ADVANCED DATA STRUCTURES LABORATORY
IT5412

A MINI PROJECT REPORT

Submitted by

Gowtham Rajasekaran 2022506084
M. Vigna Saktheeshwaran 2022506026
A. Vicky yashwa 2022506049

DEPARTMENT OF INFORMATION TECHNOLOGY
MADRAS INSTITUTE OF TECHNOLOGY
ANNA UNIVERSITY- CHENNAI
CHENNAI - 600 044

Februray / May 2024

Route planning using The Kruskal’s Algorithm.

PROBLEM STATEMENT:

The goal is to calculate the most cost-efficient route through multiple nodes or intersections by minimizing the distance traveled and/or time taken, utilizing Kruskal’s Algorithm.

AIM

To determine the capabilities of the Kruskal’s algorithm in route planning and optimisation.

ABSTARCT

This project aims to explore the application of Kruskal's Algorithm in route planning and optimization to determine the most cost-effective path among several nodes or junctions. By minimizing the distance traveled and/or time taken, Kruskal's Algorithm constructs a Minimum Spanning Tree (MST) that connects all given stations with the least total edge weight.

The implementation involves defining the stations and calculating the distances between them, followed by the application of Kruskal's Algorithm to find the MST. The project also includes generating a visual representation of the MST using Graphviz. This approach demonstrates the efficiency and utility of Kruskal's Algorithm in solving route optimization problems in graph theory.

Challenges faced

01.

SELECTING ALGORITHM

Decision to select an optimal algorithm according to the problem's constraints and requirements was initially difficult. We approached Dr.K.Sankar, (Professor, Department of Mathematics, CEG, Anna University, Chennai) to get an idea selecting an optimal for project implemention. Sir provided guidance on Kruskal's and Prim's algorithms, which helped us determine the optimal algorithm for our specific problem.

02.

ENABLING USER INTERFACE

The project also includes generating a visual representation of the MST using Graphviz. Implementing this feature posed challenges, particularly in integrating Graphviz with the user interface. We used Eye of GNOME (Eog) to display the graph output effectively. This approach demonstrates the efficiency and utility of Kruskal's Algorithm in solving route optimization problems in graph theory.

03.

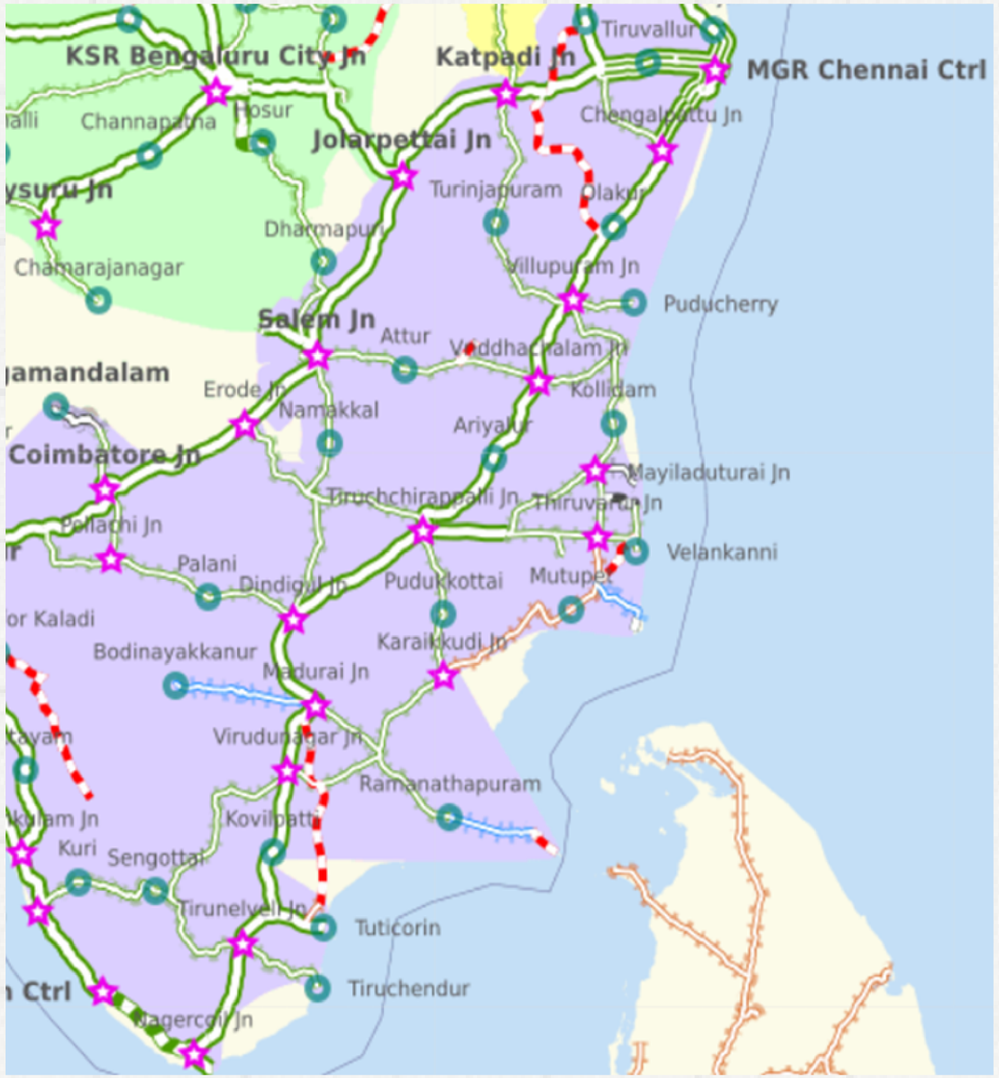
PREPARATION OF CASE STUDY

In calculating the great-circle distance, or orthodromic distance, which represents the shortest distance between two points on the surface of a sphere. This calculation is crucial for determining the most efficient rail routes.

The complexity of integrating geographical data with algorithmic processing was another hurdle. Ensuring precise coordinates and managing the vast amount of data for multiple railway stations required meticulous attention to detail. Additionally, we had to address the computational efficiency of Kruskal's algorithm when applied to such a large dataset, ensuring that the algorithm could handle the extensive calculations without compromising performance.

Our Case Study

We have achieved in finding optimal rail route planning of a part of Southern Indian railways which includes major junctions as nodes. Let us move forward to see the proceedings.



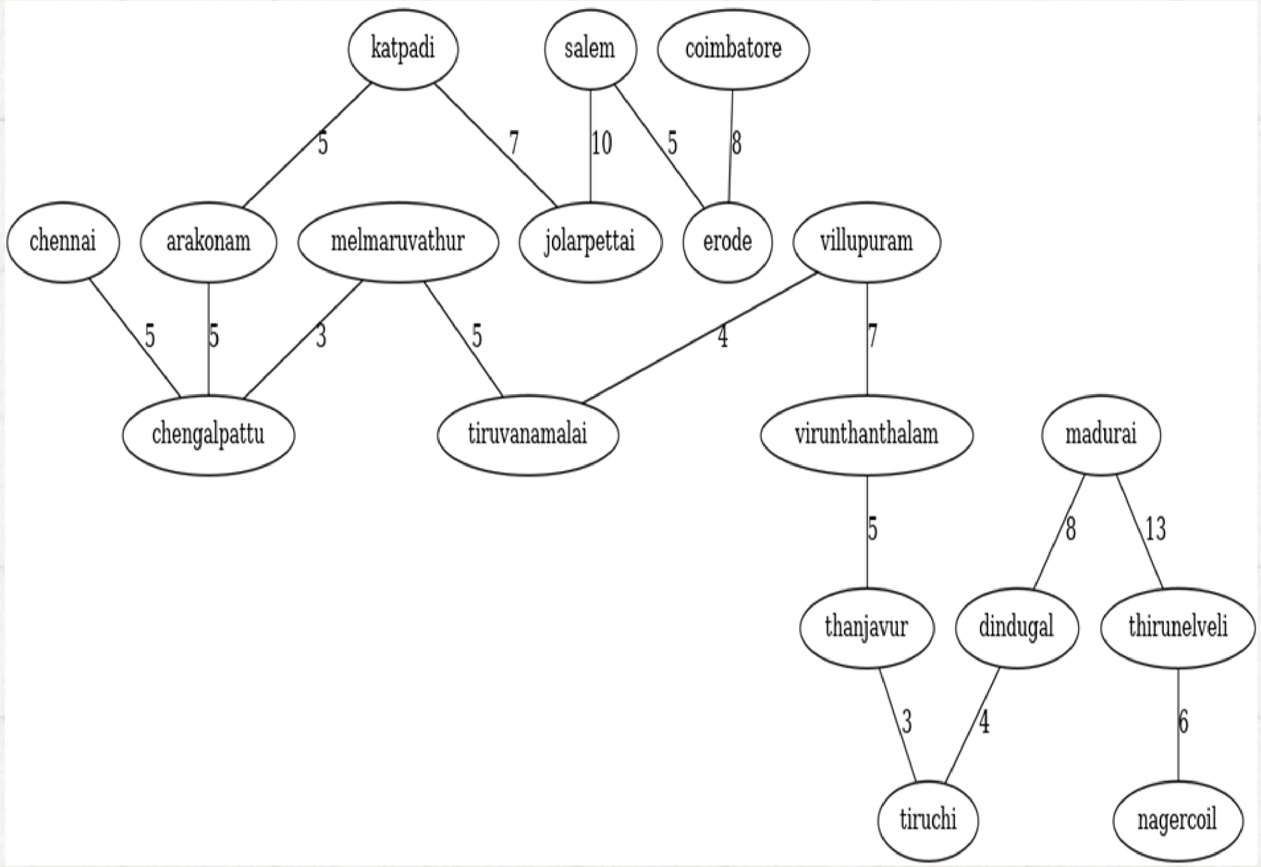
Overview

To determine the optimal rail routes using Kruskal's algorithm. We created Station and WtEdge structures to store station coordinates and weighted edges. Next, we initialized all railway stations, storing their names and coordinates in a vector named stations.

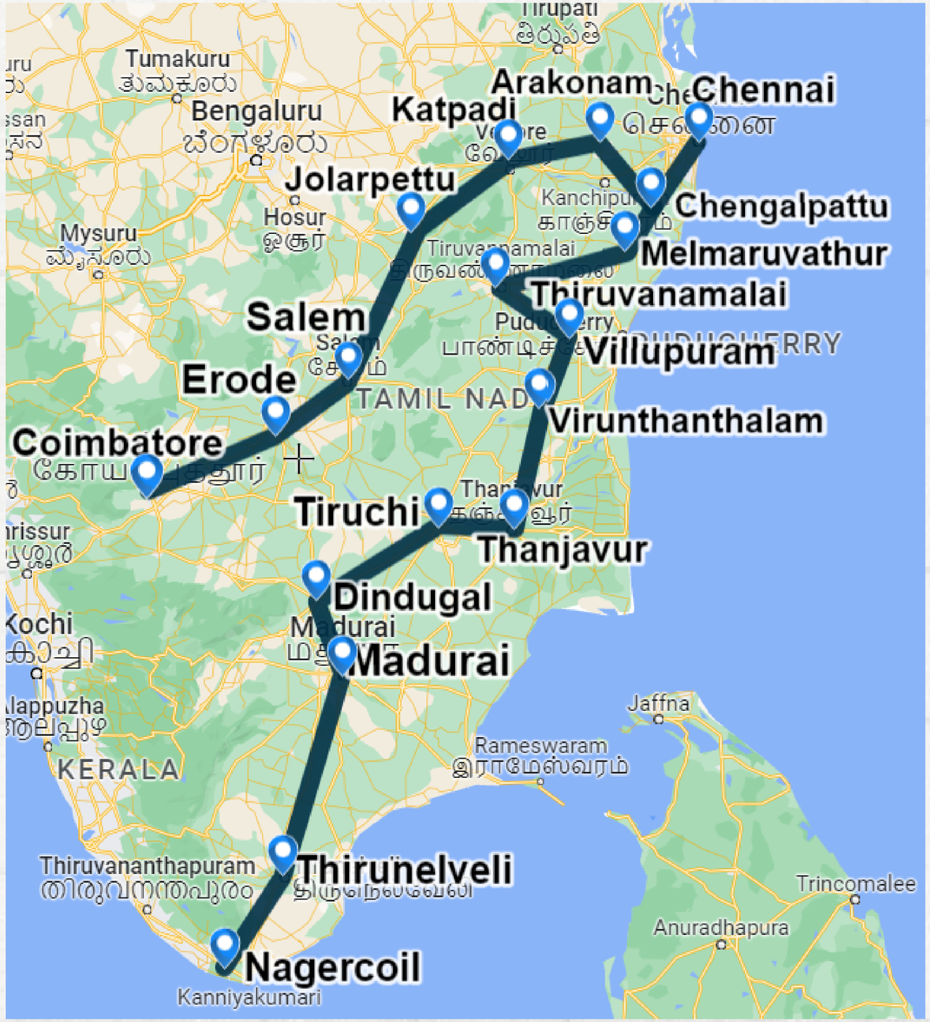
We then calculated the **great-circle distances**, representing the shortest path on a sphere, between all pairs of stations and stored these distances in a vector called distances. Sorting this vector by weight was essential to prepare for Kruskal's algorithm. We implemented functions to find the parent of a node and to union two sets, which helped in **cycle detection** and merging during the algorithm's execution. Kruskal's algorithm was then used to select edges with the smallest weights, avoiding cycles, and forming the Minimum Spanning Tree (MST).

To visualize the MST, we generated a **Graphviz file**. Ensuring the algorithm's performance was optimized to handle large datasets efficiently was a critical part of our process. Finally, we presented the results, displaying station coordinates, calculated distances, and the final MST in a clear, user-friendly manner.

MST Output



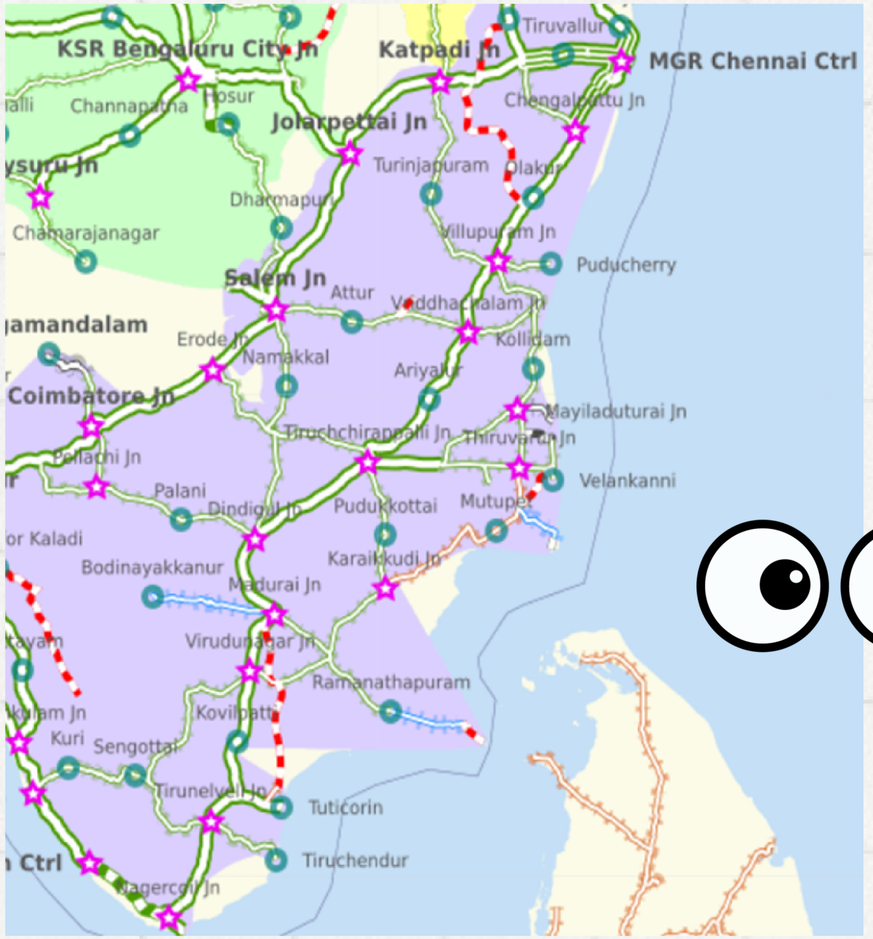
Real time Mapping of route



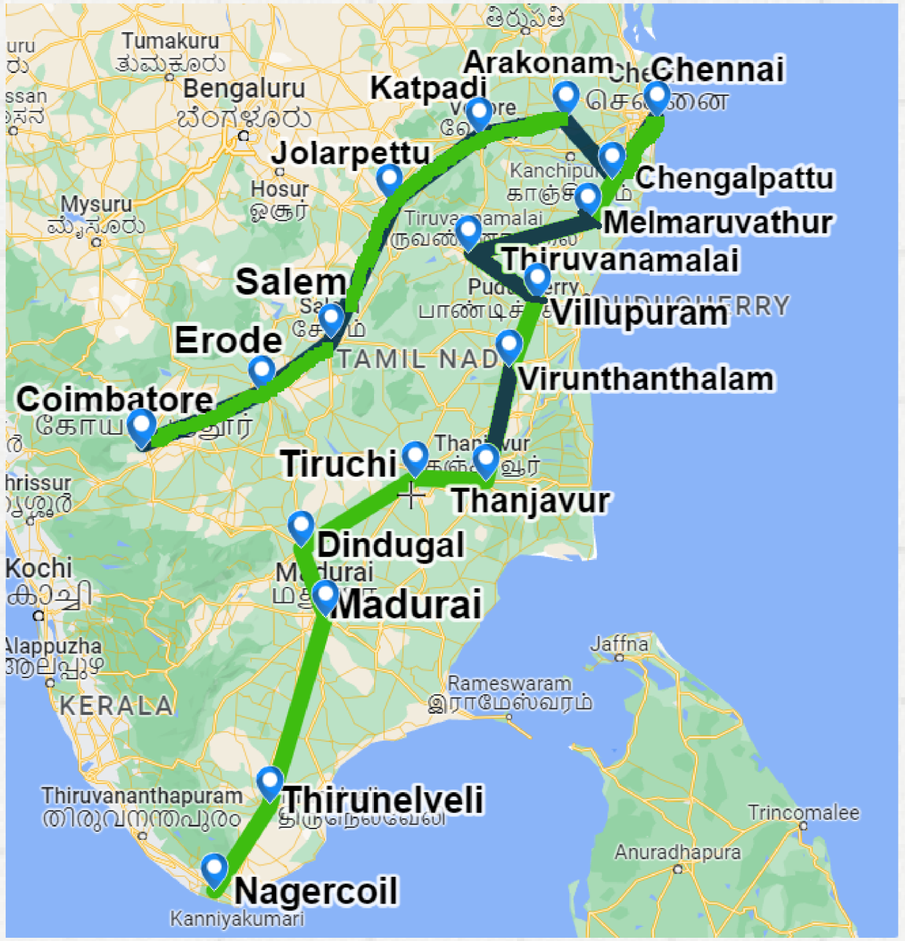
Comparing the efficiency of our project through the case study

Compare only the main lines

South India Rail Route:



Our MST:

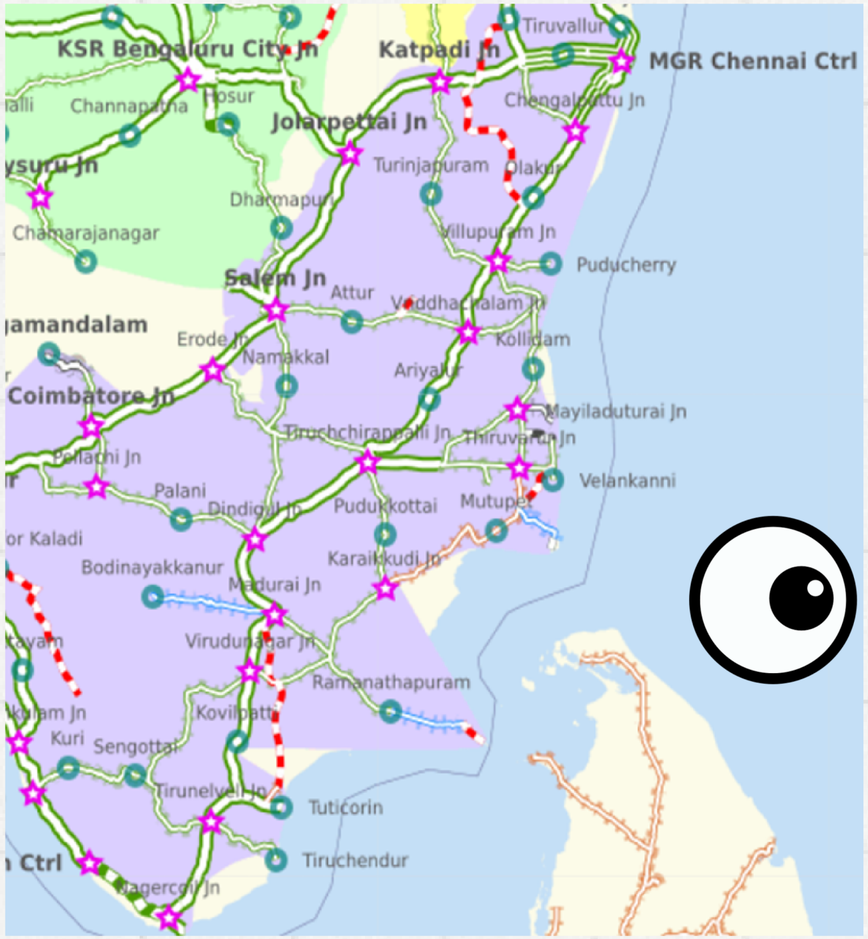


From the comparison of main lines...Match is 81%

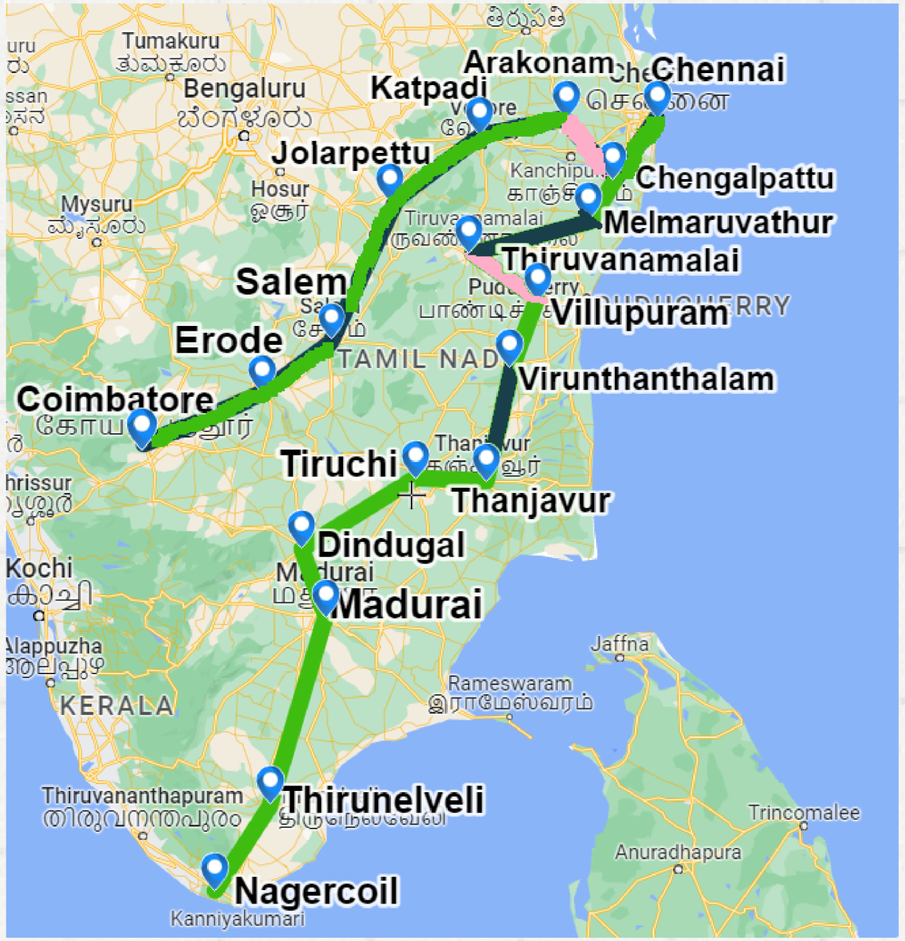
Comparing the efficiency of our project through the case study

Compare the main lines as well as sub-lines

South India Rail Route:



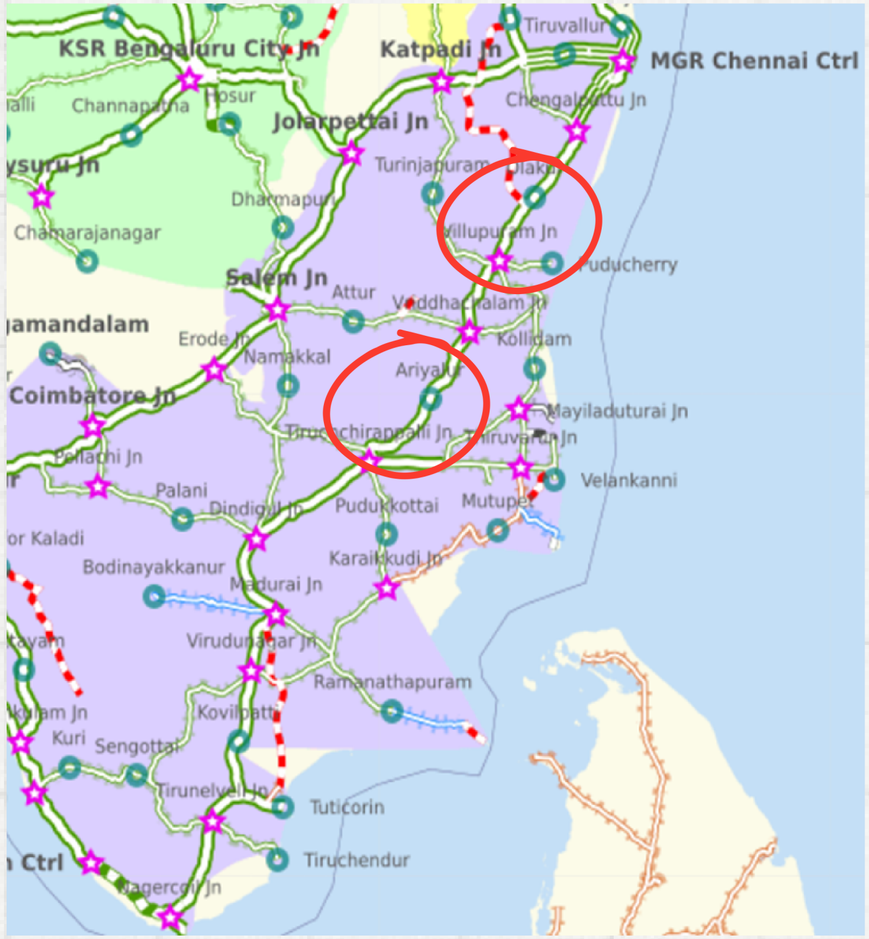
Our MST:



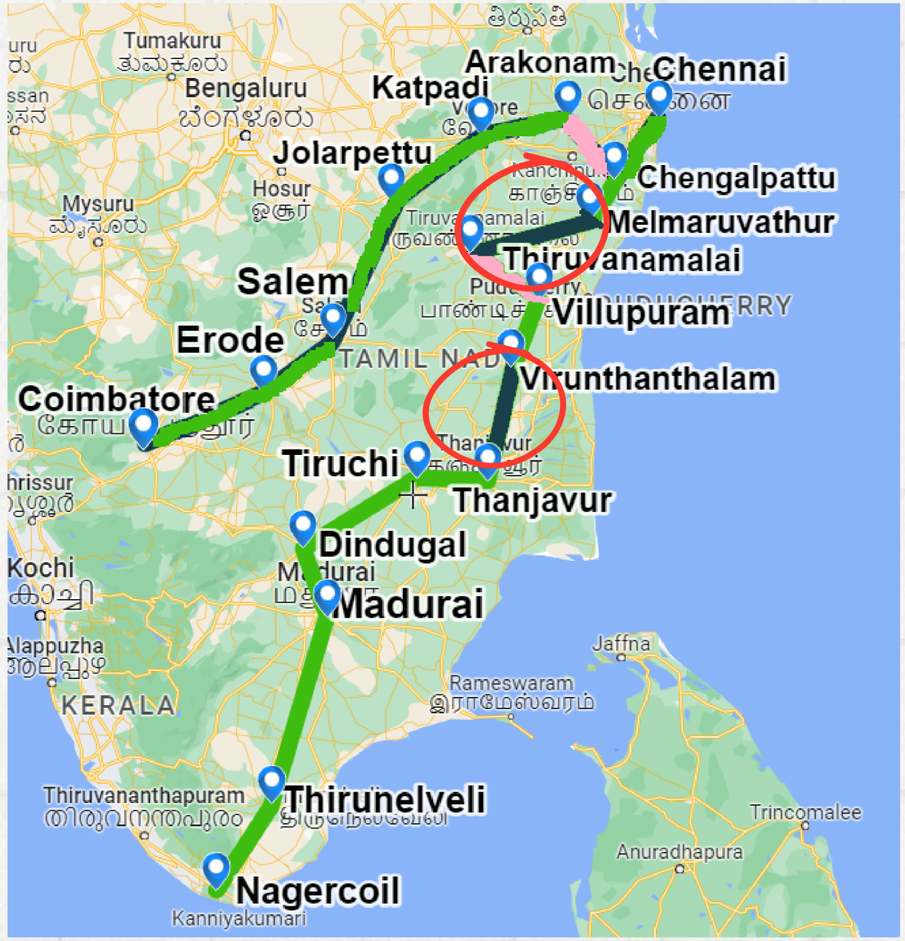
From the comparison of main and sub-lines...Match is 87%

Compare the lines which don’t match

South India Rail Route:



Our MST:



The other factors like...the need to join important cities have took priority than connecting the optimal distance. Since our krushkal algorithm is greedy about finding optimal distances, the above two tracks doesn’t match.

Result

Key outcomes include:

1. **Efficiency:** Kruskal's Algorithm minimized travel distance and time, proving its effectiveness for cost reduction in route optimization.
2. **Scalability:** The algorithm efficiently handled varying numbers of nodes and edges, showing its capability to manage complex networks.
3. **Visualization:** Using Graphviz to visualize the MST provided a clear understanding of the optimized routes, validating the algorithm's results.

Overall, Kruskal's Algorithm is a robust solution for optimizing routes, enhancing operational efficiency, and serving as a basis for future improvements in network management.

References and Online Tools

https://indiarailinfo.com/atlas

https://www.scribblemaps.com

https://graphviz.org/

Thank you very much!

Gowtham Rajasekaran	2022506084
M. Vigna Saktheeshwaran	2022506026
A.Vicky yashwa	2022506049