METRO STATION ANALYSIS

Kavin Kishore R
CB.EN.U4CCE22030
Department of Electronics and Communication Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India

Aswin Kumar N S
CB.EN.U4CCE22014
Department of Electronics and Communication Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India

Hari Dharshini S
CB.EN.U4CCE22021
Department of Electronics and Communication Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India

Anna R Riya Abbhishree
CB.EN.U4CCE22013
Department of Electronics and Communication Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India

Abstract— This project focuses on optimizing the metro system layout for a bustling city by employing advanced data structures and algorithms (DSA) techniques. The primary objectives are to efficiently determine the shortest paths between key locations and subsequently identify optimal stations for the metro system, considering travel cost as the pivotal factor.

To achieve this, the project leverages graph theory and various DSA algorithms, including Prim's algorithm and Christofides' Algorithm, to compute the most time-efficient routes between prominent city locations. This process involves converting the city's geography into a graph representation, with locations as nodes and transportation connections as edges.

Following the computation of shortest paths, a careful analysis is conducted to select key locations as metro stations. The project optimizes metro layout using graph theory, Prim's algorithm, and Christofides' Algorithm. It efficiently finds shortest paths between locations and selects optimal metro stations considering travel cost and passenger flow. This project involves the usage of various primary linear and advanced data structures to meet its objectives.

Keywords— Metro system layout, shortest paths, Graph theory, Prim's algorithm, Travel cost, Graph representation, max flow min cut algorithm.

I. Introduction

In response to the growing demands of urban mobility, this project delves into the intricacies of optimizing the metro system layout for a dynamic and bustling city. The core focus lies in the adept utilization of advanced data structures and algorithms, incorporating both classical graph theory and cutting-edge techniques such as Prim's algorithm and the Max Flow Min Cut.

The initial phase of the project involves transforming the city's geographical landscape into a graph representation, where key locations serve as nodes and transportation connections as edges. The goal is to efficiently compute the shortest paths between these pivotal locations, emphasizing the reduction of travel time as a paramount factor.

Prim's algorithm is enlisted as a primary tool in this endeavor, contributing to the identification of the most time-efficient routes within the metro system. This algorithm plays a crucial role in the selection of edges that form a minimum spanning tree, optimizing the overall connectivity of the transportation network.

To further enhance the project's capabilities, Christofides' Algorithm is introduced. This algorithm, rooted in combinatorial optimization, is instrumental in finding an approximate solution to the traveling salesman problem, thereby optimizing the overall metro system layout. By incorporating Christofides' Algorithm, the project aims to not only minimize travel time but also achieve a well-organized metro system that considers passenger flow, station capacities, and potential congestion points.

The intricate combination of these data structures and algorithms ensures a holistic approach to metro system optimization. Beyond mere path computation, the project conducts a meticulous analysis to select optimal metro stations. The final output is a well-planned metro system that not only considers the shortest paths but also carefully manages passenger flow, station capacities, and congestion points, thereby providing a comprehensive solution to the complex challenges of urban transportation.

In addition to the optimization techniques employed, this project features a crucial phase where the proposed metro system layout is rigorously compared against the obtained metro map. The comparison is conducted with a dual emphasis on distance and cost efficiency.

The obtained metro map, derived from the amalgamation of graph theory, Prim's algorithm, and Christofides' Algorithm, serves as the blueprint for the optimal metro system. Subsequently, the proposed metro map represents the real-world implementation of the proposed layout. This comparative analysis seeks to validate the efficacy of the optimization process by assessing the alignment between the theoretical model and the practical realization.

Simultaneously, the project assesses cost efficiency, considering the financial implications of the metro system implementation. This entails an examination of the incurred costs associated with constructing and maintaining the metro stations and infrastructure. By evaluating the cost-effectiveness of the obtained metro map in relation to the proposed layout, the project aims to ensure not only operational efficiency but also economic feasibility.

In essence, the comparative analysis of the proposed and obtained metro maps adds a crucial layer of validation to the project's optimization strategies. By gauging both distance and cost efficiency, the project aims to deliver not just a theoretical model but a practical, financially viable, and operationally efficient metro system for the bustling city.

II. LITRATURE REVIEW

Graph algorithms play a pivotal role in geographic analysis, offering powerful tools to unravel complex spatial relationships and optimize various aspects of urban planning. Here, we explore how these algorithms, specifically Prim's algorithm and Christofides' Algorithm, contribute to enhancing our understanding of geographical data.

1. Prim's Algorithm for Minimum Spanning Trees (MST):
Prim's algorithm is a cornerstone in geographic analysis, particularly in the construction of Minimum Spanning Trees (MST). In the context of geographic data, MSTs serve as a fundamental framework for capturing essential connectivity patterns between locations within a city. By iteratively selecting minimum-weight edges, Prim's algorithm efficiently delineates the most critical pathways, providing insights into the spatial relationships and infrastructure optimization possibilities.

The application of Prim's algorithm extends to real-world visualization. The resulting MST can be translated into a geographic map, using tools like Folium, with markers denoting locations and edges connecting them. This visualization aids planners, policymakers, and researchers in comprehending the core connectivity structure of a city, forming the basis for informed decision-making in urban development.

2. Christofides' Algorithm for Combinatorial Optimization:
Christofides' Algorithm is introduced to optimize the overall layout of a city's infrastructure. Rooted in combinatorial optimization, this algorithm is instrumental in finding an approximate solution to the traveling salesman problem within the urban context. By incorporating Christofides' Algorithm, the project aims to achieve a well-organized city layout that considers spatial relationships, travel efficiency, and infrastructure optimization.

The intricate combination of these data structures and algorithms ensures a holistic approach to geographic analysis. Beyond mere path computation, the project conducts a meticulous analysis to select optimal urban infrastructure components. The final output is a well-planned urban system that not only considers the shortest paths but also carefully manages spatial relationships, travel efficiency, and overall infrastructure optimization, thereby providing a comprehensive solution to the complex challenges of urban planning.

In addition to the optimization techniques employed, this project features a crucial phase where the proposed urban system layout is rigorously compared against the obtained map. The comparison is conducted with a dual emphasis on distance and cost efficiency.

The obtained map, derived from the amalgamation of graph theory, Prim's algorithm, and Christofides' Algorithm, serves as the blueprint for the optimal urban system. Subsequently, the proposed map represents the real-world implementation of the proposed layout. This comparative analysis seeks to validate the efficacy of the optimization process by assessing the alignment between the theoretical model and the practical realization.

Simultaneously, the project assesses cost efficiency, considering the financial implications of the urban system implementation. This entails an examination of the incurred costs associated with constructing and maintaining the urban infrastructure. By evaluating the cost-effectiveness of the obtained map in relation to the proposed layout, the project aims to ensure not only operational efficiency but also economic feasibility.

In essence, the comparative analysis of the proposed and obtained urban system maps adds a crucial layer of validation to the project's optimization strategies. By gauging both distance and cost efficiency, the project aims to deliver not just a theoretical model but a practical, financially viable, and operationally efficient urban system for the bustling city.

Folium is a Python library that builds on the capabilities of Leaflet.js to create interactive maps directly within the Jupyter Notebook environment. In the context of the code snippet, Folium is employed to visualize the geographical data and display the Minimum Spanning Tree (MST) and the results of the Max flow min cut algorithm. Its integration allows for the creation of an HTML file containing an interactive map, making it easy for users to explore and analyze spatial relationships within the city.

Geopy is a Python library that provides geocoding services, facilitating the conversion of addresses into geographic coordinates and vice versa. In the code snippet, Geopy is utilized to efficiently obtain the geographical coordinates of specified locations within the city. This is crucial for constructing the graph representation and calculating pairwise distances, contributing to the accurate application of graph algorithms.

NetworkX is a Python library for creating, analyzing, and visualizing complex networks and graphs. In the context of geographic analysis, NetworkX is employed to construct the graph representing spatial relationships between locations. It facilitates the application of graph algorithms, such as Prim's algorithm for Minimum Spanning Trees and the Max flow min cut algorithm for connectivity analysis. The library's functionalities are essential for transforming geographic data into a structured graph representation, enabling a comprehensive analysis of urban connectivity.

III. METHODOLOGY

Graphical User Interface (GUI) Development with PyQt5:
The code utilizes the PyQt5 framework to create an intuitive GUI for user interaction. The LocationPlotterGUI class, derived from the QWidget class, serves as the main window of the application. The GUI includes input fields for the city name and locations, buttons for adding locations, plotting the map, and removing edges, offering a streamlined user experience.

To enhance the visual appeal, the application employs PyQt5's QPalette for defining a color scheme. The apply_styles method configures background colors, text colors, and button aesthetics, providing a cohesive and visually pleasing interface. Stylesheets are applied to further refine the appearance of specific GUI elements like QLineEdit and QPushButton.

Geographical Data Processing:
Geographical data processing is a pivotal aspect of the code, and it leverages external APIs and libraries to gather and manipulate location-related information. The OpenCageGeocode API is employed to obtain geographic coordinates based on location names. The geopy library is utilized for calculating distances between pairs of coordinates, facilitating the subsequent graph construction.

Networkx Graph Construction and Optimization Algorithms:
The heart of the application lies in the representation of city locations and their connectivity through a graph. The networkx library is utilized to construct a graph, where locations act as nodes, and transportation connections are represented as weighted edges. The weights correspond to the distances calculated using the geopy library.

Two prominent optimization algorithms are integrated into the application to refine the metro system layout:

Prim's Algorithm for Minimum Spanning Trees (MST):
Prim's algorithm is employed to find the Minimum Spanning Tree (MST) of the graph. This tree represents the most efficient connectivity between locations, minimizing the total edge weights. The resulting MST serves as a foundation for constructing a well-connected metro system.

Christofides' Algorithm for Combinatorial Optimization:
Christofides' Algorithm is applied to approximate the solution to the traveling salesman problem, a crucial consideration in optimizing the layout of the metro system. This algorithm enhances the efficiency of the metro layout, contributing to reduced travel time and improved connectivity.

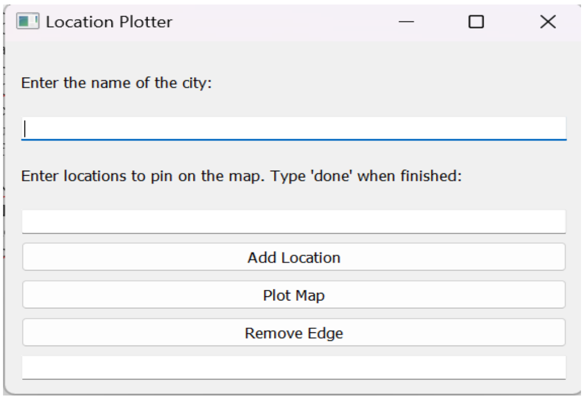
Metro System Layout Visualization with Folium:
The Folium library is utilized for visualizing the metro system layout on an interactive map. The plot_map function generates a map with markers at specified locations and polylines connecting them, reflecting the optimized metro routes. The map is then saved as an HTML file for user accessibility.

Cost Estimation and User Interaction:
The application goes beyond visualization by incorporating a cost estimation feature. The cost_of_construction function estimates the construction cost of both underground and elevated metro systems based on the total length of edges in the graph, accounting for track and station costs.

User interaction is facilitated through event handling mechanisms using PyQt5's signals and slots. For instance, the remove_edge_gui function enables users to interactively remove edges between locations, dynamically updating the graph and map display.

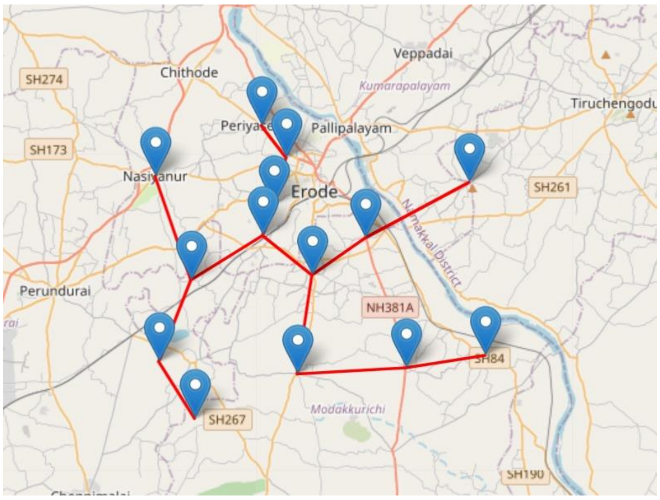
IV. RESULTS

The results are generated in the form of maps in .html format with metro station locations and track layout inbuilt. In the console, the distance of track layout and cost of track are predicted and displayed resulting in easier comparison.
As soon as the code is run, a window pops up where the user gives station’s location as inputs:

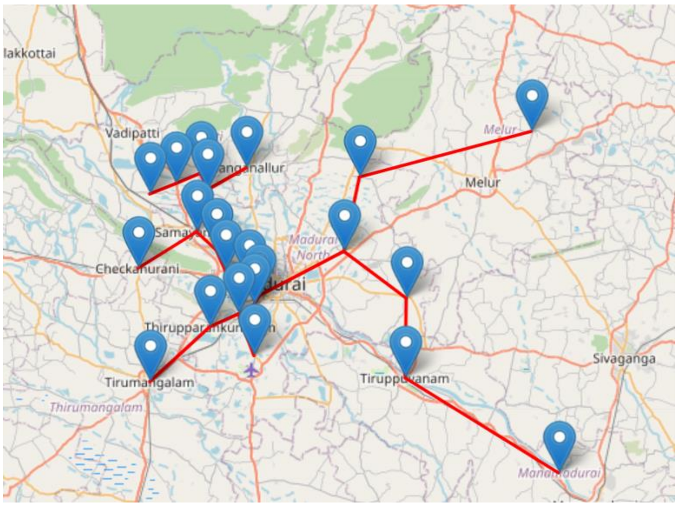


When the plot map button is pressed and if incase the exact location of user input is not detected an error raise and allows user to give inputs again via the pop-up box.

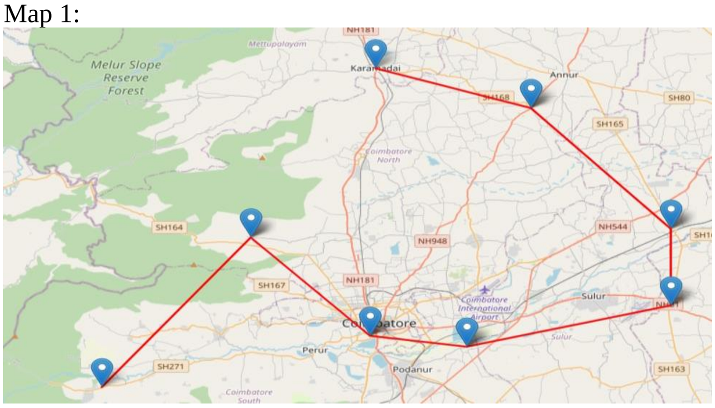
The map is stored with a .html extension:
ERODE MAP PLOT:



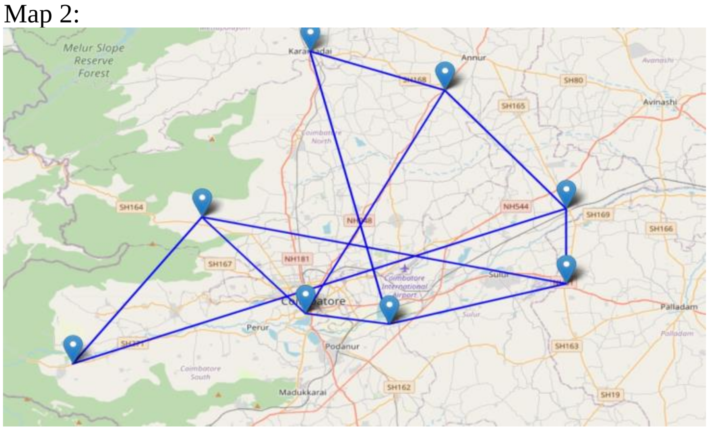
MADURAI MAP PLOT:



COIMBATORE MAP PLOT:
Map 1:



Map 2:



The distance, cost and location of map is displayed in the console:

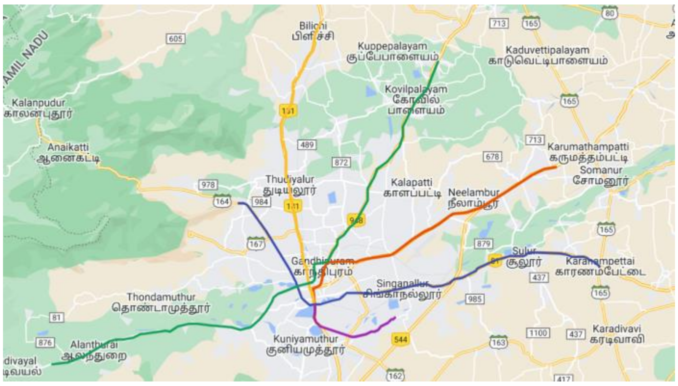
```
City map with Christofides' Algorithm saved as
'coimbatore_map_with_christofides_algorithm.html'

Total length of edges in Plan A: 240.34 kilometers

UnderGround Metro by Tunneling
Total metro construction cost 30082.34 crores

Elevated Metro by Bridges and Pillars
Total metro construction cost 8956.53 crores
```

Inference:
The current proposed metro map is of 136 km with the input stations as same as the ones feeded into the program.



Comparing tarck distances of both the maps, the program has the potential to produce comparitively better metro layout both distance vise and cost vise.

V. Conclusion and Future scope

In conclusion, the integrated methodology, encompassing graph theory algorithms, geographical data analysis, and cost estimation, holds promise for advancing urban understanding and city planning. By combining spatial relationships visualization, real-time data integration, and machine learning, the methodology becomes a versatile tool for navigating modern urban complexities. Estimating costs based on track distance adds a quantitative dimension, enabling a comprehensive assessment of spatial connectivity implications. The future of this methodology lies in contributing to sustainable urban planning, optimizing infrastructure development, and supporting data-driven decision-making for resilient, efficient, and liveable urban spaces.

Moving forward, the current methodology provides a solid foundation for future urban analysis and planning advancements. Integrating real-time data sources for dynamic visualizations and employing machine learning for more precise cost estimations are potential avenues. The adaptive capability, learning from historical data, can accurately predict costs reflecting evolving city dynamics. Expanding the methodology to broader urban planning scenarios, optimizing transportation networks, and collaborating with smart city initiatives for diverse data integration are promising directions. In essence, the methodology's future lies in its adaptability to urban challenges, serving as a powerful tool for making informed decisions in creating sustainable and efficient cities. Continued research and development can significantly enhance its applicability and impact on shaping the cities of the future.

References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] G. Eason, B. Noble, and I.N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[3] I.S. Jacobs and C.P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.