Lab assignment 11: Monte Carlo Simulations

Q1 Nicholas Pavanel, Q2 Hayley Agler, Q3 both

December 2020

Q1

a)

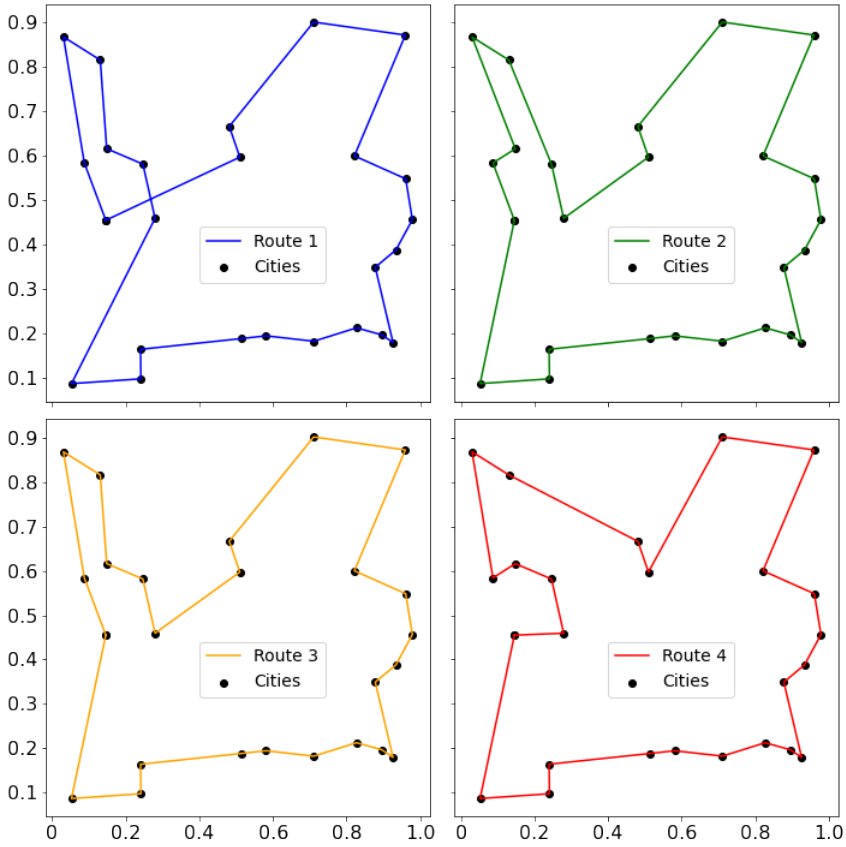

Figure 1: This figure shows different solutions to the travelling salesman problem as found by a simulated annealing optimization scheme.

In this subsection we use simulated annealing optimization to solve the travelling salesman problem. Our simplified travelling salesman problem is as follows: a salesman needs to visit n cities that are randomly distributed across a two dimensional domain in a route that minimizes the distance taken. Thus the question is how do we optimize the travelling salesman's route.

We optimize the travelling salesman's route by using simulated annealing. Figure 1 shows various routes that are produced as part of the optimization process.

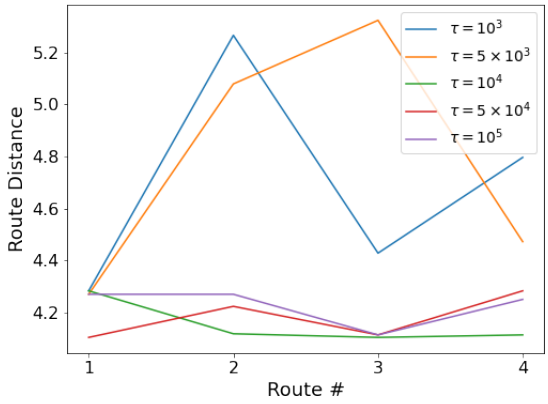

Figure 2: This figure shows the how the distance of routes found with a simulated annealing optimization scheme changes with τ. τ is commonly referred to as the cooling factor as it determines how long the optimization process will occur for, and is referred to as the scheduling time constraint in the travelling salesman problem.

In Figure 2 we explore the effects that the scheduling time constant τ has on the optimization process. Figure 2 shows two things. Firstly, that in general, larger values of τ produce routes that have shorter distances. This is shown by the fact that most routes for $\tau = 10^3, 5 \times 10^3$ have much larger distances than routes for larger values of τ, and is expected as larger values of τ allow the system to optimize the route for a longer amount of time. Secondly, Figure 2 shows that there exists an upper limit value of tau such that any further increases do not necessarily shorten the average distance of the routes. This is shown by the fact that values of $\tau = 10^4, 5 \times 10^4, 10^5$ have routes that generally all have comparable distances and is also expected as there does exist an optimal route.

b)

In this subsection we use simulated annealing optimization to find the global minima of two different two-variable functions. In contrast to subsection a) where we interchanged cities in our route to see if change resulted in a shorter route distance, we use Monte Carlo moves of the form $(x, y) \rightarrow (x + \delta x, y + \delta y)$ to check for function values that are lower. Steps of $(x + \delta x, y + \delta y)$ where made from a Gaussian distribution with a mean of zero and a standard deviation of one.

i)

In this subsection we find the global minima of the function

$$f(x, y) = x^2 - cos(4\pi x) + (y - 1)^2 \tag{1}$$

with the optimization scheme described at the start of the section. Figure 3 shows how the simulated annealing optimization scheme's values change over time. Figure 3 shows that the optimization scheme finds a global minima at $(x, y) = (0, 1)$. Figure 3 also shows the y-value smoothly converges to its minimal value, whereas the x-value initially fights between other local minima before converging the global minimum value.

ii)

In this subsection we find the global minima of the function

$$f(x, y) = cos(x) + cos(\sqrt{2}x) + cos(\sqrt{3}x) + (y - 1)^2 \tag{2}$$

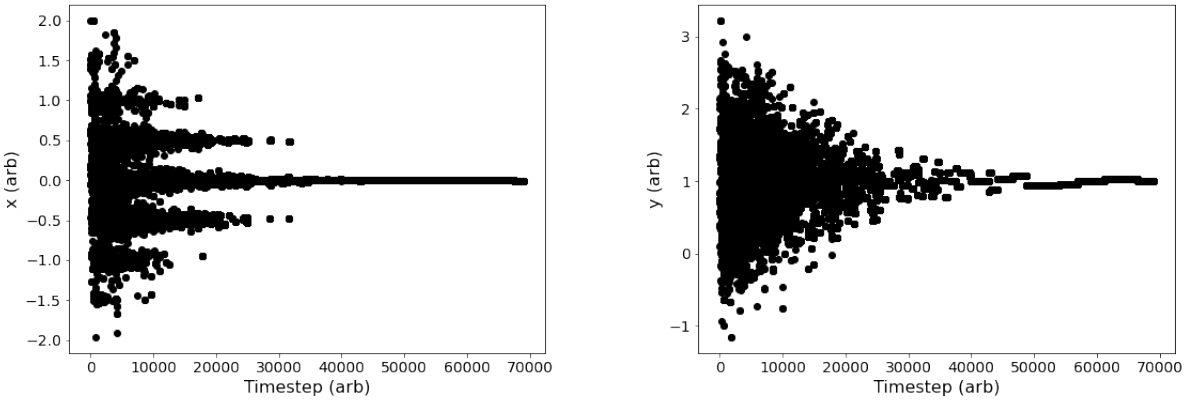

Figure 3: On left, this figure shows the time evolution of the x-values found by the simulated annealing optimization scheme over its runtime. On right, this figure shows the time evolution of the y-values found by the optimization scheme over its runtime.

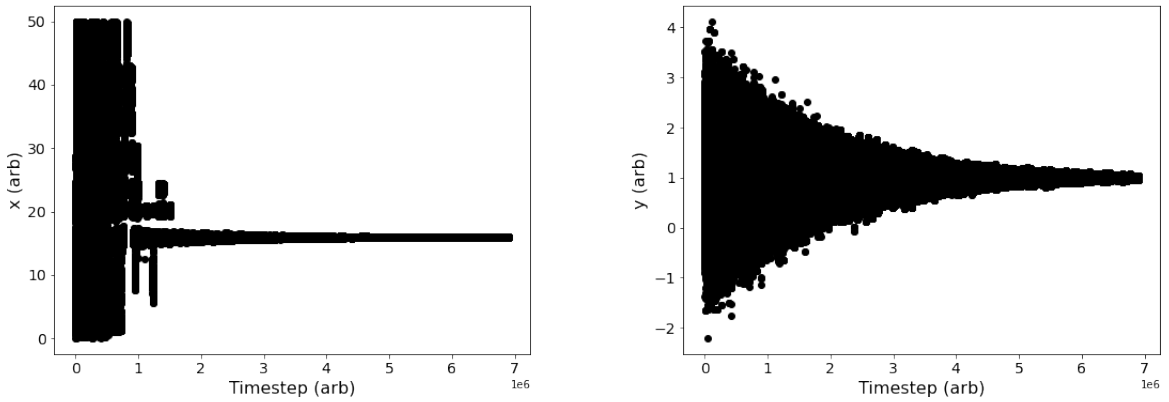

Figure 4: On left, this figure shows the time evolution of the x-values found by the simulated annealing optimization scheme over its runtime. On right, this figure shows the time evolution of the y-values found by the optimization scheme over its runtime.

with the optimization scheme described at the start of the section. Figure 4 shows how the simulated annealing optimization scheme's values change over time. Figure 4 shows that the optimization scheme finds a global minima at $(x, y) = (16, 1)$. Figure 4 also shows the y-value smoothly converges to its minimal value, whereas the x-value seems to clearly display random Monte Carlo motion before finding and latching onto the global minimum value.

Q2

In this question we were to write a program to perform a Markov chain Monte Carlo simulation of the Ising model on the square lattice for a system of 20 x 20 spins. The energy of the system was calculated with:

$$E = -J \sum_{(ij)} s_i s_j \tag{3}$$

with $J = 1$ and the magnetization with:

$$M = \Sigma_i s_i \tag{4}$$

where $s = \pm 1$ is the spin of the dipole. The probabilities of the values in the Markov chain were chosen with the Metropolis algorithm with acceptance probability given by:

$$P_a = \begin{cases} 1 & \text{if } E_j \leq E_i \\ \mathrm{e}^{-\beta(E_j - E_i)} & \text{if } E_j > E_i \end{cases} \tag{5}$$

where $\beta = 1/k_b T$, and $k_b = 1$, $T = 1$. The algorithm was run for a million Monte Carlo steps and the total magnetization for each run was plotted in Figure 5. The system develops a spontaneous magnetization around 40000 steps, as can be seen in the plot. This corresponds to where the energy begins to stabilize, again round 40000 steps. The program was run several more times to give the plots in figures 7-10. In these plots, the dipoles are lining up in the same direction to produce an overall ferromagnetic magnetization of the system, resulting in a low energy. If the dipoles line up with spins $s = 1$, the magnetization is positive like in fig 9, otherwise it is negative like in figures 5 and 7.

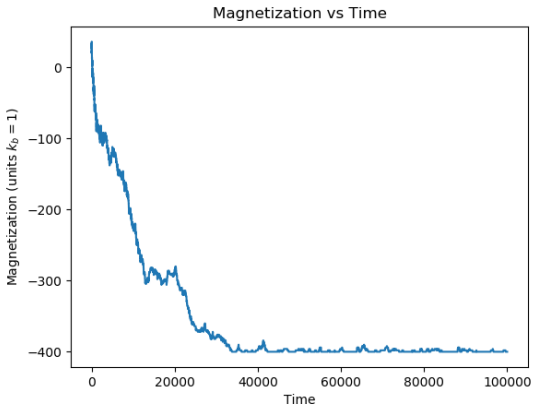

Figure 5: Total magnetization of the system, reaches negative magnetization at around 40000 steps.

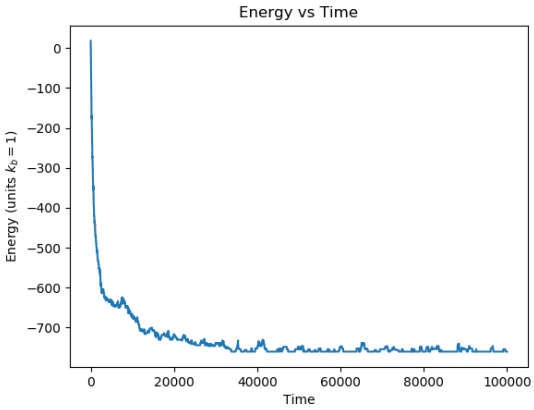

Figure 6: Total energy of the system, stabilizes at around 40000 steps.

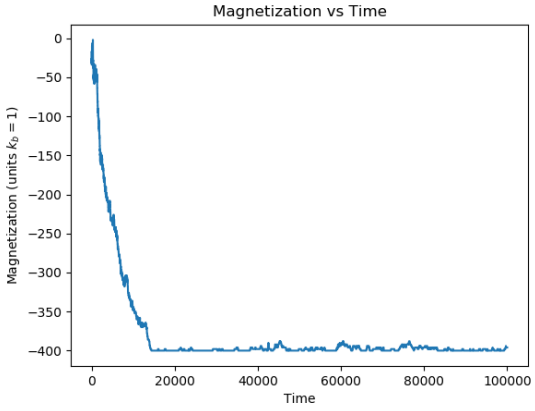

Figure 7: Total magnetization of the system, reaches negative magnetization at around 20000 steps.

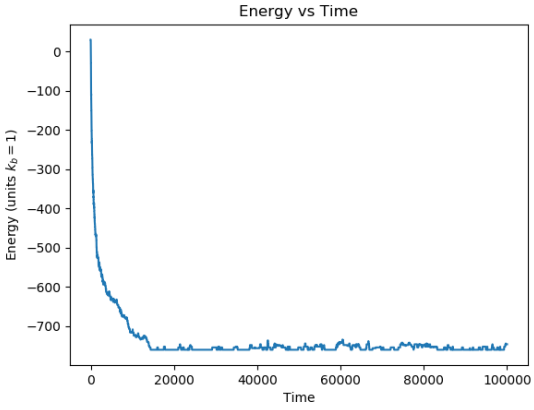

Figure 8: Total energy of the system, stabilizes at around 20000 steps.

An animation of the system displaying the dipole spins over time was produced for T=1,2, and 3. For T=1, we can see the dipoles of the system lining up to produce a magnetized system. For T=2 and T=3, the spins of the dipoles change more sporadically and do not begin to line up to produce a magnetization. This

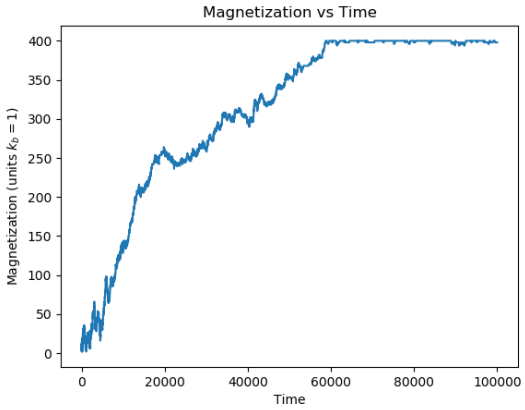

Figure 9: Total magnetization of the system, reaches positive magnetization at around 60000 steps.

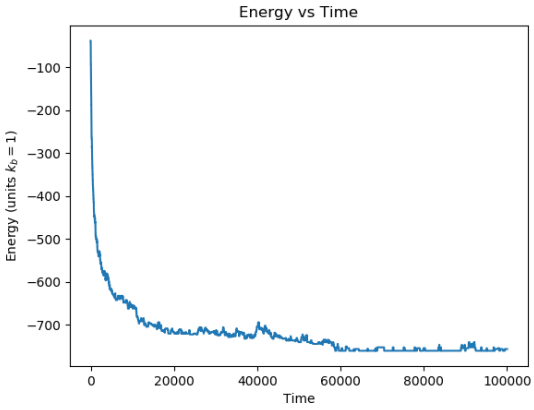

Figure 10: Total energy of the system, stabilizes at around 60000 steps.

is because a higher temperature leads to a high acceptance probability (equation 3) if $E_j > E_i$. For higher T, $e^{-\beta(E_j - E_i)}$ approaches 1, meaning more spin flips are accepted so the system has a more difficult time reaching magnetization.

Q3

a)

The script

`L11-Qprotein-start.py`

was run using the default parameters of $N = 30$, $T = 1.5$, $\epsilon = 5$, $n = 10^5$. The script creates two figures, one of the final structure of the protein and the other is energy as a function of the Monte Carlo step. In the energy figure (figure 11), the energy decreases in 3 major jumps from -10 to -20, -20 to -35 and then to -45.

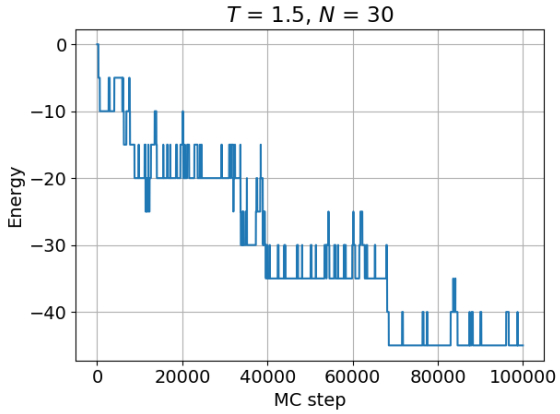

Figure 11: Q.3a)

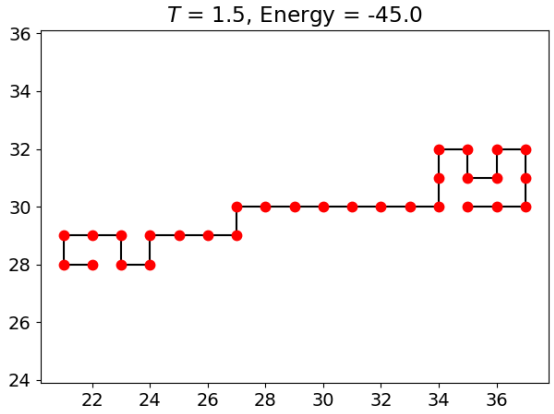

Figure 12: Q.3a)

Then the program was run again with temperatures T=0.5 and T=5.0. For the case of T=0.5 (see figures 13 and 14), within the first 1200 steps, the energy drops to -10 and remains -10 for the rest of the steps. The protein does not have many folds, which makes sense considering that the magnitude of the energy is small in comparison to the T=1.5 case.

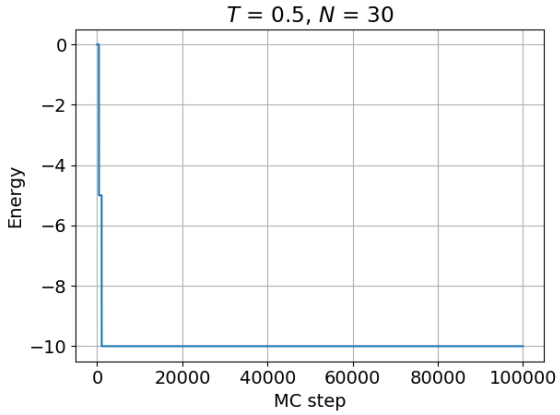

Figure 13: Q.3a)

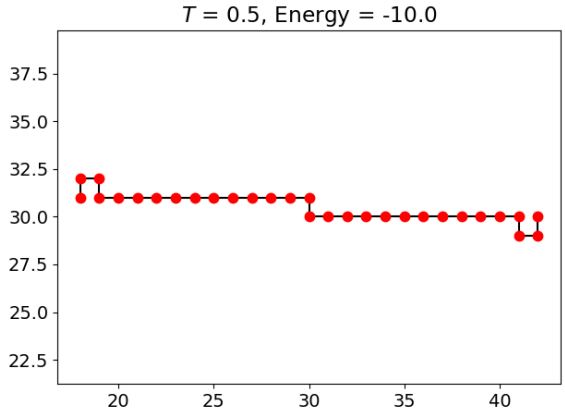

Figure 14: Q.3a)

For the T=5 case, there is much more variation in the energy, and it does not drop to a minimum right away like in the previous energy plots. When the initial temperature is higher, the protein has more energy, and there are many more possible folded states available. As such, we see much more variation in the energy as the protein goes through different folded states before reaching its lowest energy, and most folded state.

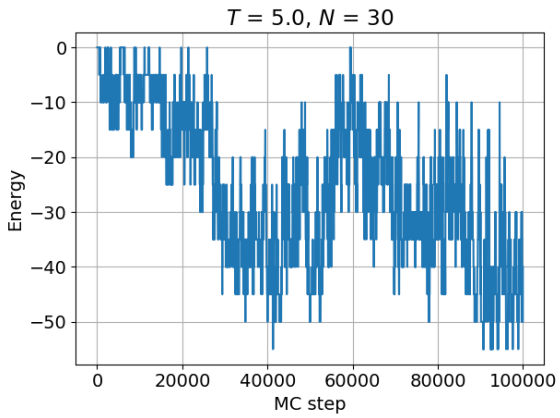

Figure 15: Q.3a)

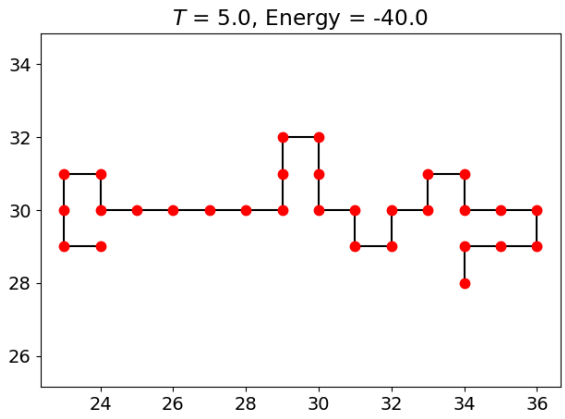

Figure 16: Q.3a)

b)

Now the simulation was run for 1,000,000 steps for the T-0.5 and T=1.5 cases. The typical energy of the protein is lower for the T=1.5 case. The more folds the protein has, the more interactions that can take places between monomers, hence resulting in a higher (more negative) energy. It seems like in the case with the lower temperature, the protein structure does not change much from its initial conditions. Since the initial structure is just a straight horizontal line, the low temperature means there is not many folds that occur which results in a lower temperature. This is exemplified in T=0.5 case. Since there is not a lot of initial energy, the protein does not have many possible folded states that it can reach. This is varies minimally from its initial condition.

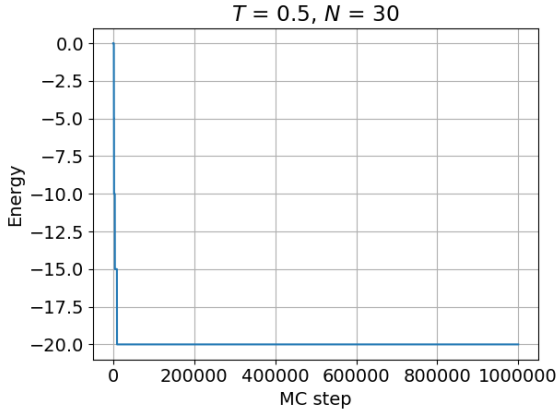

Figure 17: Q.3b)

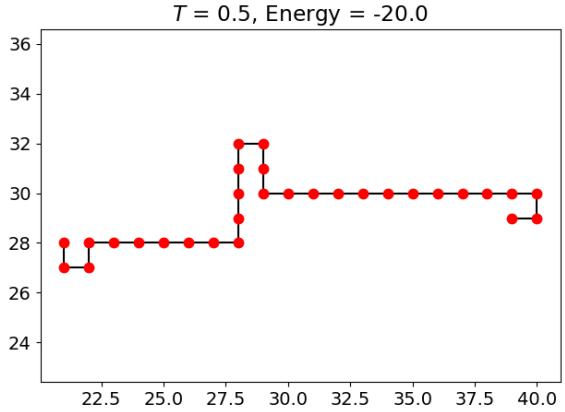

Figure 18: Q.3b)

c)

In this question, we start with a higher temperature and steadily decrease the temperature to T=0.5 over the course of the simulation. The approximate energy the protein has averaged over last half of simulations is -72.41.

Over the last quarter of the simulation, (T=0.5 approx),the energy is -70. For the T=0.5 case in b) when there was no simulated annealing, the energy was about -20. Simulated annealing slows the rate at which the temperature is dropped, and with a higher initial energy, allows the protein to fold into a more complex state. This is shown clearly in Figures 20 and 22 as the end protein with simulated annealing is much more

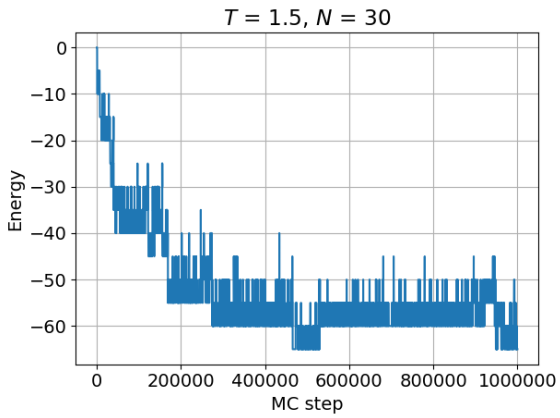

Figure 19: Q.3b)

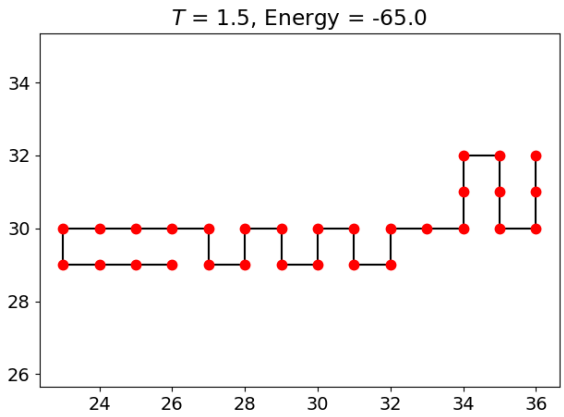

Figure 20: Q.3b)

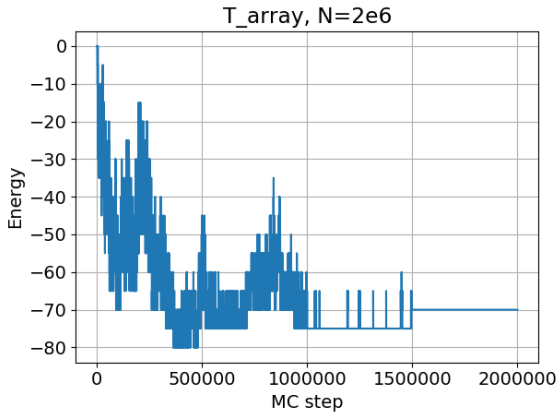

Figure 21: Q.3c)

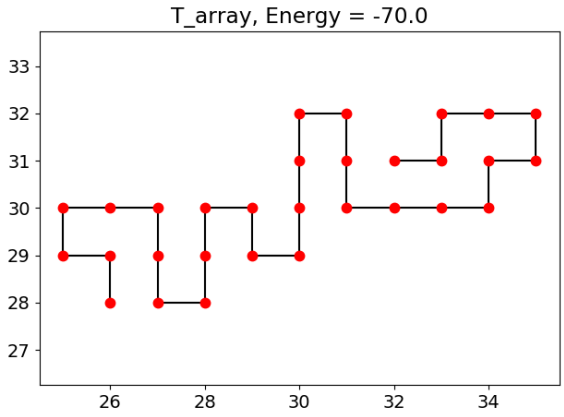

Figure 22: Q.3c)

complex than the end protein without simulated annealing.

d)

In this part, we simulate the protein folding starting at a temperature of T=10, stepping by $\delta T = 0.5$ until a temperature of T=0.5 is reached. At each temperature, 500,000 steps were simulated and the mean energy and standard deviation were calculated. The mean energy at each time step can be seen in figure 23 and the error bars represent the standard deviation in the mean. There are a few sharp jumps in the energy over small temperature ranges, like when energy goes from 2ϵ to 3ϵ, which are likely indicative of a phase shift.

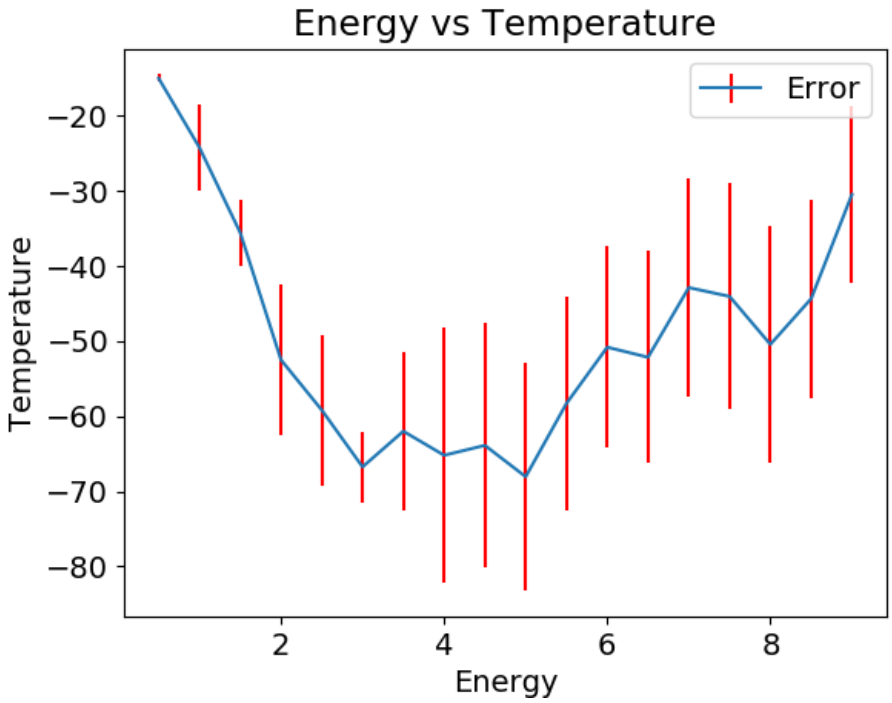

Figure 23: Energy vs temperature for Q.3d)