Analysis of Algorithms

Development of an Optimization Algorithm to Solve 2-TSP Problem

Yusuf İbrahim Matur 524123004

Mehmet Efe Selamet 150122058

Aşın Yavuz Tuna 150121016

1. **Problem:**

Travelling Salesman Problem (a.k.a TSP) is a popular optimization problem where a salesman is to visit all cities exactly once to find the shortest route between these set of cities that must be visited. It's known to be in the category of NP-Hard problems, which is considered that there are no solutions in polynomial time.There are many variations of TSP problem that are solved by different optimization methods.One of the variations of TSP is called mTSP, where there are multiple number of salesmans that are supposed to visit a set of cities. In this study, for given sets of cities with different densities, an algorithm for approaching the optimal solution of mTSP problem where m=2 (for 2 salesmans) is developed.

2. Method

There are 2 methods developed for solving 2-TSP problem:

1- K-Means + Nearest Neighbors + 2-OPT Method (Yusuf İbrahim Matur, Mehmet Efe Selamet)
2- AYTUNA Method (Aşkın Yavuz Tuna)

2.a. K-Means + Nearest Neighbors + 2-OPT Method O(m * n^2)

2.a.1 K-Means Clustering Algorithm O(k * n, k is the number of repeats in centeroid convergence)

K-means algorithm is used as an iterative clustering algorithm. It takes the distance as the measurement standard, gives the K clusters in the data set, calculates the average value of the distance, and then gives the initial centroid. Each cluster is described by the centroid [4]. The goal is to form the disjoint groups of n data points {x1,x2,…,xn} into k < n sets{S1,S2,…,Sk} to minimize the total average value (including the square distance from the point to the centroid).[1]

2.a.2. Nearest Neighbors Algorithm O(n^2)

It starts with a randomly chosen city and repeatedly adds the closest unvisited city to the last city in the tour until all the cities have been visited [16]. The steps of the nearest-neighbour algorithm are given as: Step1: Randomly pick the initial city. Step2: Find the closest unvisited city and add to the current tour. Step3: Is the cardinality of the unvisited cities is ? If not, repeat Step2, otherwise go to Step4. Step4: Terminate the algorithm. Since the tours quality might depend on the starting city chooses, a better result can be obtained by repeating the procedures for different starting city

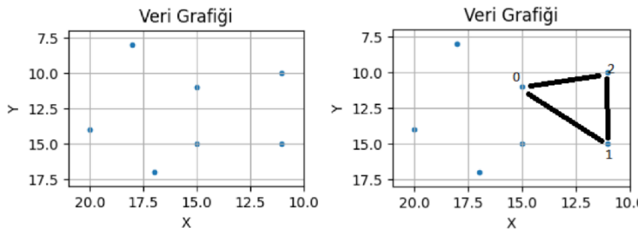
2.a.3. 2-OPT Algorithm O(m * n^2 , m = amount of swap limit)

In optimization, the 2-opt algorithm is a straightforward local search method for addressing the traveling salesman problem (TSP). Originally introduced by Croes in 1958, with the underlying concept previously suggested by Flood, the 2-opt technique aims to improve an existing route by eliminating self-crossings. The core idea is to take segments of the tour that intersect and rearrange them to form a shorter, non-crossing path. A thorough 2-opt local search involves evaluating every feasible pair of edges and performing the swap if it results in a shorter tour. This method is not only applicable to the TSP but also extends to various related problems, such as the vehicle routing problem (VRP) and its capacitated version, with slight adjustments to accommodate specific constraints.

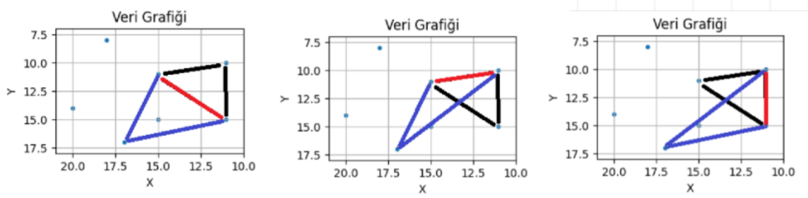
2.b. AYTuna Method O(N^2)

1. Put first 3 point to arraylist.
2. Check between which points new point should be putted then put it between them.
3. Do this for every element

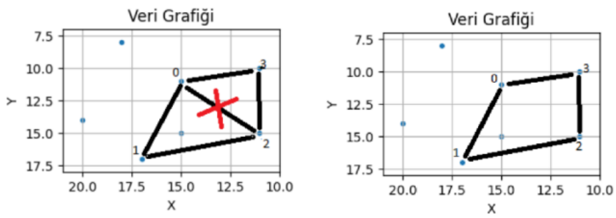
Put first 3 point to arraylist



Check between which points new point should be putted



Pick the most optimal connection for the new point



3.Tests

2opt is a very optimized algorithm but it is not time or space efficient. Because of this for larger number of input we have decided to use our own implemented algorithm AYTuna for 50000 input test.

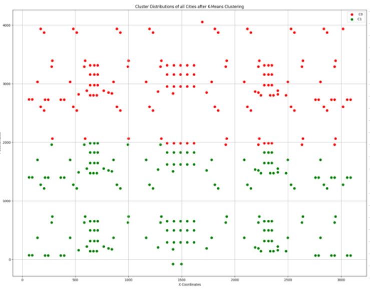
3 Test cases with different city numbers are used.

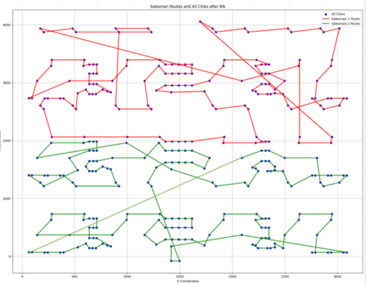
Test Case 1	318 Cities	NN Dist: 56524	KM+N N+2 Opt Dist: 46816	AYTuna Dist:48408
Test Case 2	938 Cities	NN Dist: 3490	KM+N N+2 Opt Dist: 3069	AYTuna Dist: 3328
Test Case 3	7397 Cities	NN Dist: 28824582	KM+N N+2 Opt Dist: 28717817	AYTuna Dist: 29121303
Test Case 4	50000 Cities	NN Dist: -	-	AYTuna Dist: 3820356

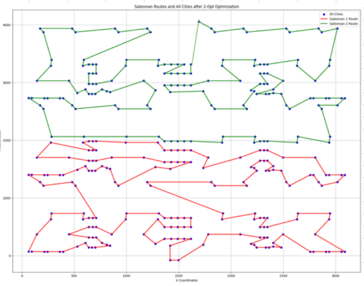
First test Case:318 Cities

Method:K-Means + Nearest Neighbors + 2-OPT Method

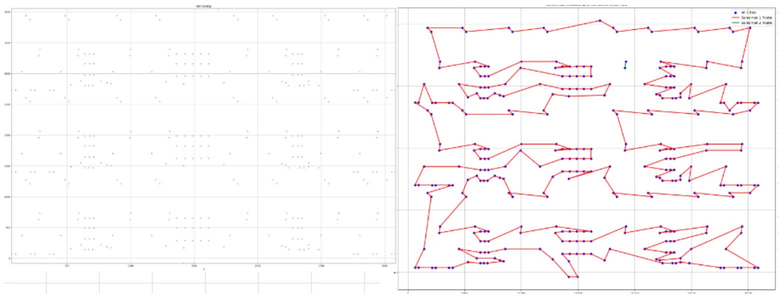
K Means KM + NN KM + NN + 2-OPT







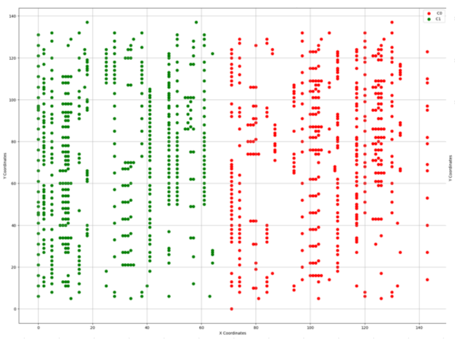
Method: AYTuna method

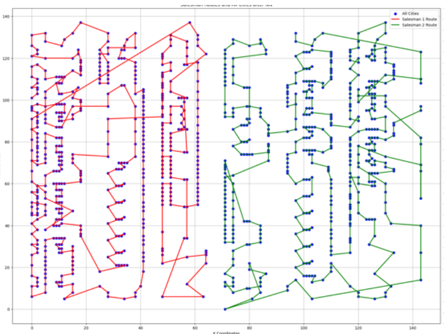


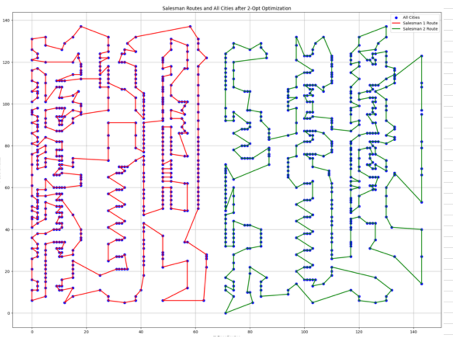
Second Case: 938 Cities

Method:K-Means + Nearest Neighbors + 2-OPT Method

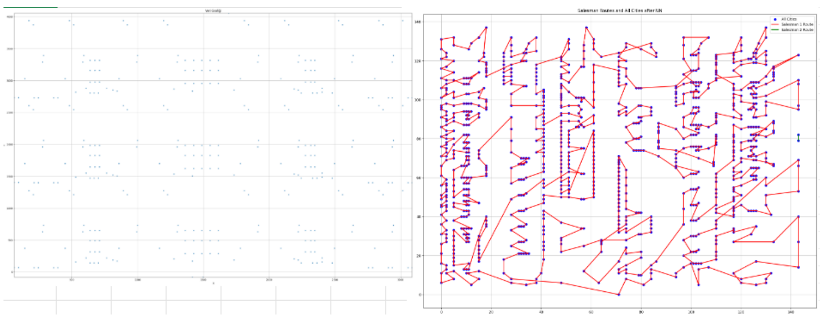
K Means KM + NN KM + NN + 2-OPT







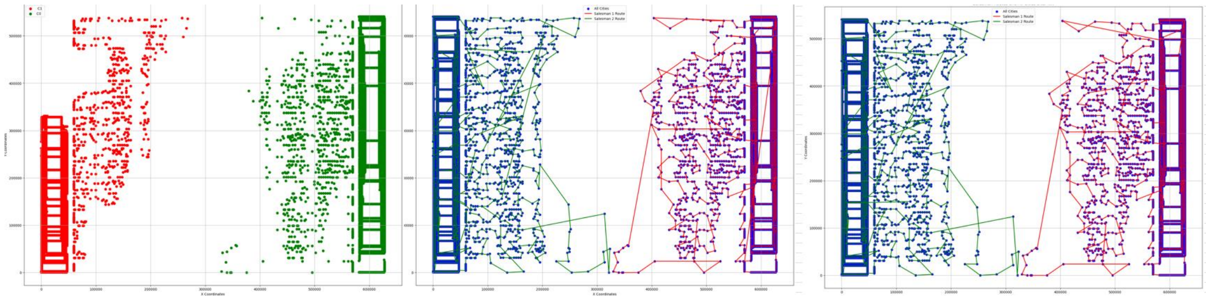
Method: AYTuna method



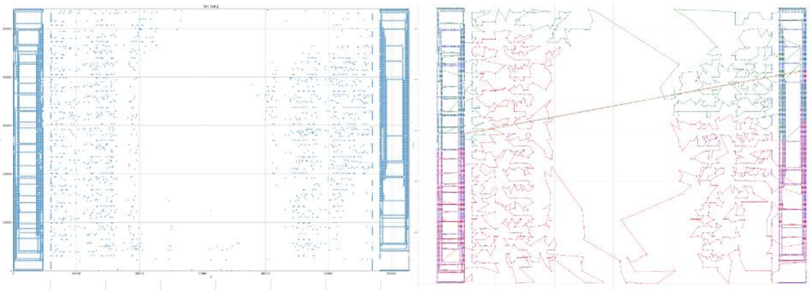
Third Case: 7397 Cities

Method:K-Means + Nearest Neighbors + 2-OPT Method

K-Means KM + NN KM + NN+2-OPT

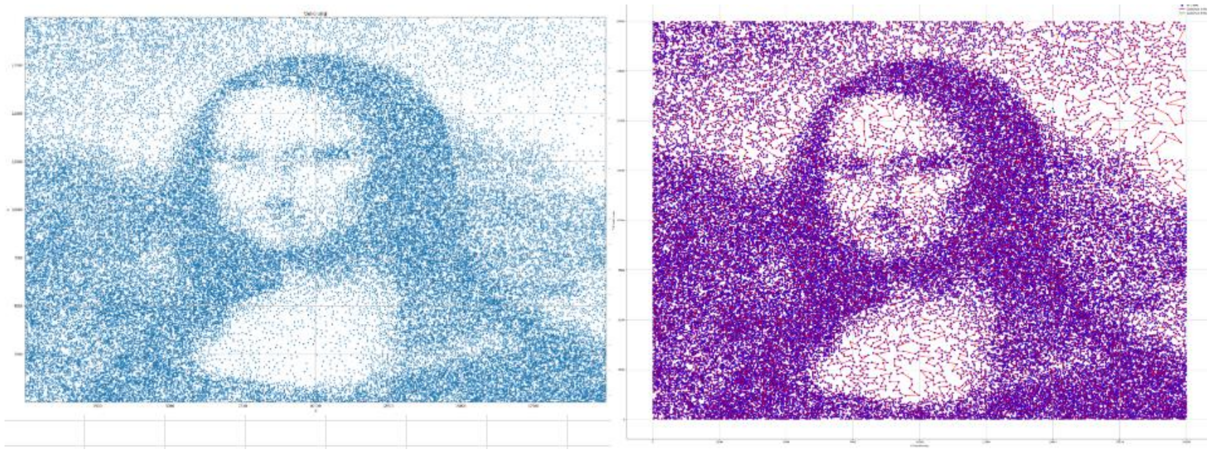


Method: AYTuna method



Fourth Case: 50000 Cities

Method: AYTuna method



References:

1: https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning