University of Amsterdam

Assignment 3

Simulated annealing: solving the TSP and the Thompson problem

December 14, 2019

Students:
Marcus van Bergen
10871993

Steven Raaijmakers
10804242

Tutor:
Marleen Rijksen

Group:
20

Lecturer:
A. Hoekstra

Course:
Stochastic Simulation

1 Introduction

Nowadays, a wide range of problems can be solve via algorithms. With advances in modern technology, it is very possible to take such problems and formulate them in such a way that they are solvable by one of these algorithms; like a probabilistic problem solver.

In this report we apply simulated annealing, a probabilistic problem solver, to two different problems: the Traveling Salesperson problem and the Thompson problem. This paper starts by giving background information on both problems and what we are aiming to solve. Then, we give background information regarding stimulated annealing and its operation. The subsequent section elaborates on how we solve both problems using simulated annealing. Afterwards, in the methods section, we explain what experiments we will perform and how we will perform them. The corresponding results are presented in the proceeding section. Finally, our report is completed with a discussion and conclusion where we elaborate on our results, limitations of our experiments and possibilities for future research.

2 Background Material

2.1 TSP

The traveling salesperson problem (TSP) was first formulated in 1930 and consists of the following: given a set of cities; what is the shortest route that visits every city at least once and returns at the origin city. The TSP is classified as an NP-hard problem.

2.2 Thompson Problem

The Thompson Problem, describes the problem of finding the minimal potential energy configuration of k particles constrained to the surface of a sphere or circle. Our goal for this research is to use Simulated Annealing to find the minimal potential energy configuration, E, for a given number of k particles in a circle (two dimensions).

2.3 Simulated Annealing

As previously mentioned simulated annealing (SA) is a probabilistic problem solver. The algorithm aims to find the global maximum or minimum of a given cost function.

Applying it to the Thompson problem, the algorithm takes after the process of heating and cooling crystals. By doing so the crystals, after they are cooled, lose their structural defects yielding a desirable result [2].

More generally speaking, when applying SA we start with an initial configuration and a starting temperature T_0. The temperature is decreased according to the cooling schedule whenever a Markov chain of length l has been created. In each Markov chain we start with a current configuration i which will be perturbed via some operator, creating candidate j. Assuming that we try to find a global minimum, candidate j will be accepted as the next state X_i in the Markov chain with the following probability:

$$\mathrm{P}_T(X_i = j) = \begin{cases} E_j \leq E_i \rightarrow 1 \\ E_i < E_j \rightarrow e^{\frac{-|E_i - E_j|}{T}} \end{cases} , \tag{1}$$

with $E(i)$ denoting the energy (or cost) of i and T the current temperature. If j does not get accepted we set $X_i = i$.

This process repeats itself until a Markov chain of length l has been created. The temperature is than decreased and the last state of the Markov chain is used to create a new Markov chain using the same principle.

n of such Markov chains are created until the SA terminates. In our experiments we denote the number of Markov chains that have to be created before termination by n_{max}.

2.3.1 Cooling schedules

The temperature and its decrease are important characteristics of SA since it determines the acceptance probability as described in equation 2.3. The most common way is to decrease the temperature is by multiplication with factor α, resulting in an exponential decrease. The corresponding formula can be described as follows:

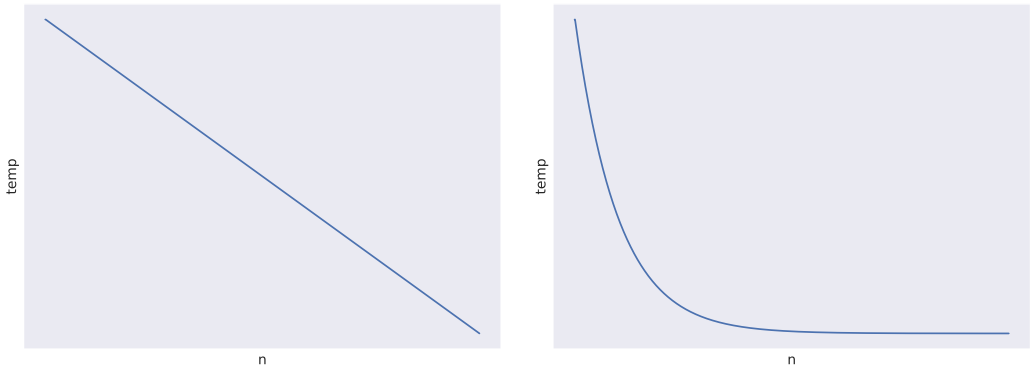
$$\mathrm{T}_{n+1} = \mathrm{T}_n \cdot \alpha, \tag{2}$$

where T_n is the temperature for Markov chain n.

Another common way to decrease the temperature is in a linear fashion, where the temperature is subtracted with value B every step n:

$$\mathrm{T}_{n+1} = \mathrm{T}_n - \mathrm{B} \tag{3}$$

Figure 1 show an example of the generated curves for the temperature as a function of n.



(a) Linear cooling schedule

(b) Exponential cooling schedule

Figure 1: Decrease of temperature per step, $\alpha = 0.99$.

3 Methods

3.1 TSP

We have access to a configuration of cities consisting of 280 unique cities of which we also possess the optimal route. This optimal route is known to be the route with the lowest cost, which in this case means it has the shortest path length while visiting every city and ending at the origin city. The path length is derived using the euclidean distance between the cities, which is symmetric, meaning that $E(A, B) = E(B, A)$.

We will use the 2-opt transition as an edit-type for SA, i.e. to make perturbations between different states of the Markov chain. Using this operator we pick two random cities in a configuration of which the order of the cities between them will be swapped. The initial configuration is a random configuration of the cities. For different experiments we keep the initial configuration constant which ensures fair comparison.

3.2 Thompson Problem

In this subsection we will elaborate on our approach to solving the Thompson Problem using SA. In our case we will constrain our Thompson problem to the surface of a circle. In particular we will be constraining the particles to be located within a unit circle. Our model also works under the assumption that all particles have the same charge.

The SA algorithm will try to approximate the configuration of particles where the energy configuration is lowest. According to [1] given a list of particles we can calculate the total energy configuration of the particles as follows:

$$E = \frac{1}{2} \sum_{i=1}^{k} \sum_{\substack{j \neq i \\ j=1}}^{k} |\bar{r}_i - \bar{r}_j|^{-m} . \tag{4}$$

In the context of SA, equation 4 is the cost function we are aiming to minimize. Following [3], which also uses equation 4 we have chosen $m = 2$, which is the euclidean norm. Furthermore, k signifies the amount of particles in our circle where both $\bar{r}_i, \bar{r}_j$ represent the coordinates of particles i, j as a vector.

SA tries to minimize the cost of the particles by using Markov Chains. At each iteration of an arbitrary Markov Chain, our SA algorithm will generate a new candidate solution, which might be accepted. In the case of the Thompson problem this will be a new configuration of the k particles in the circle. We generate candidate solutions in two ways, these methods are called the edit-types:

- Perturb particles randomly. For each particle both the x, y component are perturbed in either a positive or negative direction. The direction to perturb is chosen with a probability of 0.5, thus each perturbation either changes x or y. Because the perturbations are completely random, we have chosen to allow boundary conditions for this method. Meaning if the particles is perpetuated outside of the circle, it will wrap to the other side.
- Perturb particles with force vector. For each particle we can calculate the total repelling force and direction exerted on it by the other particles. The equation to calculate this force vector is given as follows:

$$\vec{F}_{i,j} = \frac{\vec{r}_{i,j}}{|r_{i,j}|^3}, \tag{5}$$

  where $r_{i,j}$ is the distance between two particles. Meaning, if we have a particle i, we have $k - 1$ force vectors calculated using equation 5, where each vector is the force exerted on i by each other particle. We sum these vectors to get the total force and direction as a vector of the repelling forces exerted on the particle. Thereafter we perturb i with a scale, ϵ, of this total vector. This same process is then applied to all other k particles. This moves the particles in the direction to which they are being repelled. We do not support boundary conditions for this edit-type.

3.3 Cooling schedule

To investigate the effect of the cooling schedule on the convergence of our solutions we run simulations for both cooling schedules described in section 2.3.1. In order for fair comparison we use the α from the exponential cooling schedule to derive a value for B in the linear cooling schedule via:

$$B = \frac{T_0 - T_0 \cdot \alpha^n}{n}, \tag{6}$$

with T_0 being the initial temperature. This derivation ensures that simulations for both cooling schedules have the same end temperature for equal α, T_0 and n.

3.4 Markov chain length

We are also interested in the effect of the length of the Markov chain l on the convergence of our simulations to the optimal solution. We therefore choose to vary l, where we keep α, T_0, and n constant.

4 Results

4.1 TSP

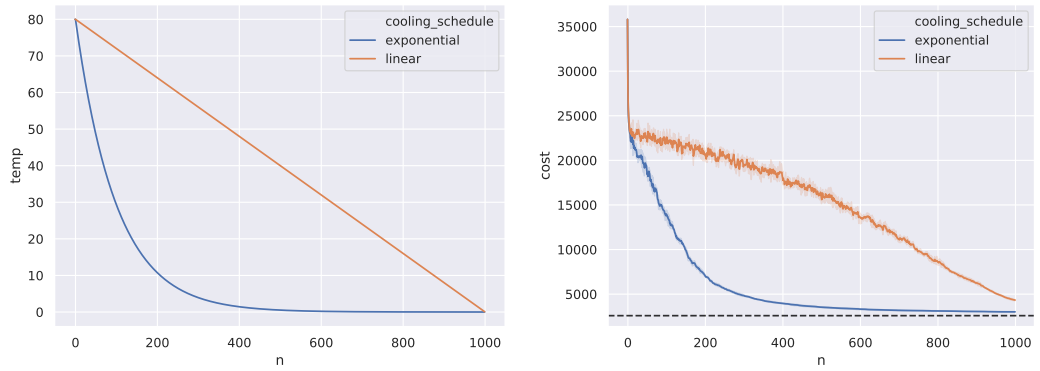
For the TSP, our experiments have been applied to the configuration of 280 cities, spanning an area with a surface of 280 by 160. The corresponding optimal route for this solution has a cost of 2586.77 via the euclidean distance. Our initial (randomly generated) configuration has a cost of 35813.67.

In the plots we visualize over n, showing the corresponding value of the last state in Markov chain n. Since we repeat every experiment 5 times we take the mean and the standard deviation of these values. Note that the standard deviation pales in comparison to the scale of the y-axis.

4.1.1 Cooling schedule

For the cooling schedule experiments we have varied the cooling schedule between exponential and linear, for $n_{max} = 1000$, $l = 300$, $\alpha = 0.99$, $T_0 = 80$ with 5 repetitions.

Figure 2a illustrates how the temperature decreases as a function of n, ensuring that in both experiments T_0 and T_n are equal same. In Figure 2b we see the corresponding cost changing over n. Both plots show us that the cost seems to follow the temperature function, meaning that the exponential cooling schedule converges quicker. Also, the plots show us that the exponential cooling schedule converges to a lower minimum, we therefore use the exponential cooling schedule for subsequent experiments.



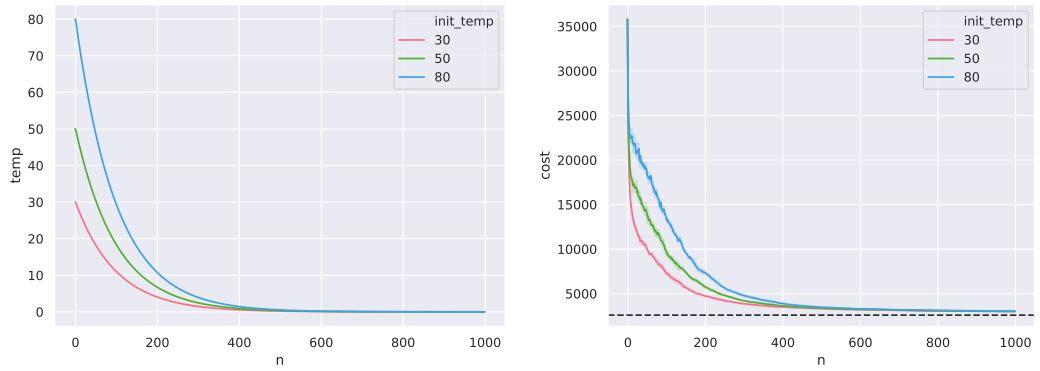
(a) Temperature over n

(b) Cost over n

Figure 2: Exponential versus linear cooling schedule for $n_{max} = 1000$, $l = 300$, $T_0 = 80$ and $\alpha = 0.99$.

Temperature

As the temperature is part of the cooling schedule we run the same experiment with different $T_0 = \{30, 50, 80\}$ for equal α, l and n_{max}, as shown in Figure 3. These plot strengthens the idea that the cost is strongly correlated to the temperature. This would mean that a low temperature would be a better option since this seems to converge quicker. However, we know from theory that lower T_0 reduces the overall probability of jumping out of local minimums in comparison to higher temperatures. We therefore choose to keep $T_0 = 80$.



(a) Temperature over n

(b) Cost over n. Dashed line denotes the cost of the optimal solution.

Figure 3: SA for different T_0 with $n_{max} = 1000$, $l = 300$, $\alpha = 0.99$ using an exponential cooling schedule.

4.1.2 Markov chain length

Figure 4 shows the convergence of our simulations to the optimal solution of the 280 cities configuration, for Markov chain lengths $l = \{100, 200, 300, 400, 500\}$ with $T_0 = 80$, $N_{max} = 1000$ and $\alpha = 0.99$. We see that for larger l the final solution has a lower cost, which is also shown in Table 1. The table shows us that the standard deviation is relatively small and thus is hard to visualize in the corresponding figure.

l	Cost mean	Cost std.
100	3613.00	84.60
200	3149.51	28.14
300	3002.77	66.75
400	2958.72	80.61
500	2844.63	28.89

Table 1: The mean cost and its standard deviation of the last state in Markov chain n_{max} of 5 repetitions for different Markov chain length l.

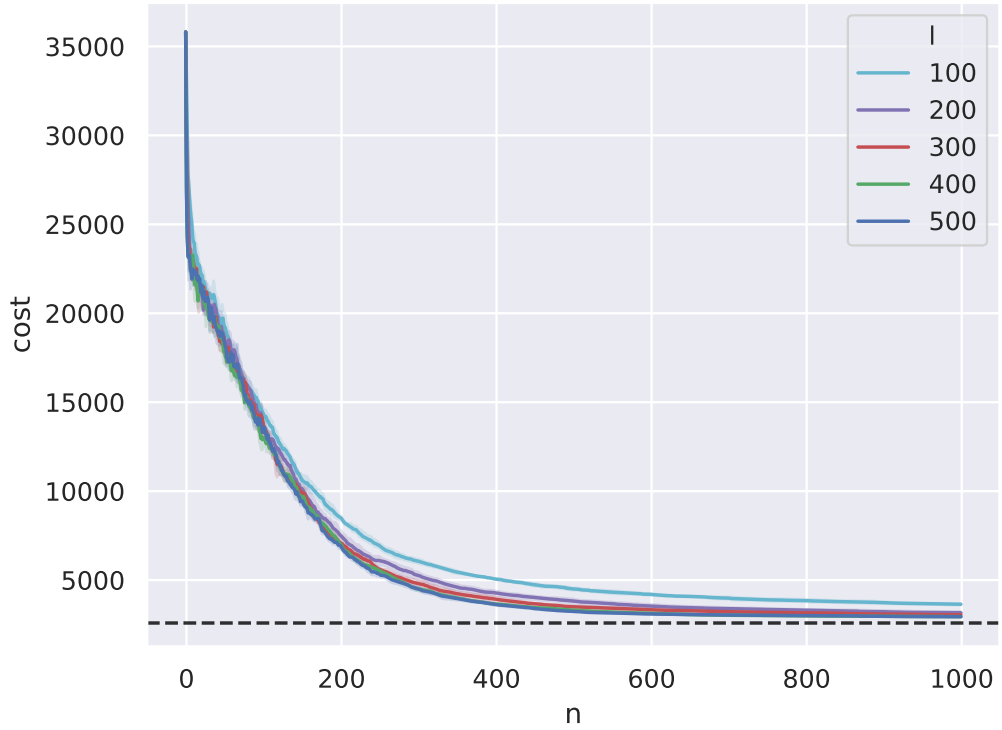


Figure 4: Mean cost of last state for every Markov chain n of length l, with 5 repetitions. The cost of the optimal solution is denoted by the dashed line.

4.2 Thompson Problem

For the Thompson problem we have several experiments which we are going to execute. First we will start by discussing the behavior of the different edit-types (perturb particles randomly and perturb particles with force vector). Afterwards we will demonstrate the effect of different cooling schedules, namely linear and exponential cooling. Afterwards, we will show what the effect is of different Markov Chain lengths and if different lengths are relevant to our problem. For all experiments the initial starting configurations, for different values k, are the same.

4.2.1 Edit-types

As discussed in section 3.2, for our Thompson problem we will be experimenting with two edit-types. With $k = 6$, meaning we have six particles in the unit circle we see interesting

behavior. For the first edit-type, perturbation particles randomly, as the temperature seems to drop the SA algorithm approaches the optimal solution. For second edit-type, where each particle follows its force vectors, after a very small amount of Markov Chains the algorithm has already found a solution, and does not seem to deviate much from this solution (see figure 6 for visualization). The figure below shows the aforementioned results. Once again, in the plots we visualize over n, showing the corresponding value of the last state in Markov chain n. Since we repeat every experiment five times we take the mean and the standard deviation of these runs.

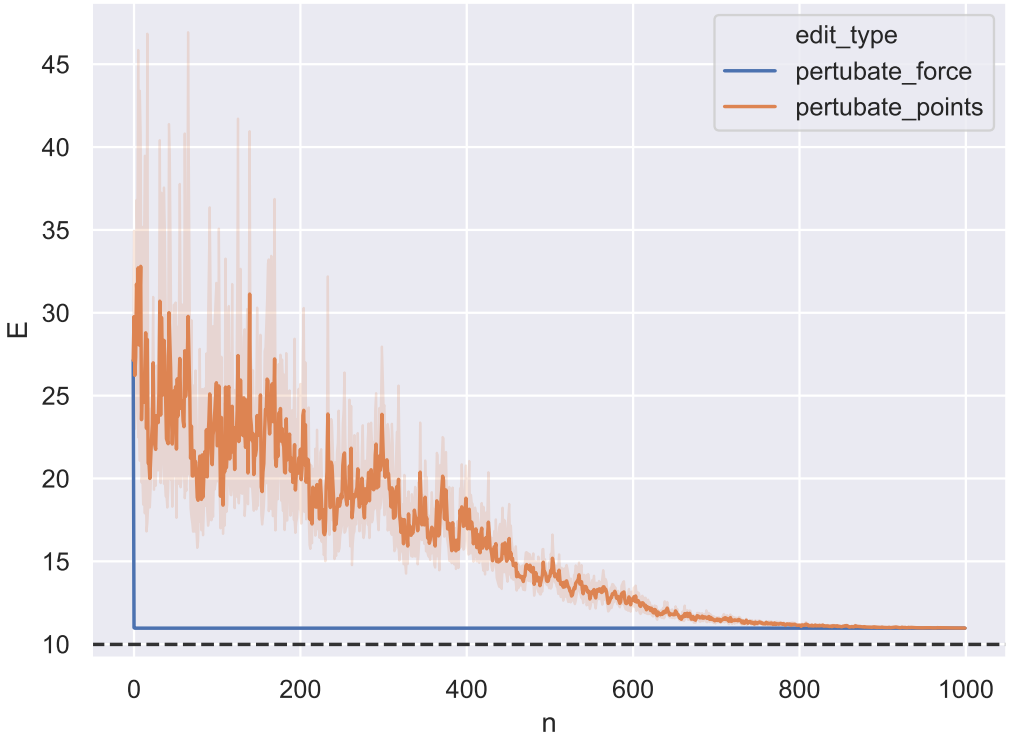
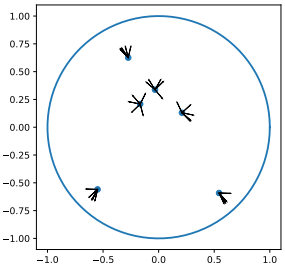
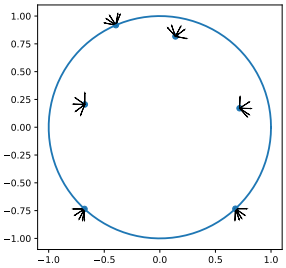


Figure 5: $k = 6, l = 300, n = 1000, T_0 = 80, \alpha = 0.99$ cooling scheme = exponential, reps = 6 Perturb force versus Perturb particles random

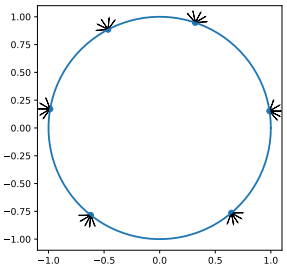
Also below a demonstration of $k = 6$ with edit-type perturb force vector, showing that by stepping along the force vector all particles are pushed to the edge of the circle.



(a) Initial configuration $k = 6$



(b) Best solution after $l = 20$

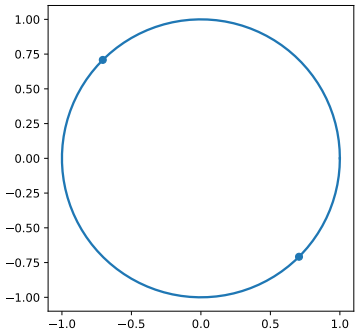


(c) Convergence to stable equilibrium $k = 6$

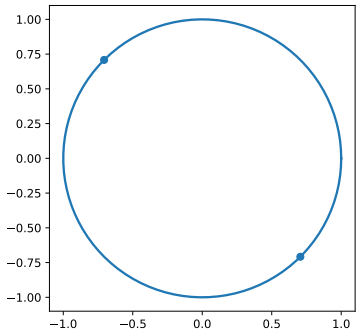
Figure 6: Visualization of $k = 6$ showing repelling motion yielding sub-optimal E state. Edit-type is perturb particles force vector, where the vectors on each particle represent the direction being pushed by surrounding particles. This all happens in one Markov Chain (hence the steep drop seen in figure 5)

We can explain the different behavior between the edit-types. For simplicity assume we are given two particles in a circle, in other words $k = 2$. When we put two particles in a circle randomly with each other, due to their repellent nature, the forces they exert on each-other specify that they should move away from each other in opposite directions. By advecting the particles in the direction of the forces being exerted on said particle, they will end up on opposite ends of the circle. We also take measures to ensure that we do not immediately step to the end of the circle when stepping the force vector. This is done by scaling all force vectors to a unit force-vector and stepping with said unit-force-vector with a small step $\epsilon = 0.01$. The calculation of force vectors is not random. Due to our particles being seeded with the same starting positions every experiment, the outcome for the next simulation will be similar.

In contrary, the edit-type where we perturb the particles with a random factor is able to experiment many different combinations of the particles in the space. In the case where $k = 2$ the SA algorithm, is eventually able to find a solution where both particles are at opposite ends of the circle (in case $k = 2$ this is optimal), albeit at different end-points than that advection would have generated. We have once again shown this below:

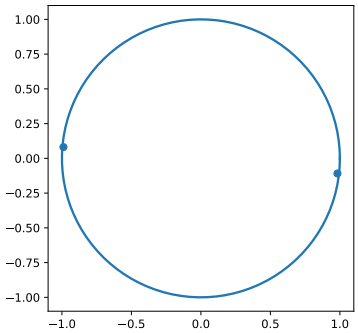


(a) Run one

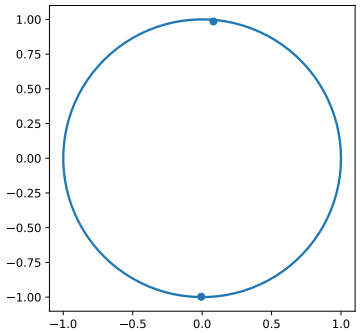


(b) Run two

Figure 7: $k = 2, l = 100, n = 1000, T_0 = 80, \alpha = 0.99$ cooling scheme = exponential. Run with edit-type particle force advection; started with the same initial configuration. Advecting means the particles step in the same direction, which leads them to be positioned approx. the same. This configuration is yielded almost immediately.



(a) Run one



(b) Run two

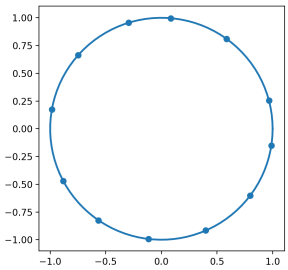
Figure 8: $k = 2, l = 100, n = 1000, T_0 = 80, \alpha = 0.99$, cooling scheme = exponential. Run with edit-type random perturb particles; started with the same initial configuration. Random perturbations leads both particles to face opposite, but at different spectra. This configuration is usually achieved at the end of the SA experiment.

As we can see in figure 7, it is clear that the advection approach seems to miss certain configurations where the E could be minimal. This same effect is observed when trying to discover some of the "magic number" configurations of the Thompson Problem with higher k.

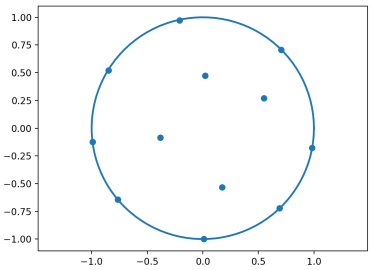
4.3 Magic Behavior of the system at $k = 12$

In section 4.2.1 we have shown how both the edit types behave and why they behave in a certain way. Recall the mechanics of advection and force vectors, many potential configurations are not explored. Also recall that for all experiments with k, in this case $k = 12$, the staring configuration is the same. For the Thompson problem it is known that for certain k, the minimal E configuration can yield interesting results. According to [1], the minimal E

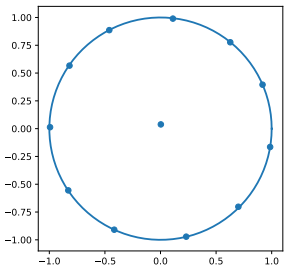
for $k = 12$ leads the particles to align in an icosahedron form (in two dimensions). Using SA we have attempted to find the minimal E for $k = 12$ yielding the following results:



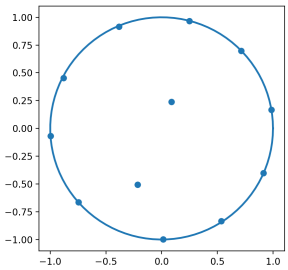
(a) edit-type = perturb particles with force advection. $E \approx 60$



(b) edit-type = perturb particles randomly. Closest approximation to magical icosahedron. $E \approx 58$



(c) edit-type = perturb particles randomly. $E \approx 60$



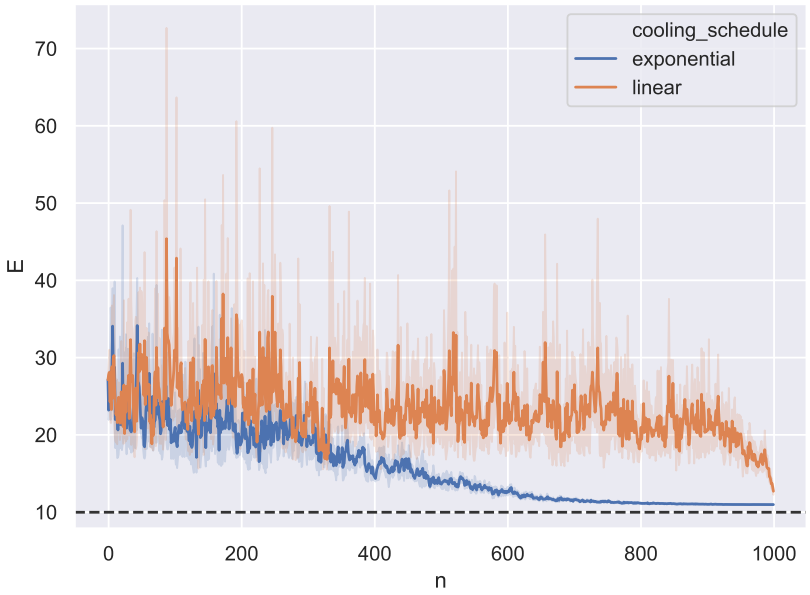
(d) edit-type = perturb particles randomly. $E \approx 61$

Figure 9: Four different runs for $k = 12, l = 100, n = 1000, T_0 = 80, \alpha = 0.99$, cooling scheme = exponential and varying edit-type.

Moreover, in [1], the minimal energy configuration for $k = 12$ is $E \approx 49$. Our SA algorithm approximates this quite closely as seen in figure 9b but is not able to find the exact optimal value. Furthermore, using edit-type perturb particles with force vector always results in all particles being pushed to the edge of the circle. This behavior is expected and demonstrated in figure 6. Due to the edit-type perturb particles force vector missing potential optimal configurations, we have chosen to use random particle perturbation for all consecutive experiments.

4.4 Cooling schedule

Similarly to section 4.1.1, we have experimented with different cooling schedules. We decided to vary the cooling schedule between exponential and linear, for $n_{max} = 1000$, $l = 300$, $\alpha = 0.99$, $T_0 = 80$ with 5 repetitions.



(a) Linear and exponential cooling schemes

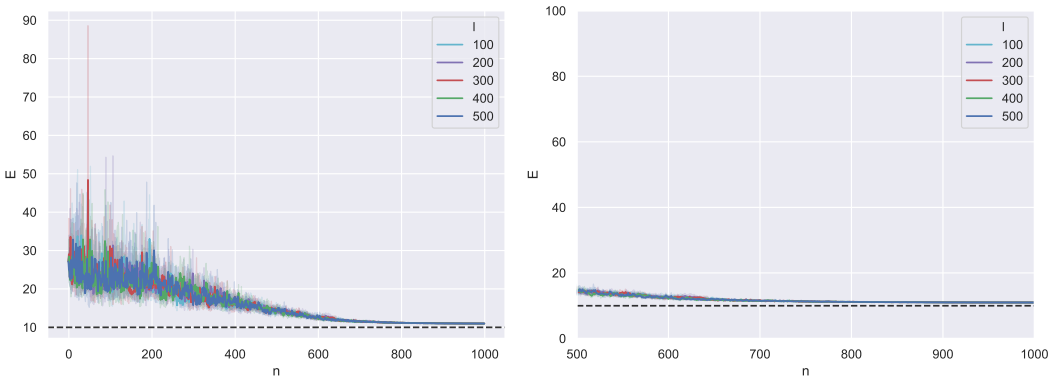
Figure 10: $k = 6, l = 300, n = 1000, T_0 = 80, \alpha = 0.99$, cooling scheme = variant, edit-type = random particle perturbations. The dashed line is the optimal value for $k = 6$ found in [1].

From Figure 10 it is clear that a linear cooling schedule, in general, has a higher variance that that of the exponential. It is also apparent that the linear cooling scheme converges to the optimal solution relatively late compared to the exponential cooling scheme. Moreover, the linear scheme minimized worse compared to the exponential cooling schedule.

Due to its lower variance, steadier convergence and better minimization of E we have chosen to use the exponential cooling scheme for the subsequent experiments.

4.5 Markov Chain Length

We have also taken the liberty to vary the Markov Chain length for our experiments. In figure 11 we observe the convergence of minimal E for $k = 6$ with 5 repetitions.



(a) Energy change over time for different l.

(b) Zoomed in.

Figure 11: $k = 6$, $l = \{100, 200, 300, 400, 500\}$, $n_{max} = 1000$, $T_0 = 80$, $\alpha = 0.99$ and an exponential cooling scheme.

Each Markov chain length follows near identical convergence and approximates the optimum nearly the same. Do recall that the higher the Markov Chain Length, the higher factor of computation ($l = 500$ needs 500% more iteration in comparison to $l = 100$).

5 Conclusion and Discussion

5.1 TSP

Starting with the experiment with different cooling schedules (Figure 2) we have found that an exponential cooling schedule is the preferable cooling schedule. In comparison to the linear cooling schedule it shows a quicker convergence to a global minimum, while also providing a solution with a lower cost.

Thereafter, we ran the experiment for an exponential cooling schedule using different T_0 (Figure 3). This showed us that for lower T_0 the simulation converges quicker to a global minimum. However, higher T_0 increases the overall probability of jumping out of a local minimum which can be an important characteristic of applying the SA. We therefore choose to not use low temperature in our subsequent experiments, but keep $T_0 = 80$ instead. Since we are not sure of the result of this experiment, future work would consist of investigating this property more in depth. Another parameter we could have investigated is α since, as mentioned earlier, the temperature and its decrease determine the acceptance probability of a candidate.

Finally, the experiment using different l shows us that for bigger l the simulation converges earlier while also resulting in a better ending solution. However, the SA needs to create a Markov chain of length l for every n resulting in $n \cdot l$ total iterations, increasing rapidly for large n and increasing l. Therefore, when using SA l must be chosen in such a way that it balances the computational time while the solution must also provide a good approximation of the global minimum. Finally, the values in Table 1 for this experiments illustrate the standard deviation is not always decreasing for higher l, which is can be due to the low number of repetitions we have used.

For future experiments we should use more repetitions, which has not been done for this report due to the (computational) time limits. Using more repetitions also allows for an implementation in a parallel fashion, saving computational time while also giving a more correct representation of our simulation.

5.2 Thompson problem

We started with experimenting different edit-types for the Thompson problem. We found that the edit-type random perturb particle was able to explore more combinations than the force vector edit-type. We found the random perturb edit-type more suitable for our SA experiments. Furthermore, our force vector edit-type does not contain stochastic properties, it simply calculates the total force exerted on a particle. In our experiments this means that all particles were pushed to the edge of the circle. We speculate this to be the case due to the fact that the particles always repel, leading to the edge being the only "stable" equilibrium for the particles. However, this is very problem specific so we do recommend further research in this aspect of this edit-type. Furthermore, it would be interesting to introduce random noise to the force vectors to better model noise often encountered in nature. This too could possible have effects on the outcomes of this specific edit-type. In general, for future research, we recommend experimenting with different edit-types given the role of importance they play in yielding a suitable outcome; both in speed and accuracy.

We also experimented with finding "magic formations" of our system. In this context "magic formations" would be unexpected positioning of the particles where E is minimized. For $k = 12$ we were not able to replicate the icosahedron form, what should be the magic formation. Our SA system was able to come close, where our minimal $E \approx 58$ differed from the (approximate) know minimum $E \approx 49$. It is possible that by using a different, more suited edit-type in combination with SA, it could be possible to achieve $E \approx 49$ for $k = 12$ and the "magic formation".

We also investigated the effect of different cooling schedules on the convergence to minimal E for $k = 6$. We found that, just as in SA for TSP, exponential cooling performed favorably compared to linear. The exponential scheme yields lower variance, faster convergence and a closer approximation of the minimal E. For our implementation of the Thompson problem and its parameters, it is clear that exponential is better than linear. However for future research we suggest considering other cooling schemes such as logarithmic or quadratic. Furthermore, we suggest investigating the effect Markov chain lengths have on the performance of the cooling schedule.

Finally, we also experimented with different Markov chain lengths. We used the exponential cooling scheme. In contrast to SA for TSP, we were not able to find a significant difference between the chain length and performance of SA. Due to the computational expense accrued with an increased chain length, based on our results, we would suggest using a lower chain length given the near identical performance of lower versus higher chain length. It was out of the scope of this research to also test the effect of chain length with a different cooling scheme; this too could prove to interesting.

References

[FB88] RH Frickel and BV Bronk. "Symmetries of configurations of charges on a sphere." In: *Canadian journal of chemistry* 66.9 (1988), pp. 2161–2165.

[Hoe19] Alfons Hoekstra. *Markov Chains, Metropolis Algorithm, and Simulated Annealing.* `https://canvas.uva.nl/courses/10868/files/1447100/download?wrap=1`. Dec. 2019.

[MDH96] JR Morris, DM Deaven, and KM Ho. "Genetic-algorithm energy minimization for point charges on a sphere." In: *Physical Review B* 53.4 (1996), R1740.