AI BASED TRIP PLANNER (SAFAR KA SAATHI)

Chandana C[1], Shubham Mittal[2], Saloni Kumari[3], Shivangi Singh[4], Vinod S[5]

[1] Assistant Professor, Department of Computer Science Engineering, SJBIT, Bengaluru, 560060
[2,3,4,5]Student, Department of Computer Science and Engineering, SJBIT, Bengaluru, 560060

***Abstract* — Trip planner applications are developed using the capabilities of advanced computing and the accessibility of cellphones. They have quickly gained popularity as a tourism tool due to their versatility. Customers can plan a trip with little effort and a few simple clicks. We are also incorporating social components, such as the opportunity to create a vacation club to discuss spending and images. Tourists face a number of challenges, including the inability to rely on local guides due to their ties to local businesses and the risk of being overcharged. The Traveling Salesman Issue (TSP) is an NP-hard problem that will be extensively used to discover the best route for our tour. There are various map APIs available, including Google Maps and Azure Maps. We will, however, use the Google Maps API to collect the distance matrices of the user's preferred sites, and then apply our algorithm to this matrix to produce a viable route.**

***Keywords* — NP-hard problem, Trip planner, Salesman Problem, Traveling, suitable route**

I. INTRODUCTION

A trip planner is a tool used to organize trips. Organizing a trip requires researching fascinating places to visit, finding convenient lodging, and finding eateries.

It only takes a few mouse clicks. Users merely need to enter the city they want to visit. Based on previous user evaluations and ratings, a user will be given with a list of all tourist destinations. You must choose your preferred lunch or dinner time. Choose the places you want to see. Our system will automatically construct a route and timetable for the full day with a simple click. You will be given a selection of neighboring eateries for lunch and dinner so that you do not have to travel too far to eat well.

For our online application's social feature, a user can post his journey in order to find new travel companions, and they can speak in a group chat. This group chat will also allow you to share high-resolution photos. The Split Smart app, which allows users to share and track holiday expenditures, inspired us. To offer a holistic vacation experience, the same feature will be added to our app.

We intend to develop an app that meets all of a traveler's needs, eliminating the need for them to move between apps. They can instead use a one user-friendly app.

The Traveling Salesman Problem is the focus of this investigation. TSP is an NP-hard problem that will be heavily used to select the best tour route. We'll rely extensively on map APIs like Google Maps and Microsoft Azure Maps. We will collect distance matrices for the user's selected sites using the Google Maps API, then apply our algorithm to that matrix to produce an optimal route. It will also be used to locate restaurants along the proposed path. When the travel begins, the customer will be shown with hotels that fit their budget. When they start their journey, we will display their current location, as well as the desired route and durations, using Google Maps' map user interface feature and adding real-time traffic data.

As the number of online social networks and communication networks has grown in popularity, there has been a surge in interest in graph data management and graph mining research in recent years. Dynamic graphs have been studied by some scholars as a sequence of updates to static graphs. Previous research has primarily focused on static graphs. Many graphs in the real world, on the other hand, are temporal graphs, in which one vertex communicates with another at specified times.

Indeed, temporal graphs are frequently turned into static graphs because their static counterpart is significantly easier to handle [4].

The highly connected components (SCCs) of a static network, for example, may be computed in linear time, but there is no known polynomial-time method for computing the SCCs of a temporal graph. When a temporal graph is transformed to a static graph, all time information that is necessary for comprehending the interactions within the graph is lost. Furthermore, the resulting static graph usually contains incorrect information, leading in severe errors in graph or object connection comprehension.

The travelling salesman problem is determining the shortest path to visit each city and return to the starting point given a collection of cities and the distance between each pair of towns. The issue is straightforward, but its resolution is tough. The Traveling Salesman Problem is NP-hard having a large search space that cannot be solved in polynomial time. Many researchers are currently looking into the topic of travelling salespeople. Vehicle routing, microchip manufacture, GSM packet routing, and drilling in printed circuit boards are some of its applications.

Informally, a temporal graph is a graph that changes throughout time. Information and communication networks, social networks, transportation networks, and a variety of physical systems can all be simply defined as temporal graphs. This category includes almost every

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

network with a dynamic topology. The majority of modern communication networks are essentially dynamic, including mobile ad hoc, sensor, peer-to-peer, opportunistic, and delay-tolerant networks.

II. RELATED WORK

Several studies and projects have already been completed to ensure a comfortable ride. And getting information from a human has improved the experience.

Temporal Algebra Graph [1]: These representations are utilized in this study to capture the structure of networks ranging from the Internet to personal associations, roadways, sensors, and metabolic processes. While static graph analysis is a well-established topic, there is a growing emphasis on understanding and portraying network dynamics over time.

Traveling salesman problems in temporal graphs [2]: To exemplify the concept of time, several well-known combinatorial optimization problems are used. We focus on problems that are represented by temporal graphs. A temporal graph D = (V, A) can be thought of as a series of static graphs G1, G2,..., Gl superimposed on this (static) collection of nodes V. At time t, every Gt = D (t) = (V, A (t)) is referred to as a D instance, and l is D's lifetime. Our main focus is on temporal graph analogues of travelling salesman problems.

Review of Literature on the Traveling Salesman Problem [3]: The Traveling Salesman Problem is a well-known combinatorial optimization problem with a simple formulation but a challenging solution.

From an algorithmic approach, a primer on temporal graphs [4]: In this paper, we look at the most recent computer science research on temporal graphs and temporal graph difficulties.

Dynamic graph deep learning with temporal graph networks [5]: Because of their ability to learn complicated networks of interconnections or interactions, Graph Neural Networks (GNNs) have lately gained importance. These can be found in a wide range of applications, including particle physics and biology, as well as social networking sites and recommendation systems.

Time allocation to minimize temporal graph reachability [6]: We provide a novel way for altering temporal graphs in this paper: the number of active periods associated with each edge remains constant, but the relative order of active edge sets can be changed.

Time-Dependent Graph Applications and Algorithms [7]: A time-dependent graph, informally, is one whose structure changes over time. The weights associated with edges in such networks change dynamically over time, indicating that the edges are activated by a sequence of time-dependent items.

Path Issues in Temporal Graphs [8]: We demonstrated the importance of temporal pathways in temporal graph analysis by comparing shortest - path in static graphs to temporal pathways in temporal graphs.

III. METHODOLOGY

Breadth-first search (BFS):

It is an algorithm for exploring or traversing graph or tree data structures. It begins at the tree's root and investigates all neighboring nodes at the current depth level before proceeding to the nodes at the next depth level.

For instance:

The following is an example of the breadth-first tree created by performing a BFS starting with Frankfurt on German cities:

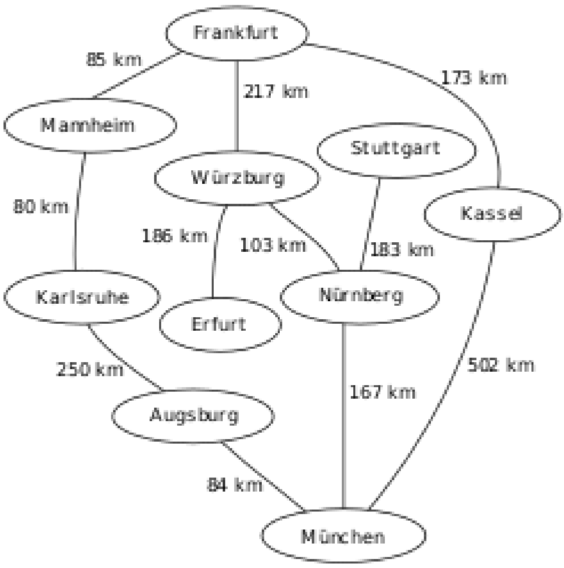

Figure (1): An example map of Southern Germany with some connections between cities

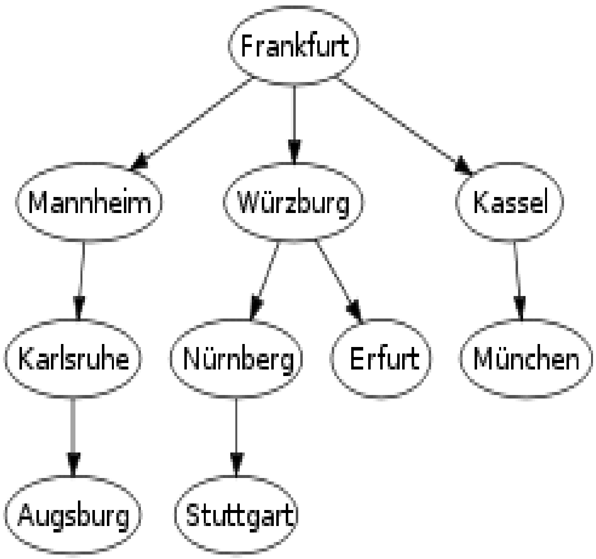

Figure (2): The breadth-first tree obtained when running BFS on the given map and starting in Frankfurt

Traveling Salesman Problem:

It is a reference to a well-known issue in the field of computer science. A salesman on the road visits n different cities that are all connected to one another (weighted graph). It is necessary for him to plot a route that allows him to stop in each city exactly once while still completing the trip in the shortest amount of time possible. In most cases, the beginning and ending nodes of a circuit are the same, which results in a closed loop.

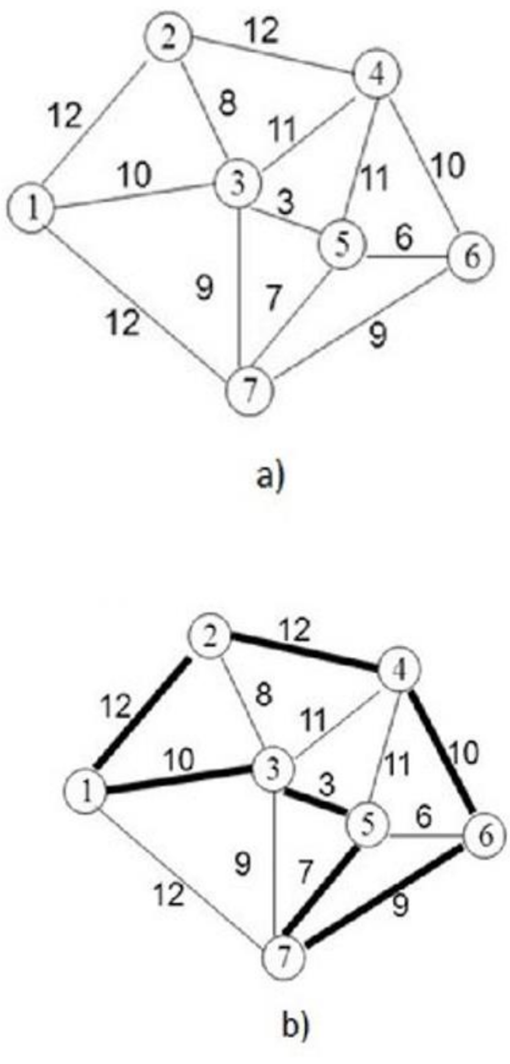

Figure 4: TSP and it's solution

IV. RESULT

In this paper, travelling salesman problem will be used to find the best suitable path for our tour. The simplest solution is to try all the permutations of the paths (brute force method) and then select the best path, but it would be extremely inefficient as it will have a time complexity of O(n!) . So, the algorithm we will be using is the heuristic algorithm to solve Travelling salesman problem and it will give an approximate solution (as it is an NP-hard problem) which will be good enough for our use case.
Like the rule set of the first depth search, the width search also traverses the nodes in a graph. It will start at the root node, which was chosen earlier, and then traverse all its neighboring nodes. Each adjacent adjacent node is scanned respectively until the entire graph is traversed. This search mode is the quality used for algorithms which must always choose the viable high quality path. Applications of this form of search include finding the shortest direction between two nodes, demonstrating whether a graph can be split into two components, finding the minimum expansion tree in an unweighted graph, a web crawler, and GPS navigation structures to find nearby places. For example, a GPS navigation car uses the BFS algorithm to track a journey. The traveler points of interest you need to go to are the connected nodes based on their distance from each of them, and the BFS algorithm plans the shortest ride to each of these attractions.
In this framework, the User gets the most limited or closest way of the course entered by the client. The area is set apart on the incorporated google map. The client needs to enter the beginning and end area for the objective, and the client needs to add five unique stops where he needs to end or visit during the outing. The client will get the most limited way by utilizing the calculation utilized in-application.
You can invest less energy accomplishing administrator work and other monotonous undertakings via computerizing them with programming. The time saved can be better spent chipping away at getting new business and other income producing assignments.

V. CONCLUSION

Based on the information presented in the table above, it is possible to draw the conclusion that the vast majority of travel apps do not offer fully customizable itineraries. Furthermore, even if the trips are fully customizable, the itineraries are still pre-planned and cannot be optimised for speed if they are fully personalised. The vast majority of trip planning applications are unable to produce fully customizable and ideal itineraries; hence, this was the fundamental problem that "bon voyage" answered. The secondary applications enable users to avoid using several apps by acting as an all-in-one app.

VI. REFERENCES

[1] Shabnam Sangwan, Literature Review on Traveling Salesman Problem ,Published on : June, 2018

[2] Othon Michail, An Introduction to Temporal Graphs: An Algorithmic Perspective , Published on : 27 Feb, 2019

[3] Othon Michail and Paul G. Spirakis, Traveling Salesman Problems in Temporal Graphs, Published on March, 2019

[4] Esther Oh, TravelMind — Travel With A Peace Of Mind (UX Case Study), Jan, 2019, Accessed on: Mar. 1,2021, [Online], Available: https://blog.prototypr.io/ux-case-study-travelmind-travel-with-a-peace-of-mind-643a816af8db

[5 Emanuele Rossi, Temporal Graph Networks for Deep Learning on Dynamic Graphs, Published on: 9 October, 2020

[6] Maharshi J. Pathak and Ronit L. Patel, Comparative Analysis of Search Algorithms, Published on: June, 2018

[7] Dickinson, J.E., Ghali, K., Cherrett, T., Speed, C., Davies, N., Norgate, S. Tourism and the smartphone app: capabilities, emerging practice and scope in the travel domain., Current Issues in Tourism., Published on: 2014

[8] Silu Huang, James Cheng, Huanhuan Wu, Temporal Graph Traversals: Definitions, Algorithms, and Applications, Published on: 9 Jan, 2014

[9] John Tang, Ilias Leontiadis, Salvatore Scellato, Vincenzo Nicosia, Cecilia Mascolo, Mirco Musolesi, and Vito Latora , Applications of Temporal Graph Metrics to Real-World Networks, Published on: 29 May, 2013

[10] Huanhuan Wu, James Cheng, Silu Huang, Yiping Ke, Yi Lu, Yanyan Xu, Path Problems in Temporal Graphs

[11] Xiaoli Sun, Yusong Tan, Qingbo Wu, Jing Wang and Changxiang Shen, New Algorithms for Counting Temporal Graph Pattern, Published on: 20 September 2019

[12] Yishu Wang, Ye Yuan, Yuliang Ma, Guoren Wang, Time-Dependent Graphs: Definitions, Applications, and Algorithms, Published on: 25 September 2019

[13] Vera Zaychik Moffitt, Julia Stoyanovich, Temporal Graph Algebra, Published on: September 2019

[14] Jessica Enright, Kitty Meeks, Fiona Skerman, Assigning times to minimise reachability in temporal graphs,Published on: 20 August 2020"