Lindsey Blackmore

Shivanee Persaud

CS 3510: TSP Algorithm Explanation

Our group developed a simulated annealing algorithm in order to solve the traveling salesman problem (TSP). The algorithm is based on the practice of annealing glass or metal, which is the process of raising the material to a high temperature and then gradually reducing the temperature. This allows the material to grow outwards in local areas leading to less stress. Using this process, simulated annealing randomly changes an original path based on a logical, decreasing temperature.

1. Start with a random tour. This was implemented using the nearest neighbor algorithm. Store this solution and its tour length for later comparisons. The pseudocode below is what the initial_solution function in our code is based on.

```
select a random start node
add that start node to a solution array
create a set of unvisited nodes that includes all points except the starting one
while there are still nodes in the unvisited set:
        select the nearest neighbor to the most recently visited node
        add that node to the solution
        remove that node from the unvisited set
return the solution tour and the tour length
```

2. Select a new candidate tour. This was implemented by selecting two random points and reversing the path that was between them.

```
select random node1
select random node2
iterate through nodes in reverse order
```

3. If this candidate tour has a shorter tour length, accept it and store the solution and tour length.

```
add first node to end of tour to create full loop
calculate distance of candidate tour
if the new distance is less than the distance that was stored
        update values
```

4. If the candidate tour is longer than the current solution, there is a still a chance that it will be accepted.

```
calculate probability between new distance and stored distance
if calculated probability of new distance is high
        update values with new distance
```

5. Go back to step 2 and repeat a certain number of iterations, decreasing the temperature by an alpha factor each iteration.

```
update temp value based on alpha
update iteration by 1
add distance to running list of distances
```