public repository: `https://github.com/3mmdrew/stochastic_assignment_3.git`.



```
Total commits: 23
Total ctimes: 86
Total files: 16
Total loc: 5299
| Author           |   loc |   coms |   fils |  distribution   |
|:-----------------|------:|-------:|-------:|:----------------|
| 3mmdrew          |  2851 |     10 |      4 | 53.8/43.5/25.0  |
| francescotiepolo |  2250 |     11 |      9 | 42.5/47.8/56.2  |
| andrew crossley  |   198 |      2 |      3 | 3.7/ 8.7/18.8   |
```

Figure 5: **Git Fame.** Screenshot of Git Fame with division of git workload. The usernames *3mmdrew* and *andrew crossley* are both referring to Andrew Crossley.

References

[1] C. C. Skiścim and B. L. Golden, "Optimization by simulated annealing: A preliminary computational study for the tsp," in *Proceedings of the 15th Conference on Winter Simulation - Volume 2*, ser. WSC '83. IEEE Press, 1983, p. 523–535.

[2] P. Bhattacharyya and A. Chatterjee, "Geometric properties of optimal and most probable paths on randomly disordered lattices," in *Statistics of Linear Polymers in Disordered Media*, B. K. Chakrabarti, Ed. Amsterdam: Elsevier Science, 2005, pp. 271–303.

[3] C. D. . S. Sangwan, "Literature review on travelling salesman problem," *International Journal of Research*, vol. 5, 2018.

[4] T. A. Miller, C.E. and Zemlin, "Integer programming formulation of travelling salesman problems," *J. ACM*, vol. 7, pp. 326–329, 1960.

[5] S. C. Gavish, Bezalel; Graves, "The travelling salesman problem and related problems," *Operations Research Center Working Paper;OR 078-78*, 1978.

[6] P. Hansen and N. Mladenović, "First vs. best improvement: An empirical study," *Discrete Applied Mathematics*, vol. 154, no. 5, pp. 802–817, 2006.