REAL TIME ROUTE RECOMMENDATION SYSTEM

Yam Chen Xi TP061635

Ng Wen Xuan TP060653

Abstract

This document discusses on the critical issue of traffic congestion in urban areas, specifically focusing on Klang Valley, Malaysia. The study reviewed the contexts of the existing route recommendation applications from various research, particularly on their ability to provide real time recommendations as well as adaptation to a dynamic traffic condition. This research presents a novel approach by designing and implementing a real time route recommendation system to assist motorists to seek the optimal routes between 50 shopping complexes in Klang Valley. A road network is simulated with real time updates of motorists' data. The proposed system not only enhances travel efficiency by providing diverse routes but also ensures robust performance metrics, including but not limited to attributes such as execution speed and system response time. A threshold was established to define real time responses while emphasizing the potential for a revolution in urban transportation.

Introduction

Traffic congestion is a common issue in many urban places around the world, especially in developing countries like Malaysia. It causes different negative impacts such as air pollution, fuel consumption, travel delay, and stress for drivers and passengers. According to the TomTom Traffic Index (2022), Kuala Lumpur ranked as the fifth most congested city in Southeast Asia in 2022, with an average of 53 minutes spent in traffic jams every day. One of the main factors contributing to traffic congestion is the high demand for private vehicles, which is driven by the lack of efficient and reliable public transportation systems. Therefore, many urban motorists rely on route recommendation applications to help them navigate through the complex road networks and avoid traffic jams.

However, most of the existing route recommendation applications have two major limitations (Wang et al., 2014). First, they do not provide real-time route recommendations that can adapt to the dynamic and unpredictable traffic conditions. Instead, they use historical or static data to estimate the travel time and distance between different locations, which may not reflect the recent situation on the road. Second, the applications tend to recommend the same route for all users who have the same origin and destination, which may result in more congestion on the recommended route. Therefore, there is a need for a more intelligent and personalized route recommendation system that can provide real-time and diverse routes for different users based on their preferences and constraints.

The aim of this research is to design and develop a simulated best route recommendation system that can run in real-time and provide optimal routes for motorists who want to travel between different shopping malls in Klang Valley, Malaysia. The system will use a simulated road network data that consists of 50 nodes representing shopping malls and edges representing the traffic conditions between them. The system will also collect, process, and update real-time traffic data periodically from the driving data sent by many simulated motorists concurrently. The system will be implemented in C language while utilizing FreeRTOS as its real time operating system handler. The system should recommend the best route for each user based on their preferences and constraints, such as time, distance, and popularity of location.

The contribution of this research is twofold. First, it will propose a novel approach for real-time route recommendation that can handle multiple concurrent queries from different users and provide diverse routes that can reduce traffic congestion and improve travel efficiency. Second, it will evaluate the performance of the simulated system in terms of execution speed, response time, resource efficiency, and maximum throughput.

The rest of this paper is organized as follows. Section 2 reviews the related literature on route recommendation systems and their challenges. Section 3 describes the design and implementation of the simulated best route recommendation system. Section 4 presents the results and analysis of the performance evaluation of the system. Section 5 discusses the implications and limitations of the research. Section 6 concludes the paper and suggests future directions for improvement.

Literature Review

Real-time route recommendation systems are growing increasingly prominent because of the escalating urban traffic congestion globally. With an emphasis on the Kuala Lumpur environment, this literature review analyses current studies on these systems and examines their role in offering effective real-time congested-aware route recommendations for urban commuters.

Urban traffic congestion has become a critical issue with the growth of economies, as identified in prior research by Wang et al. in 2014. To address this challenge, route recommendation systems have been introduced as potential solutions. However, these systems suffer from limitations such as recommending similar routes for all users without considering real-time traffic data adequately. The dynamic and unpredictable nature of traffic conditions and delays in updating data pose significant obstacles to the development of an efficient real-time route recommendation system. Accurate route estimations are more difficult to achieve when there is a longer time gap between current and projected traffic conditions.

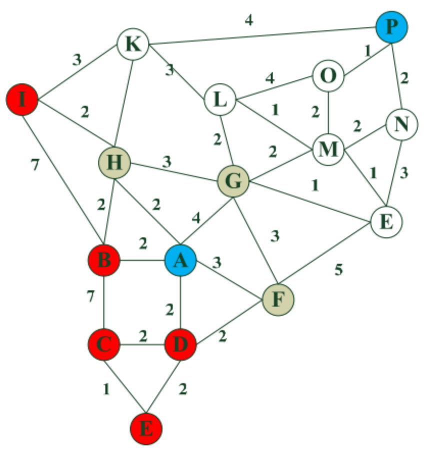

Figure 1: Real Time Routing Proposed

To address the concerns, The researchers have introduced R3, a real-time route recommendation system. R3 utilizes taxi data to incorporate current traffic information and tailor high-quality route recommendations for individual users to address the concerns. Unlike conventional systems, R3 offers multiple route options with the objective of reducing congestion and minimizing travel time. The system operates through a two-phase algorithm consisting of an expansion phase and an estimation phase, enabling real-time recommendations based on present traffic conditions. Implementation of R3 has yielded favourable results using historical taxi driving data to predict a practical traffic model. It employs an efficient tree-based index structure known as G-Tree to compute optimal routes as shown on Figure 1. Integration of multiple G-Tree for different time intervals were demonstrated to allow real time route recommendations within a specific interval. Empirical evidence consistently demonstrates that compared to Google Maps and Baidu Map, R3 outperforms them by providing greater time savings and improved accuracy in routing decisions. This underscores how R3 holds promise as it seeks to mitigate traffic congestion while enhancing overall driving experiences (Wang et al., 2014).

In a study conducted by Hendawi et al. in 2017, the researchers identified several challenges in providing personalized routing services in smart cities. These challenges include accurately capturing and representing road network characteristics, managing map updates, and efficiently processing personalized queries without compromising system scalability.

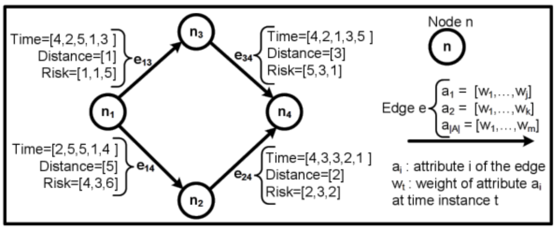

Figure 2: Attributes Time Aggregated Graph (ATAG)

To tackle the challenges of personalised route planning, PreGo presents a unique method that integrates diverse geo-tagged information with real-time road network data. This makes it possible to generate dynamic representations of the road network that account for different cost elements throughout the span of the day. PreGo makes a significant contribution by introducing the Attributes Time Aggregated Graph (ATAG), a versatile model for representing road networks. ATAG enables immediate searches for the best routes based on various preferences and time requirements. PreGo provides the Time-Parameterized Multi-Preference Shortest Path (TP SP) algorithm to further improve this capacity. This algorithm can find all feasible routes via the road network graph in a single run utilising user-defined parameters related to preferences and time constraints (Hendawi et al., 2017).

Furthermore, PreGo introduces the bidirectional TPSP algorithm to minimise unnecessary branching and accelerate response times. PreGo additionally employs the Best-start TP SP algorithm, which recommends the best time to start a trip after taking into consideration user preferences. Dynamic data sources and innovative algorithms that take into consideration user preferences are utilised to accomplish this. PreGo is noteworthy for integrating real-time data by retaining two versions of the ATAG structure, one for past information and one for real-time changes. The result makes it possible to offer up-to-date route recommendations according to the current condition of the roads (Hendawi et al., 2017).

Song and Jiao (2023) presented a real-time intelligent recommendation algorithm for cross-regional city-level tourist routes while considering epidemic normalization. The algorithm introduces multi-time scale constraints to enhance the capability of real-time recommendations. They generated a tourist correlation model for the routes utilizing limited training sample data. Moreover, to extract relevant features for the route recommendation system, both global and mixed kernel functions were incorporated into the algorithm.

The algorithm had carried out adaptive learning using the particle swarm optimization (PSO) procedure with multi-time scale constraints. Additionally, it ensures convergence control by mining geographic information from datasets of cities. The study considered different tourists' personal interest preferences and their social characteristics in urban tourism by combining them with a gradient algorithm for particle swarm evolution and self-adaptive optimization. This was done to make the real-time recommendation system more intelligent in considering the tourists' needs (Song and Jiao, 2023).

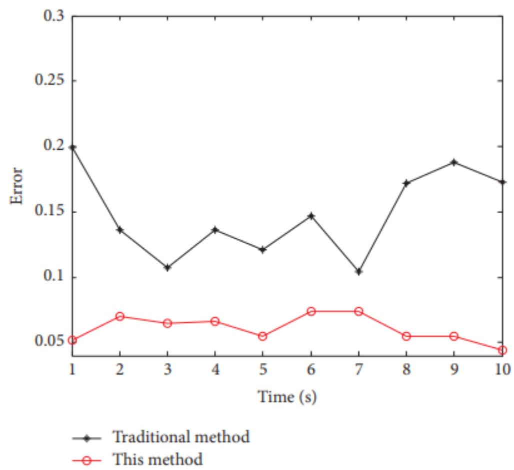

Figure 3: Comparison of Error Rates between two models

The outcomes of the model built demonstrated the algorithm's effectiveness, yielding accurate and precise recommendations for cross-regional city-level tourism routes. Furthermore, the algorithm exhibits robust convergence in swarm intelligence optimization (SIO), avoiding local optima during real-time tourism route recommendations. This contributes to enhanced intelligence and global stability in recommending cross-regional city-level tourism routes under epidemic normalization (Song and Jiao, 2023).

In conclusion, the literature review highlighted the challenges posed by traffic congestion and the shortcomings of current systems that lack real-time traffic data into consideration. The R3 system by Wang et al. (2014) stands out for its personalised and dynamic route recommendations based on taxi data, consistently outperforming traditional mapping systems. Furthermore, Hendawi et al.'s (2017) PreGo solution addresses these issues by combining real-time data with revolutionary algorithms. Song and Jiao (2023) propose an advanced city-level recommendation system for tourism itineraries that takes into consideration personal preferences and time restrictions. The collective implications of these research findings underline the necessity of integrating real-time data, advanced algorithms, and user-centric approaches for developing optimal route recommendation systems that possess the potential to revolutionise urban transportation.

Research Design and Methodology

Tests Proposed

Four types of unit testing were proposed to be conducted on different tasks in the system.

The first task is to examine the **user query's response time**. This test could help in evaluating the system's responsiveness to user queries by reporting the elapsed time of the queries submitted at the end of each run. The process of this test is to first simulate a set of user queries with different source and destination, then, a function for recording the time of each query will be used. In the end, the minimum, maximum, total and average time taken will be provided as an analysis report for the tester.

The second task is to assess how is the system's ability to **update its route information** when changes occur in the road network. For instance, the update of routes can be conducted using Dijkstra's Algorithm which is an algorithm focuses on finding the best route in a graph. The results of the update will be displayed in the console by showing the nodes that the new route will pass by. The tester will then be able to ensure that the route is improved.

The third task is to evaluate the system's ability to adapt to the **changes in motorist speed**. The system will display the updates in motorist data, especially in the changes of speed from point A to point B. The system's response to these updates will be monitored and their accuracy of recommended route after speed updates will also be justified.

The last task is to verify the **correctness** of the system's route recommendations, including distance, speed and time calculations. For example, the ability of the system to provide travelling time in terms of minutes is highly important in this program thus correctness test will be done to ensure no errors occur.

Resource Requirements

The resource required for this program can be distributed into two parts, software and hardware.

For the software segment, a few resources will be utilized for the system to be fully functional as a route recommendation system. First, it is important to equip FreeRTOS in the system to fully simulate the program in a real time scenario during task scheduling. Next, the usage of C compiler and its development environment is also important as it is used to code and execute the program with FreeRTOS.

Two algorithms are planned to be deployed into the system. One, the Dijkstra's Algorithm. It is used to conduct search towards the graph to find the best route in it (GeeksforGeeks, 2018b). The other one is the Haversine Formula, it is an accurate way to obtain the distance between two points on the surface of a sphere (GeeksforGeeks, 2018a). This formula will be used because the distance will be calculated using the coordinates of the various malls, thus, the formula could help in finding the distance just by deploying the function.

For the hardware part, it is simple, it just needs a computing system that can run this Real-Time Route Recommendation System. The laptops of both researchers could run the tasks assigned in the program.

Methodology Selected

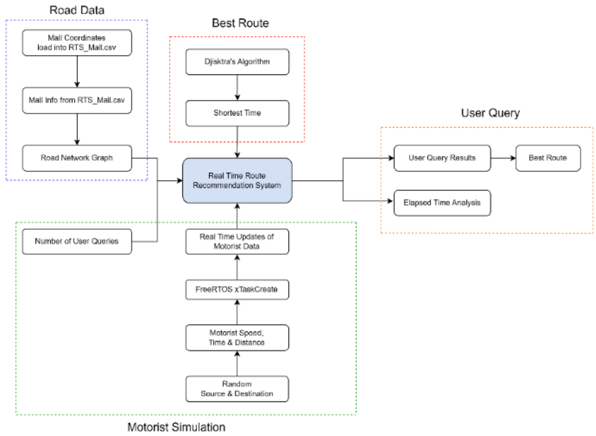

Figure 4: System Flowchart

The above figure shows the planned ways in which the Real-Time Route Recommendation System will be conducted shown in a flowchart.

First, road data initialization. The coordinates of the 50 shopping malls in Klang Valley will be obtained from Google Maps and then load them into a csv file known as "RTS_Mall.csv". The csv file is then read by the system to load into it. The road network will then successfully be available in the program and ready for the next steps.

The motorist simulation and best route sections are working closely together. The number of user queries to be prompted will be decided first. Then, a randomize source and destination malls are generated by the system before proceeding to provide a randomized motorist data which includes speed, distance and time information. The shortest time and best route will be calculated by the Dijkstra's Algorithm while being processed by the FreeRTOS API. Then the real time updates of all motorists' data are displayed in the console.

The user query box represents the output expected when the system finished its processes. First, the user query results, the pre-determined number of results and best routes should all be provided by the system when it ends. Besides that, the elapsed time should also be provided at the end for analysis purposes. The minimum, maximum, total and average elapsed time of the user queries can help in understanding the efficiency of the entire system.

Reasons for This Approach

1. **Efficient Road Data Initialization**

By using real world coordinates, the system ensures an accurate representation of the road network in the program. This is efficient as it represents a more real-life scenario in this route recommendation system.

2. **Integration of Motorist Simulation and Best Route Calculation**

The dynamically generated user queries and motorist data allow the system to simulate real-world scenarios whereby a traffic jam could occur at any time if there is an unfortunate incident. Therefore, the combination of best route calculations is also useful as this is needed in the real life as well.

3. **Real-Time Updates and FreeRTOS Integration**

The real-time updates of different motorist scenarios ensure that the system could adapt quickly into changing conditions. The integration with FreeRTOS also helps in facilitating multitasking while enhancing positively on the system's performance and therefore making it more towards real time.

4. **Comprehensive User Query Output and Elapsed Time Analysis**

Users can receive a comprehensive information about all the motorist data and user queries at the end of the simulation. The elapsed time analysis also allows easy performance evaluation to ensure the system meets the expected output.

Other Possible Approaches

There are a few more other methods to complete a route recommendation system when compared to the current one.

First, a static road data initialization could be performed to obtain a simpler data attribute. For example, the coordinates of a mall A can be defined as (1, 2) instead of long latitudes and longitudes to complete a route recommendation program. However, the simplicity in this setup may lack accuracy as no real-world data has been utilized.

Next, the user queries could be processed in a batch processing mode rather than a real time scenario. This will ease the development of the entire system as the understanding of FreeRTOS API will not be required. However, the real-time responsiveness will be reduced and the program may take a while to complete.

Lastly, there are also many other algorithms offering best route searching such as A* and Floyd-Warshall. These algorithms are approaches that could be tried in the future to explore and compare their difference between each other.

Results and Discussion

User Query

User query	Total Elapsed Time	Min elapsed time	Max elapsed time	Average elapsed time
1000	0.1050	0.0000	0.0010	0.0001
5000	0.5040	0.0000	0.0010	0.0001
10000	0.9960	0.0000	0.0010	0.0001
15000	1.5070	0.0000	0.0010	0.0001
20000	2.0300	0.0000	0.0010	0.0001
25000	2.5080	0.0000	0.0010	0.0001
30000	3.0030	0.0000	0.0090	0.0001
35000	3.4830	0.0000	0.0090	0.0001
40000	4.0230	0.0000	0.0090	0.0001
45000	4.5220	0.0000	0.0090	0.0001
50000	5.0020	0.0000	0.0090	0.0001

Table 1: Statistics on Number of User Query and Its Durations

Figure 5: Line Chart of Number of User Queries Against Total Elapsed Time

The first task conducted was to report on the user query's response time. Table 1 is based on a comprehensive analysis provided by the system consisting of total, minimum, maximum and average elapsed time per query. The performance and efficiency of the system's responsiveness was evaluated based on the durations. Figure 4 represents a line chart which shows the relationship between total elapsed time and number of user query. The line chart shows a directly proportional relationship between both variables which indicates that as time increases, the number of queries generated also increases. The average elapsed time is very low, with 0.0001 seconds per query which indicates that the system can handle a significant number of queries efficiently. These results are the desired outcome for the entire system as it ensures that a real time environment is created.

Routing Update

```
-------------------------------------------
                Query 100

Shortest Route from Mall 4:
Start Mall      | End Mall       | Path
4               | 1              | 4 2 1
4               | 2              | 4 2
4               | 3              | 4 2 1 3
4               | 4              | 4

Path            : 4 -> 2 -> 1
Start Mall      : 4 - Amcorp Mall
End Mall        : 1 - 1 Utama
Distance        : 5.79 kilometers
Speed           : 66.20 km/h
Time taken      : 5 min 14 sec
                0.0000 sec
-------------------------------------------
```

Figure 6: Shortest Time Route Calculated via Dijkstra's Algorithm

Based on Figure 5, it shows one of the queries, Mall 4 to Mall 1, updated in the program. It will first show the shortest time route from Mall 4 to every other node using the Dijkstra's Algorithm. For instance, to go to Mall 1, the shortest time route would be via Mall 2 as it is the fastest route from Mall 4. This is because the duration calculated for the direct route from Mall 4 to Mall 1 is slower than taking a little detour to Mall 2. The information of the path is then displayed along with the distance, speed and estimated time arrival. The output shows the system's capability to update the route information dynamically as the best route could be found rapidly while reflecting its responsiveness to the changes in road network.

Motorist Simulation

```
0 * Current Speed: 93.02
1 * Current Speed: 60.00
2 * Current Speed: 69.65
3 * Current Speed: 107.75
4 * Current Speed: 80.73
5 * Current Speed: 119.61
6 * Current Speed: 65.47
7 * Current Speed: 57.51
8 * Current Speed: 86.33
9 * Current Speed: 68.52
10 * Current Speed: 50.51
11 * Current Speed: 90.34
12 * Current Speed: 103.94
13 * Current Speed: 57.53
14 * Current Speed: 117.84
15 * Current Speed: 75.86
16 * Current Speed: 113.28
17 * Current Speed: 86.52
18 * Current Speed: 65.59
19 * Current Speed: 71.18
20 * Current Speed: 83.49
21 * Current Speed: 71.74
22 * Current Speed: 79.01
23 * Current Speed: 103.53
24 * Current Speed: 108.91
25 * Current Speed: 77.24
.
.
.
Average Speed   : 97.48
Path            : 42 -> 38 -> 11 -> 48 -> 18
Start Mall      : 42 - Sunway Pyramid
End Mall        : 18 - Ikea Damansara
Distance        : 27.73 kilometers
Speed           : 97.48 km/h
Time taken      : 12 min 13 sec
```

Figure 7: Periodic Motorist Data Update

Based on Figure 6, periodic updates of motorist data will be conducted to provide different routes based on the changing conditions. For instance, the current speed will always change to simulate real time traffic conditions. After the updates, the recommended path will be displayed along with other relevant factors. The output demonstrates the system's responsiveness to changes in motorist speed within the time range. The system can provide accurate route recommendations based on the updated information.

Correctness

The correctness of the entire system can be seen from the results discussed above. The system can provide relevant information such as generating user queries in a real-time manner, consistent motorist data updates and best route recommendations. All these conditions not just ran without errors but also all met the expectations of a real time route recommendation system.

Conclusion

The “Real-Time Route Recommendation System" designed and implemented in this research demonstrates promising results in addressing and simulating the issue of traffic congestion between 50 different malls in Klang Valley, Malaysia. The aim and objectives of this project to recommend best routes based on traffic conditions were achieved with a complete simulation on FreeRTOS and C. Future research could explore alternative algorithms such as A* and Floyd-Warshall in the searching of best route and compare all their performance in this system. Further enhancements could be done by incorporating machine learning techniques to predict and adapt the traffic patterns more accurately. The other possible approaches discussed in the research design segment could also be conducted in the future to compare the difference in functionalities between static and dynamic route recommendation system. Continued research and development in this field could contribute significantly to the advancement of intelligent transportation systems.

References

GeeksforGeeks. (2018a, August 10). *Haversine formula to find distance between two points on a sphere*. GeeksforGeeks. https://www.geeksforgeeks.org/haversine-formula-to-find-distance-between-two-points-on-a-sphere/

GeeksforGeeks. (2018b, November 19). *Dijsktra's algorithm*. GeeksforGeeks. https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/

Hendawi, A. M., Aqeel Rustum, Amr Ahmadain, Hazel, D., Ankur Teredesai, Oliver, D., Ali, M., & Stankovic, J. A. (2017). *Smart Personalized Routing for Smart Cities*. https://doi.org/10.1109/icde.2017.172

Song, Y., & Jiao, X. (2023). A Real-Time Tourism Route Recommendation System Based on Multitime Scale Constraints. *Mobile Information Systems*, *2023*, 1–10. https://doi.org/10.1155/2023/4586047

Wang, H., Li, G., Hu, H., Chen, S., Shen, B., Wu, H., Li, W.-S., & Tan, K.-L. (2014). R3: A Real-Time Route Recommendation System. *Proceedings of the VLDB Endowment*, *7*(13), 1549–1552. https://doi.org/10.14778/2733004.2733027

Appendices

Workload Matrix

Area	Ng Wen Xuan TP060653	Yam Chen Xi TP061635
Documentation		
Abstract	50%	50%
Introduction	50%	50%
Literature Review	50%	50%
Research Design & Methodology	50%	50%
Results & Discussion	50%	50%
Conclusion	50%	50%
Coding Implementation		
Road Data	80%	20%
Motorist Simulation	80%	20%
Best Route	20%	80%
User Query	20%	80%
FreeRTOS Implementation	50%	50%
Overall		
	50%	50%
Signature		
	Xuan	Yam Chen Xi