ECE5801: Neural Networks, Project #6
Simulated Annealing: Traveling Salesman Problem

In this project, you will learn to solve the well-known Traveling Salesman Problem (TSP) using a simulated annealing technique. Simulated annealing is one of the well-known global minimization techniques that has been widely used in many path optimization problems and can be theoretically proven to converge towards a global minimum given that the cooling schedule is properly designed for a very large number of iterations. This algorithm is simple to implement and works well on many search problems. The drawback is that it requires a very large number of iterations, and also there is no clear rule on the cooling schedule that leads to a global minimum. Nevertheless, you will notice that simulated annealing finds a decent solution that would be sufficient for many applications.

Traveling Salesman Problem (TSP): *Given n-city locations specified in a two-dimensional space, find the minimum tour length. The salesman must visit each and every city only once and should return to the starting city forming a closed path.*

The cost (or objective) function of TSP is the tour length, which is computed by adding Euclidian distances between two cities on the path. One intuitive approach for solving TSP is the direct search. The tour path is iteratively updated by changing the path and then choosing the new path if the tour length was reduced. The important question is then how to change the path. One well-known method is called the Lin's 2-cut algorithm. This algorithm is illustrated through Figure 1. First, two segments are randomly selected from the present tour path. In this example, the segments 3-4 and 8-9 are selected. Next, the cut ends are cross-connected to form a new path. In effect, this algorithm untwists or twists the path. If the path is chosen such that the cost function is decreased after the twist, it ends up untwisting the path from the twisted.

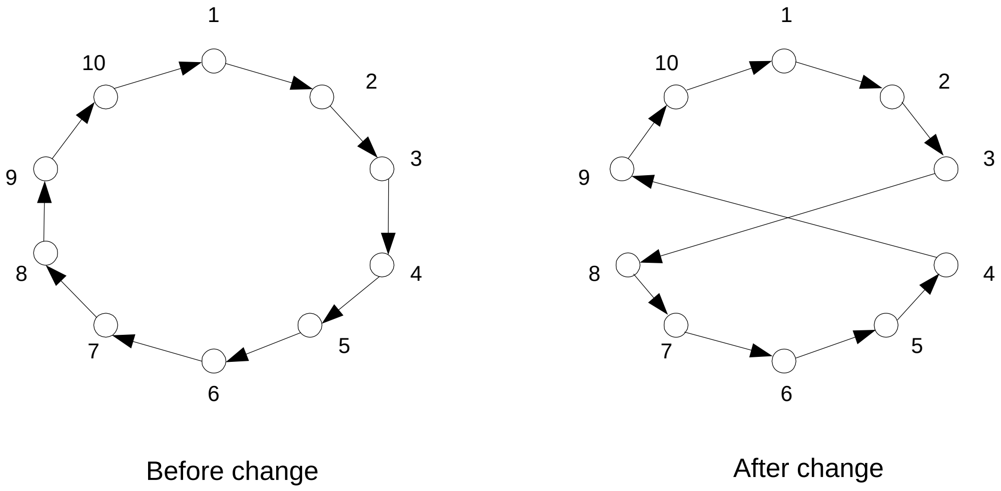

Figure 1: Lin's 2-Cut Algorithm

Let the cost function for the path P_i be denoted as $f(P_i)$. A direct local search procedure that uses the Lin's 2-cut algorithm can be summarized as shown next:

Procedure: Direct Local Search

```
1. Initialize Path, $P_i$, and $Maxiter$

2.  Do

   Generate Path, $P_j$ using the Lin's 2-cut algorithm
   If $f(P_j) < f(P_i)$ Then $P_i := P_j$
   $k=k+1$
 Loop While $k < Maxiter$
```

This direct search approach works well for small-scaled problems. However, as the scale of the problem gets larger (like in most real-world problems), the search can quickly get stuck at a local minimum, from which the solution can no longer improve no matter how many iterations are applied. If one attempts to do one-by-one search of the whole solution space of the TSP, it can take long time, i.e., many years. This is due to a particular characteristic of TSP that it belongs to a class of problems, called the NP-complete problem.

Simulated annealing overcomes the stuck at a local minimum problem using a hill-climbing technique in the search procedure. It is implemented by accepting the new path based on a Gibbs distribution, such that even if the cost of the new path is higher than the old path, the new path may be accepted according to a probability distribution established by the Gibbs function. The details of the simulated algorithm for solving TSP are summarized below.

Procedure: Simulated annealing

```
1. Initialize Path, $P_i$, Temperature $T$, Equilibrium Length $L_k$, and $Maxiter$

       k=0

2.  Do

       For j=1 to $L_k$
           Generate Path, $P_j$ using the Lin's 2-cut algorithm
           If $f(P_j) < f(P_i)$ Then $P_i := P_j$
           Else
              if $\exp( (f(P_i) - f(P_j))/T) > Rnd$ Then $P_i := P_j$
           End if
       Next j

       $k=k+1$
       Reduce Temperature $T$
       Recompute $L_k$
 Loop While $k < Maxiter$
```

In the above procedure, how to initialize and reduce the temperature (called cooling schedule) critically affects the performance. Please remember that the cooling schedule is very important for finding a solution close to the global minimum.

<u>The Given Task</u>

- Implement both the direct local search and simulated annealing for the 100city.txt data given. The maximum iteration should be greater than 40,000.
- Run several tests based on random initialization of the tour path for direct and simulated annealing search.
- Collect the statistical data of your runs and compare the performance of the local and simulated annealing search. Print the path plots for the best cases of both the local and simulated annealing.

Report:

1. Describe the TSP problem and the basic principle of the simulated annealing.
2. Compare the results and discuss. Attach the path plots for the direct local search and the simulated annealing along with the tour length. The final tour length will be used to judge your cooling schedule.
3. Conclusion
4. Program listing