

**CPE341 Optimization Design and Reliablity Engineering**

**Topic :**

**`**

**Short path to travel around the airbnb hotel based in New york city using Simulated annealing**

**Created By**

**Mr. Napas Vinitnantharat**

**Present to**

**Prof Naruemon Wattana**

**Department of Computer Engineering, Faculty of Engineering**

**King Mongkut’s University of Technology Thonburi**

**Semester 2/2022**

**Chapter 1: Introduction**

**1.1 Project Description and Scope**

This project applies Simulate Annealing(SA) to solve the Traveling Saleman Problem(TSP).This Project simulate the situation of person who want to travel hotel in New York City and minimize the distance to travel all of the hotel.In this project will study applying simulated annealing to solve single objective function and compare the performance with brute force algorithm.Then we apply simulated annealing to solved the problem with multiobjective function by simulate the situation that someone want to travel 2 trip with the minimize distance and those two trip doesn’t has the same hotel to visit.

**Chapter 2: Model Formulation**

**2.2 Variable Definition**

- Path to travel all hotel

- Path to travel in the first trip

- Path to travel in the second trip

- Path to travel form hotel i to hotel j

) - Total distance to travel in path

**2.2 Objectives**

**2.2.1 Haversine Formula**

Minize the distance route to travel hotels in New york city. The total distance can be calculated by sum all of the distance between each hotel.In order to calculate the total distance between the hotel by using latitude and longitude The haversine formula have been used as shown in the following equation.

d - Distance between to hotel i and hotel j

r - Earth radius

- Destination latitude

- Starting latitude

- Destination longitude

- Starting longitude

**2.2.1 Minimize First Trip Distance**

)=

**2.2.2 Minimize Second Trip Distance**

)=

**2.2.3 Weight Sum Method**

In This study we set the weight at 0.5.So the WSM Objective is

)=+

**2.3 Decision Variable**

is set to the path to travel from hotel 1 to hotel 12 for single objective function and 18hotels for multi objective.It can represent as a list of number

**2.4 Contraints**

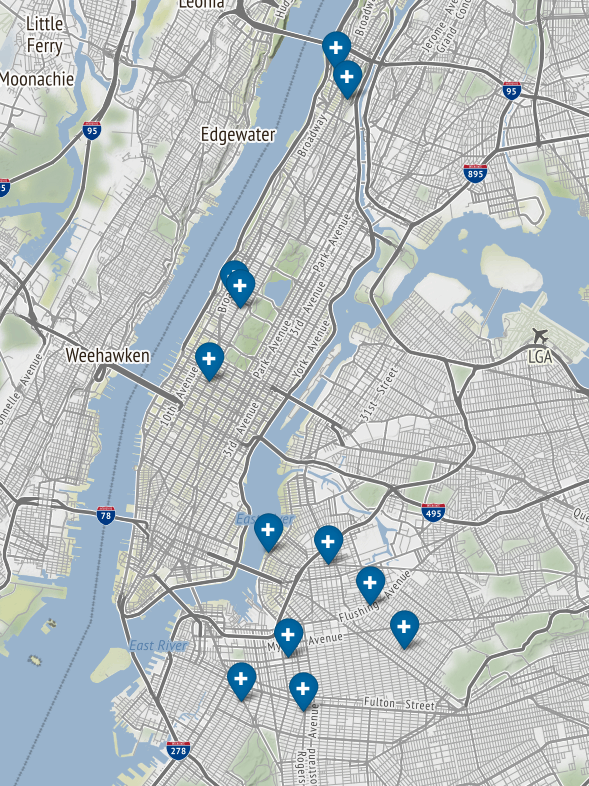
In this problem doesn’t have any contraints

**Chapter 3: Input Data and Problem Size**

**3.1 Input Data**

In this data input we use the dataset from kaggle Airbnb Open Dataset dataset.we sample 12 of the hotel in the dataset to solve single objective.These are the 12 hotels host is show as the following

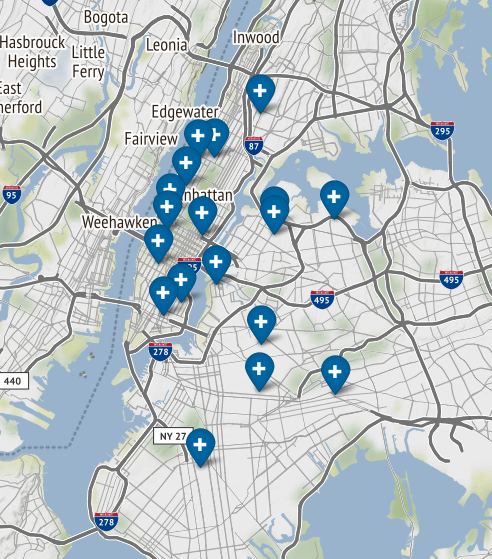
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| host name | borough | lat | long | price | service fee |
| Marjolein | Manhattan | 40.78384 | -73.97696 | 90 | 18 |
| Brittany & Matt | Manhattan | 40.8359 | -73.93842 | 458 | 92 |
| Andy | Manhattan | 40.78165 | -73.9749 | 637 | 127 |
| Jenn | Brooklyn | 40.67925 | -73.97463 | 462 | 92 |
| Camila | Brooklyn | 40.6768 | -73.95346 | 503 | 101 |
| Danielle | Manhattan | 40.84353 | -73.94201 | 1040 | 208 |
| Jose | Brooklyn | 40.69284 | -73.91872 | 116 | 23 |
| Mathew | Brooklyn | 40.71492 | -73.94492 | 85 | 17 |
| Merrily | Brooklyn | 40.70443 | -73.93039 | 987 | 197 |
| Manhattan At Times Square | Manhattan | 40.76267 | -73.98557 | 1111 | 222 |
| Aurora | Brooklyn | 40.71819 | -73.96532 | 79 | 16 |
| Jane | Brooklyn | 40.69071 | -73.95852 | 93 | 19 |



**Figure 3.1 Hotel location that have apply to test single objective problem**

for multi objective task,we sample 18 of the hotel in the dataset to solve multi objective.These are the 18 hotels host is show as the following

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| host name | borough | lat | long | price | service fee |
| Julia | Brooklyn | 40.73675 | -73.95443 | 867 | 173 |
| Lorraine | Manhattan | 40.72048 | -73.9906 | 355 | 71 |
| Cara | Manhattan | 40.72755 | -73.97876 | 491 | 98 |
| Patrice | Manhattan | 40.80283 | -73.95569 | 1170 | 234 |
| Hilary | Brooklyn | 40.64170 | -73.96554 | 697 | 139 |
| Lawrence | Brooklyn | 40.68096 | -73.92512 | 486 | 97 |
| Asad | Queens | 40.77046 | -73.87336 | 306 | 61 |
| Brad | Manhattan | 40.76249 | -73.96428 | 736 | 147 |
| Dominique | Manhattan | 40.8022 | -73.9666 | 904 | 181 |
| Lightning | Manhattan | 40.76555 | -73.98827 | 474 | 95 |
| Kimberly | Brooklyn | 40.67965 | -73.87258 | 1138 | 228 |
| Ryan | Queens | 40.76736 | -73.91462 | 152 | 30 |
| Henry | Manhattan | 40.77428 | -73.98594 | 1048 | 210 |
| Brian | Manhattan | 40.78853 | -73.9751 | 928 | 186 |
| Nathaniel | Bronx | 40.82575 | -73.92421 | 66 | 13 |
| A | Queens | 40.76266 | -73.91451 | 1151 | 230 |
| Andrew | Brooklyn | 40.70587 | -73.92358 | 1116 | 223 |
| Plamen | Manhattan | 40.74781 | -73.99436 | 198 | 40 |



**Figure3.2 Hotel location that have apply to test multi objective problem**

**Reference Data Source**

<https://www.kaggle.com/datasets/arianazmoudeh/airbnbopendata>

**3.2 Problem Size**

for a both single objective problem there are 19,958,400 possible solutions. The calculation is calculate by (12-1)!/2 = 19,958,400 solutions.for multi objective problem.There are (18-1)!/2 = 177.84 million million possible solutions.

**Chapter 4: Algorithm**

**4.1 Simulated Annealing**

In This Project the Simulated Annealing is used to find the best path solution.Simulated Annealing is c computatioal method borrowing inspiration from the field of physic introsuced by.It simulate the physical process of solid annealing. This method has been one of heuristic model to avoiding local minina.The base concept of this algorithm is accept worse candidate base o the probability dependent on the temperature and the rate of change of the fitness value or cost.

P =

P - The Probability of accepting the new solution candidate

fitness(s) - In This problem we use the total distance of the path so if delta fitness < 0 mean the the new route is shorter than the previous route.

T - Tempature which is use in the control parameter

**4.1.1 Algorithm**

While Current\_Temp <= Final\_temp:

for i until i = iterationpertemp:

find neighbor

calculate the neighbor fitness value

if fitness(neighbor) < fitness(S) # new solution is better

set neighbor to be the new solution

else

random number r in range 0 to 1

if r <

set neighbor to be the new solution

else

do nothing

update the tempurature T = T\*α

**4.2. Single Objective Problem**

for the single objective problem, 12 hotels have been considering use to find the short path to travel all hotels by using Simulated Annealing. Paramter is setup from the following

**Parameters:**

1. IntialTemp = 200

2. IteratePerTemp =100

3. FinalTemp = 0.01

4. alpha = 0.98

**Path Encoding**

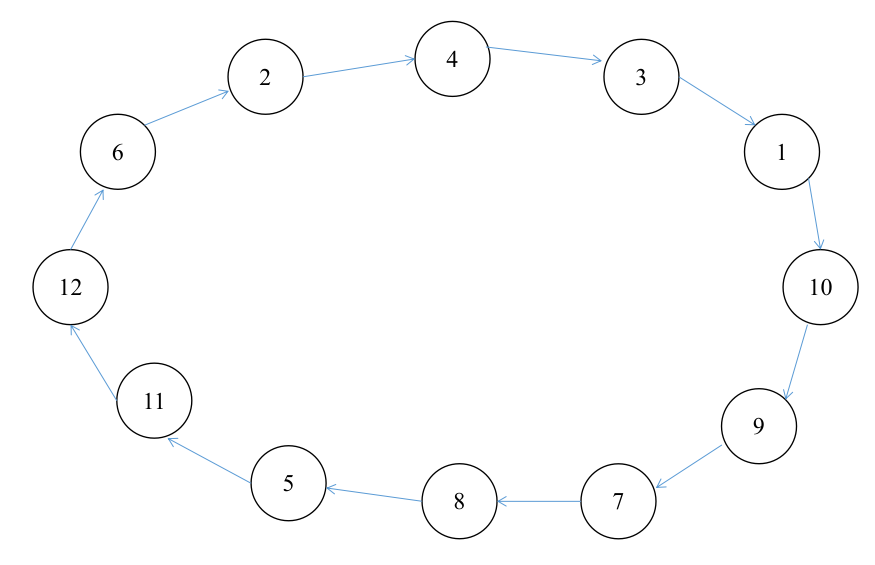
The path to travel hotel is encode as the following

1. The path is represent as a list, each number in the list is represent the index of the hotel.The length of the list is equal to the number of hotel in to visit.In this case it 12 hotels.
2. The sequence of list is represent as the order that person going to visit each hotel.

**Array P Encoding**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** | **2** | **4** | **3** | **1** | **10** | **9** | **7** | **8** | **5** | **11** | **12** |

**Array P Decoding to travel path**



**Figure 4.1** Path encoding and Decoding of single objective

**4.2.1 Find Neighbor**

In order to find the Neighbor solution of the Simulate Annealing, we can apply the swap node technique as show in the following.

**Step 1.** random the number in range equal to the number of hotel in this case we have 12 hotels

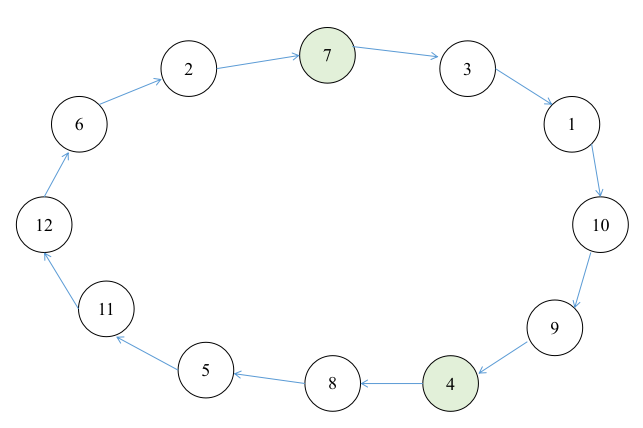
**Step 2.** Swap the number base on those index as show in the **Figure 4.2**

**Current Solution**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** | **2** | **4** | **3** | **12** | **10** | **9** | **7** | **8** | **5** | **1** | **11** |

**Neighbor Solution**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** | **2** | **7** | **3** | **12** | **10** | **9** | **4** | **8** | **5** | **1** | **11** |



**Figure 4.2** Find neighbor by swap node for single objective

**Figure 4.2** Show the mechaism of find neighbor solution by swap node by exchanges the position of two cities in a route. Two positions, i and j are selected at random and the cities in these positions are swapped with each other.for the example Figure 4.2 show path 6->2->4->3->12->10->9->7->8->5->1->11 of the current solution to path and neighbor solution 6->2->7->3->12->10->9->4->8->5->1->11

**4.2.1 Fitness value**

for single objective function we use distance as a fitness value.The distance of two point can calculated by using Haversine function.

**4.3. Multi Objective Problem**

We implement a simple shortest path to travel all hotel solving algorithm using Simulated Annealing

**Parameters:**

1. IntialTemp = 200

2. IteratePerTemp =200

3. FinalTemp = 0.01

4. alpha = 0.98

**Path Encoding**

The path to travel hotel is encode as the following

1.The path is represent as a list, each number in the list is represent the index of the hotel.for first half of the list is set to First Trip of travel and seccond half of the list is the second trip

2.The sequence of list is represent as the order that person going to visit each hotel.

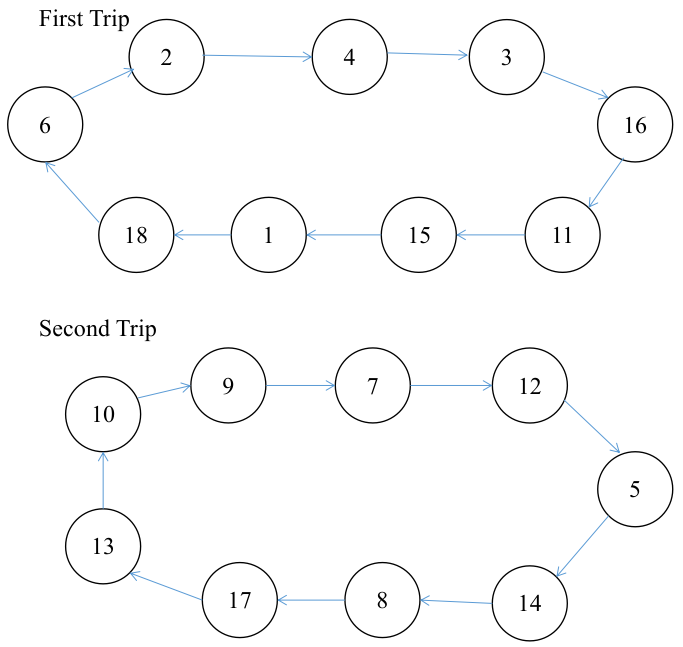
**Array P Encoding**

**First Trip**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** | **2** | **4** | **3** | **16** | **11** | **15** | **1** | **18** |

**Second Trip**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **10** | **9** | **7** | **12** | **5** | **14** | **8** | **17** | **13** |



**Figure 4.3** Path encoding and Decoding Multi Objective TSP problem

**4.3.1 Algorithm Modify for Multi Objective**

for the multi objective function we have to modify the algorithm to support the problem by using Weight Sum method(WSM).The algorithm can be show as the following.

While Current\_Temp <= Final\_temp:

for i until i = iterationpertemp:

find neighbor

calculate the neighbor fitness value of first trip

calculate the neighbor fitness value of second trip

WSM fitness = 0.5\*fitness trip 1 + 0.5\*fitness trip 2

if WSM fitness < fitness(S) # new solution is better

set neighbor to be the new solution

else

random number r in range 0 to 1

if r <

set neighbor to be the new solution

else

do nothing

update the tempurature T = T\*α

**4.3.2 Find Neighbor**

For the nulti objective part.In order to find the Neighbor solution of the Simulate Annealing we can use swap node technique similar to the single objective but apply on 2 trips.

**Current Solution**

**First Trip**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** | **2** | **4** | **3** | **16** | **11** | **15** | **1** | **18** |

**Second Trip**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **10** | **9** | **7** | **12** | **5** | **14** | **8** | **17** | **13** |

**Neighbor Solution**

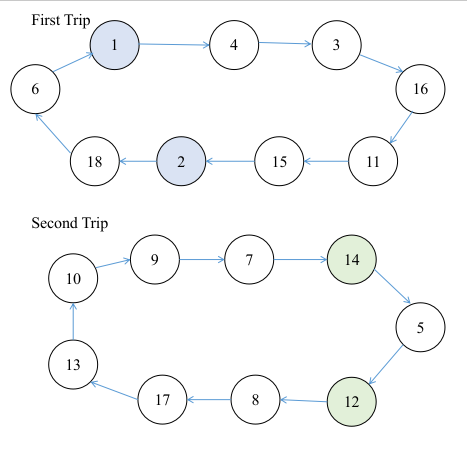
**First Trip**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **6** | **1** | **4** | **3** | **16** | **11** | **15** | **2** | **18** |

**Second Trip**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **10** | **9** | **7** | **14** | **5** | **12** | **8** | **17** | **13** |

**Figure 4.4** Find Neighbor by swap node for multi objective



**Figure 4.5** Find Neighbor by swap node for Multi Objective TSP problem

**4.3.3 Fitness value**

Fitness value can be calculated by using WSM method to minize the two trip objective function.WSM objective can calculated by the follow equation

)=+

- Path to travel in the first trip

- Path to travel in the second trip

- Path to travel form hotel i to hotel j

) - Total distance to travel in path

**Chapter 5: ​Results and Discussion**

**5.1. Single Objective Problem**

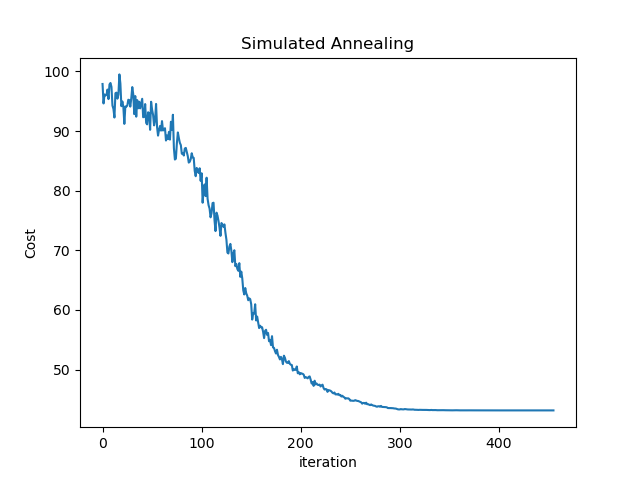
for the Multi Objective Problem.we try to minize the distance of two travel trip. In each trip has 9 hotels to visited.The fitness value is calculate by using WSM method and weight w at 0.5

**5.1.1 Simulated Annealing on Single Objective problem**

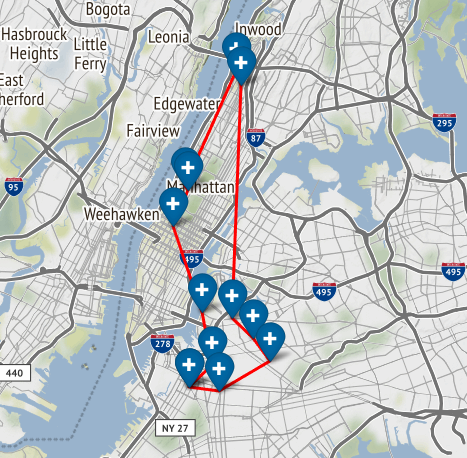
by running Simulated Annealing repeating 100 times and average the result as show in **Figure 5.2.2** . The average fitness value of the final solution is 43.05865. The best solution we have a fitness value of 42.9680.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Result** | **Time** |
| - InitialTemp = 100  - IterationPertemp = 100  - Alpha =0.98  - FinalTemp = 0.01 | **Best fitness value** is 42.9680  **Average fitness value** of the final solution is 43.05865 | 13.5256 seconds |

**Simulated Annealing on Single Objective TSP problem**

****

**Figure 5.1.1 Average Converage of applying Simulated Annealing to solve Single Objective Traveling Salemen Problem**



**Figure 5.1.2 Optinum Solution of Single Objective TSP problem**

**5.1.1 Compare with the Brute Force on Single Objective problem**

|  |  |
| --- | --- |
| **Algorithm** | **Time Excution** |
| Simulated Annealing | 13.5256 seconds |
| Brute Force | > 4 hours |

**Table 5.1.1.1** Time and Quality of soultion Comparison between SA and Brute Force

from Table 5.1.1.1 By applying test SA and brute force algotihm on single objective problem with 12 hotels. We found that the solution from brute force algorithm have the same result with the Simulated Annealing.time consumed by brute force algithm

|  |  |  |  |
| --- | --- | --- | --- |
| **Hotel Size** | **Problem Size** | **SA Time Excution** | **Brute Force Time Excution** |
| 8 | 20,160 | 8.30 seconds | 7.75 seconds |
| 9 | 181,440 | 11.31 seconds | 79.35 seconds |
| 12 | 19,958,400 | 13.52 seconds | > 4 hours |

**Table 5.1.1.2** Time and Quality of soultion Comparison between SA and Brute Force with different size of problem

**Table 5.1.1.2** Show Simulated Annealing and Brute Force Algorithm have been applied with different problem size to test the performance of the algorithm.Time for Excution Time od Brute Force algorithm is larger than the SA Time Excution when Problem size is more than 9 hotels

**5.2. Multi Objective Problem**

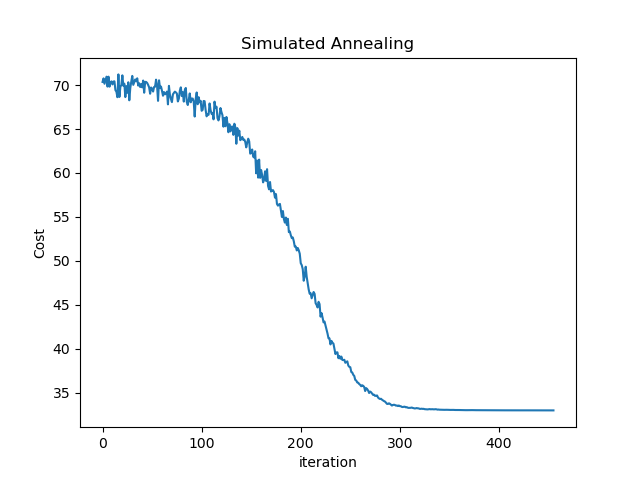
for the Multi Objective Problem.we try to minize the distance of two travel trip. In each trip has 9 hotels to visited.The fitness value is calculate by using WSM method and weight w at 0.5

**5.2.1 Simulated Annealing on Multi Objective problem**

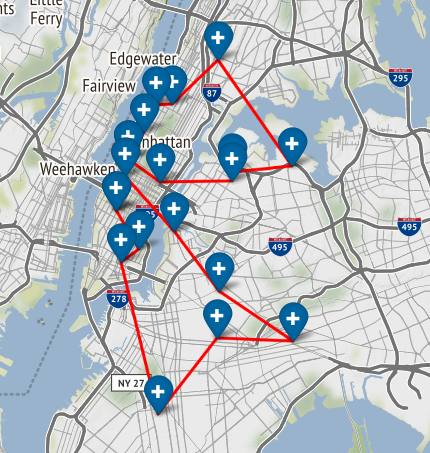
by running Simulated Annealing repeating 100 times and the average result is shown in **Figure 5.2.1**. The average fitness value of the final solution is 32.3745. The best solution we have a fitness value of 32.3692.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Result** | **Time** |
| - InitialTemp = 100  - IterationPertemp = 100  - Alpha =0.98  - FinalTemp = 0.01 | **Best fitness value** is 32.3692  **Average fitness value** of the final solution is 32.3745 | 27.6712 seconds |

**Simulated Annealing on Multi Objective TSP problem**

****

**Figure 5.2.1 Average Converage of applying Simulated Annealing to solve Traveling Salemen Problem**



**Figure 5.2.2 Solution of Multi Objective TSP problem by using Simulated Annealing**

In Figure 5.2.2 Show the solution created by using Simulated Annealing.from the observation,The result of SA show two travel trip with two loop.

**Result Discussion**

In single and multi objective Travelling Saleman Problem(TSP), Simulated Annealing (SA) could use in solve those problem.By applying Simulated Annealing with Single Objective problem.The solution of the problem by using SA is same as using brute force algorithm.The excution time of Simulated Annealing is considering faster than using brute force algorithm when the problem size is large.

In multi objective TSP problem, By applying SA and weighted sum method,The result show satisfied result.

**Reference**

[ 1 ] Jain, Y. (2021, December 11). Simulated Annealing - AITS Journal - Medium. \*Medium*.https://medium.com/ai-techsystems/simulated-annealing-580f73bd807a

[ 2 ] Alvarenga, C. W. (2021, December 15). How to Implement Simulated Annealing Algorithm in Python. *Medium*. https://medium.com/swlh/how-to-implement-simulated-annealing-algorithm-in-python-ab196c2f56a0

[ 3 ] Liang, F. (2022, November 7). Optimization Techniques — Simulated Annealing - Towards Data Science. *Medium*. https://towardsdatascience.com/optimization-techniques-simulated-annealing-d6a4785a1de7

[ 4 ] Allanah, F. (2023, January 23). Traveling Salesman Problem using Simulated Annealing | Medium. *Medium*. https://medium.com/@francis.allanah/travelling-salesman-problem-using-simulated-annealing-f547a71ab3c6