CS6380: TSP-1 Visualization using Alviz

V.C.Ashwin
EE16B041
Muqeeth Mohammed
EE16B026

November 25, 2019

1 Introduction

In this module, we have implemented different constructive approaches to the TSP problem and used Alviz 1.0 to visualize the paths. The following approaches were implemented:

- Nearest Neighbours
- Greedy Approach
- Savings Heuristic

2 Constructive Methods of TSP

2.1 Nearest Neighbour Technique

In this method, we select the nearest edge for the current node and add it to the tour. This is done until all the nodes are covered in the tour

2.2 Greedy Approach

We create a list of all the edges in the graph and sort them based on their distances in ascending order. Pick and edge and add it to the tour if there is no cycle formed and the degrees of either of the 2 nodes is not greater than 2, i.e. $degree[i], [j] \leq 2$. Repeat this process until the tour contains $n-1$ edges to form a tree.

2.3 Savings Heuristic

We start by generating $n-1$ tours of length 2 starting from an arbitrarily selected node n_{start}. Then, we perform $n-2$ merges at each stage by removing 2 edges from $n_{start}$and adding an edge to connect the hanging nodes. This is done so as to merge the tours with the maximum savings.

3 Observations

Figure 1: $n = 10$ nodes

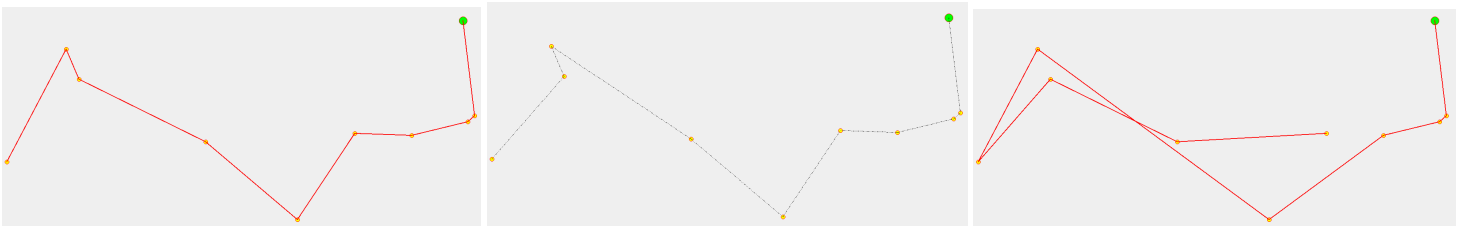


Figure 2: $n = 100$ nodes

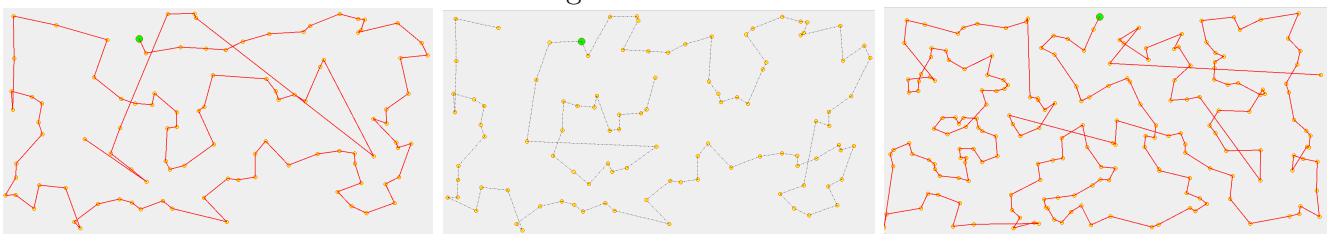
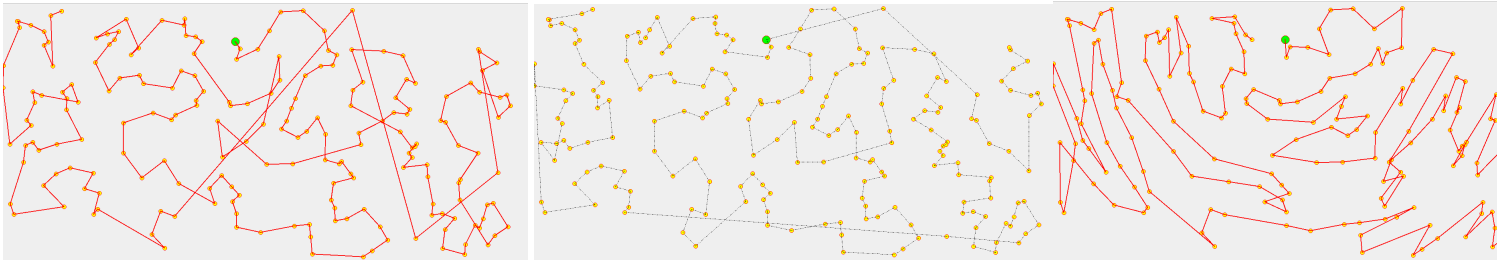


Figure 3: $n = 200$ nodes



The cost for the above tours are as follows:

Method	n=10	n=100	n=200
Nearest Neighbour	1905874	1322878	1871638
Greedy TSP	1930118	948058	2636436
Savings Heuristic	1307738	1075276	1187022

The savings heuristic performs better than other heuristics but it has runtime complexity of $O(n^2)$ for each merge and hence it cannot be used for graphs with n>1000 nodes. The greedy approach performs well than the nearest neighbour as it does not depend on the start node.

Nearest Neighbour has few long edges that are added in the final stages which increases the overall tour cost.