International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 04 | Apr-2018 www.irjet.net p-ISSN: 2395-0072

SMART CITY TRAVELLER

SOMANNA P D[1], SURAJ S RAO[2], VINAYKUMAR[3], SHUVAM PRAKASH[4], G S MADHAN KUMA[5]

[1,2,3,4,5] *Department of Computer Science and Engineering The National Institue of Engineering(NIE) Mananthavady Road, Mysuru – 570008, Karnataka,*

Abstract *–Currently, we realize that in general tourists spend a lot of time planning their trips because they need to make the most of every moment. In this context, this application aims to identify the main computing needs to support the improvement of tourist point of promotion for the traveller, by the means of a mobile application proposal. However, most of recent tourist and travellers think that they want to know the local charm peculiar to the land as well as a famous sightseeing spot. In order to achieve this, we propose a system that can automatically show a sightseeing route and plan in set time.*

Key Words: Travel, Planning, Ionic Framework, Google Maps API, Firebase.

1. INTRODUCTION

At present, in general tourists and travellers waste a lot of time planning and deciding their trips to achieve maximum satisfaction. In this context, this application aims to identify the main computing needs to support the improvement of tourist point of promotion for the traveller, by the means of an easy to use mobile application proposal.

Normally, most travellers like to visit the famous sightseeing spots as well as local charms unique to that place. To achieve this, we propose a system that can automatically show a travel route and plan for the user. This application also leads to quicker decision making with respect to places to visit.

This system is basically used to help a traveller new to the city or anyone who wants to explore a city within a specific time period. The user is supposed to enter his/her interests and preferences while signing up. Once the account has been created, the user can choose the location manually or let the system detect his/her current location as the starting and ending point of the trip. Then, the start and end time of the trip must be specified by the user. Since all the trips of a user will be stored, he/she can also view the previous trips. Smart City Traveller as the name indicates, smartly makes its way in analyzing users' interests and preferences and the time period the user is willing to explore a place and designs an itinerary and a route with the best tourist spots around the selected location such that he/she returns to the starting location by the specified end time. This makes use of shortest path algorithms for determining the route.

The system makes use of the Google Maps API to get all the places around the selected location with all their information. Then, these locations are sorted based on ratings, distance, and various other constraints to place it before the user.

2. SOFTWARE DESCRIPTION

2.1 Ionic Framework

Ionic is a complete open-source SDK for hybrid mobile application development. The original version of it was released in 2013 and built on top of AngularJS and Apache Cordova. Ionic provides tools and services for developing hybrid mobile apps using Web technologies such as CSS, HTML5, and Sass. Apps can be built with these Web technologies and then distributed through native app stores to be installed on devices by leveraging Cordova. The functionality which is found in native mobile development SDKs can be provided by Ionic. Users can build their apps, customize them for Android or iOS, and then deploy through Cordova. Ionic contains mobile components, typography, interactive paradigms, and an extensible amount of base theme. Using Angular, Ionic provides custom components and methods for interacting with them.

There are many other features such as push notifications, A/B testing, analytics, code deploys, and automated builds which helps the developers. For Android, Ionic supports Android 4.1 and higher versions. For iOS, Ionic supports iOS 7 and higher versions. Ionic 2 supports the Universal Windows Platform for building Windows 10 apps. Ionic Framework, powered by Angular.js, also supports BlackBerry 10 apps.

2.2 Google Maps API

Google Maps API allows maps to be added based on Google Maps data to an application. The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures. API calls can be used to add markers, polygons, and overlays to a basic map, and to change the user's view of a particular map area. These objects provide additional information for map locations, and improves user interaction with the map. The Google Places API for Android allows to build location-aware apps that respond contextually to the local businesses and other places near the device.

2.3 Firebase

The Firebase is a mobile and web application development platform developed by Firebase Inc. in 2011, it was then later acquired by Google in 2014. Firebase provides services such as a real-time database and backend. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud. Firebase uses a NoSQL format, which

means that it is easy to use and it doesn't contain tables or support queries hence providing an advantage over traditional relational database. Also, it is real time, so as soon as a change is made to the data (a new comment is added for example), all connected clients will immediately be updated.

3. LITERATURE SURVEY

The development of internet stimulates the emergence of various Online Tourism Agencies (OTA) to post their service information online [1]. One challenging problem of OTA is how to recommend appropriate travel routes for users with different requirements. The path planning of scenic tourism is a relatively new research field. Some recent research on the travel route planning has been carried out.

Ying Xu, Tao Hu and Ying Li [2] proposed that a new Improved PRR algorithm (IPRR) based on the PRR by considering different personalized requirements in order to recommend high-quality travelling routes for customers. The IPRR algorithm takes various factors into account, including the user's personal preferences, user types, the real-time traffic condition of the tourism region (i.e. the real-time nodes and the number of people on the path), and the historical statistical data (i.e. historical tourists number at the spot).

Graph search algorithms have often been adapted for both indoor and outdoor path planning [3]–[7]. In these applications, an operational terrain of a mobile agent (e.g. a vehicle, a mobile robot, etc.) is represented using a graph that consists of a set of nodes and a set of edges. A node represents a special location on the terrain surface and an edge represents the connection between two nodes. An edge is associated with one or more costs. For examples, the edge cost is a distance measurement when finding the shortest path and a time measurement when finding the fastest path. Early works on graph search algorithms are based on uniform cost search mechanisms. Dijkstra's algorithm [4] is a prime example for such uniform cost search algorithms. It starts a search process from a source node and iteratively selects a node for expansion until it selects a target node for expansion. Here, node selection is based on the cost between the source node and a given node without considering its remaining cost to the target node. Thus, in finding the shortest path, it expands all the nodes that are closer to the source node compared to the target node. Obviously, this results in excessive expansion of nodes that do not lie on the optimal path, thus, degrading the efficiency.

The same techniques discussed for shortest path planning might not always be used to find the fastest path in outdoor environments due to the inability of mobile agents to travel at their peak speed everywhere in irregular terrains. Mobility maps are an effective way for dealing with such irregularities. In the paper proposed by Wanmai Yuan, Nuwan Ganganath, Chi-Tsun Cheng, Guo Qing, and Francis C.M. Lau [8], a grid-based mobility maps was introduced for representing speed limitations in outdoor terrains. Further, a heuristic approach for finding the fastest path on such maps was introduced. The proposed heuristic is proven to be both admissible and consistent. Therefore, it can be used with A*-like heuristic search algorithms for obtaining fastest paths efficiently.

4. SYSTEM ANALYSIS

4.1 Existing System

In the existing system, it is necessary for user to input the name of the destination exactly. If sightseeing place is decided users do not have any problem (Google Maps). But, if the user wants to explore new places which he is not aware of then this system is not desirable. Current system shows only the top locations around the user. But, the user has to choose the places he wants to visit and search routes for each place separately. Furthermore, Google map displays it only to the route of the destination. On the other hand, in this system, the point that can propose a sightseeing route and sightseeing plan in the planned time to return is big superiority. In the existing tourist guide system, user is necessary to input an individual visit. Therefore, it is necessary for the traveller to prepare for sightseeing spot beforehand. Traveller can only visit the places which he is aware of. If it is a famous sightseeing spot, traveller can easily check it on a book or Internet. However, if it is not a famous sightseeing spot, but there are a lot of attractive places the traveller will not be aware of it.

4.2 Proposed System

In the proposed system, "planned time to return" is also considered at the time of the use. In this way, this system can propose a sightseeing route and sightseeing plan that it can guide that used present time from time to return time automatically.

The system automatically searches for places of interest around the location. So, the traveller will not miss out on any attractions which he is unaware of.

The interests and preferences of the user is also considered and the places are chosen accordingly.

In this application system, we make a list of sightseeing information according to sightseeing categories.

In the result, we display both guidance routes on the map and guidance routes by text.

The application displays a menu with the main categories of attractions available in the city, by clicking on one of the categories all the related information is displayed.

5. SYSTEM DESIGN

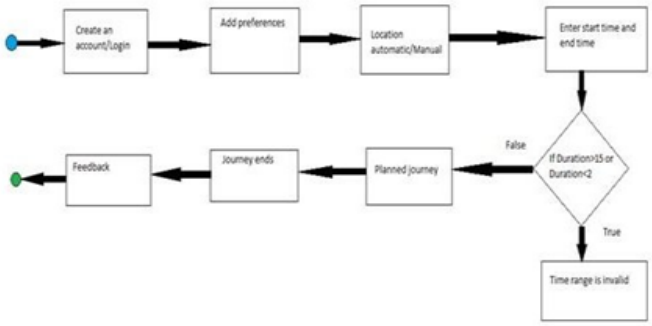

Fig. 1 Activity Diagram

In the above fig. 1, the flow of activities that occurs in the Smart City Traveller system is portrayed with the conditional branches that occur over various working scenarios.

6. IMPLEMENTATION

6.1 Improved Personalized Route Recommendation Algorithm

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes Supposing that there are n scenic spots within the tourism region, the network topology can be expressed as an undirected graph GV

E(,) , V is the set of scenic spots. E is the set of edges, (i j,)∈E represents the interconnected bidirectional path from node i to node j.CNR is the current number of ride for node i. Ri represents the hot class of node i . Supposing that the distance between two node i and j is dij and the travelling time is the sum of Path Statistical Time tij and node average Stay TimeTi. S is the matrix of the statistical path selection, each element sij in the matrix is the number of path from the node i to node j.

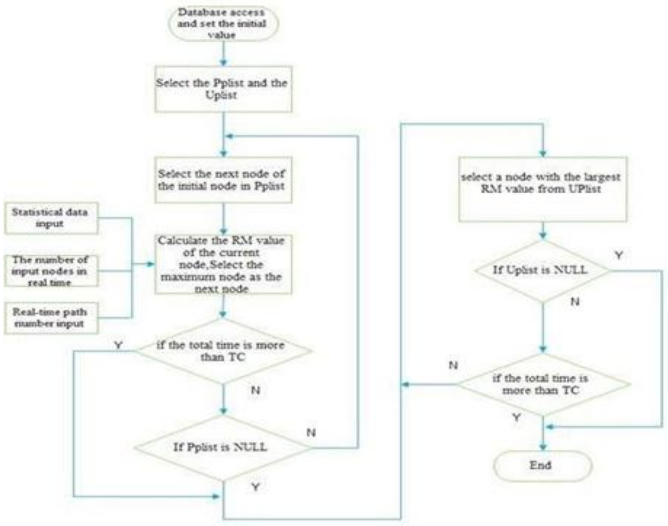

Fig 2 Flowchart

The purpose of the travel route planning problem in this paper is to generate a travel route which minimizes the objective function measured by the congestion, the travel cost, the stay time and so on, while satisfying the following constraints:

1) Customers' preference for the node based on the statistic historical data and the popularity

2) Congestion situation at each node

3) Time constraint (TC) for travelling

Based on the PRR (Personalized Route Recommendation) algorithm [8], this paper proposes an improved PRR (IPRR) algorithm to solve the travel route planning problem with a variety of constraints. Our proposed IPRR algorithm considers the historical statistic of tourists choices (matrix S and matrix T) and the personal preference [8] to calculate the RM (Recommendation Matrix). Then, the recommended travel route is calculated based on RM.

6.2 Ant Colony Algorithm

Ant colony optimization (ACO) is a technique of problem solving inspired by the behavior of ants in finding paths from the nest to food and a new search metaphor for solving combinatorial optimization problems, and has been unexpectedly successful in recent years [11-15] . The ants cooperate with each other indirectly and the communication is setup by pheromone trails of scent and find an optimal solution to their tasks using both the information (exploitation) which has been acquired in the past and search (exploration) of the new route. Therefore the premature convergence probability , are used to counteract premature stagnation of search, in order to avoid early convergence to a local minimum, maintaining some kind of elite strategy at the same time.

ACO is a new search metaphor for solving the combinatorial optimization problems, and has been successfully applied to many complex optimization problems, especially applied to the complex discrete optimization problems. The essence of the ant colony optimization process lies in: (1) the selecting mechanism: the more pheromone trail on the route, the larger probability of choosing this route; (2) the updating mechanism: the pheromone on the route can get strong when ant's pass more and moreover simultaneously also gradually volatilizes and vanishes as time goes on; (3) the coordinated mechanism: the mutual communication and coordinated operation between the ants is performed in fact through the "pheromone". The ant colony algorithm has fully used such optimized mechanism to finally find the best solution through information exchange and mutual cooperation between the individuals, and enable this algorithm to have a very strong ability to discover the best solutions.

6.3 A* Heuristic Algorithm

In computer science, A* is a computer algorithm that is widely used in path finding and graph traversal, the process of plotting an efficiently directed path between multiple points, called nodes. It is widely used because of its performance and accuracy.

A* is an informed search algorithm, or a best-first search algorithm, which means that it solves problems by searching among all possible paths to the solution for the one that gives the smallest cost (least distance travelled, shortest time, etc.), and among these paths it first considers the ones that lead most quickly to the solution. It is formulated in terms of weighted graphs: starting from a specific node of a graph, it constructs a tree of paths starting from that node, expanding paths one step at a time, until one of its paths ends at the predetermined goal node.

At each iteration of its main loop, A* needs to determine which of its partial paths need to expand into one or more longer paths. It does so based on an estimate of the cost (total weight) still to go to the goal node. Specifically, A* selects the path that minimizes.

$$f(n) = g(n) + h(n)$$

where n is the last node on the path, g(n) is the cost of the path from the start node to n, and h(n) is a heuristic that estimates the cost of the cheapest path from n to the goal. The heuristic is problem-specific. For the algorithm to find the actual shortest path, the heuristic function must be admissible, meaning that it never overestimates the actual cost to get to the nearest goal node.

7. CONCLUSION

Since travelling is one of the important aspect today, it is very necessary that proper planning need to be done beforehand in terms of time management. Most people without using the latest technology waste a lot of time just planning trips. So, an application like Smart City Traveller really helps tourists to utilize their precious time to the fullest and also enjoy their trip at the same time.

REFERENCES

1. LV H.L., WANG J.L. & DENG F, A Recommendation Algorithm For Individualized Travelling Route. Network New Media

2. Ying Xu, Tao Hu, Ying Li "A Travel Route Recommendation Algorithm with Personal Preference" 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) 2016.

3. N. Ganganath, C.-T. Cheng, and C. K. Tse, "Rapidly replanning A*," in International Conference on CyberEnabled Distributed Computing and Knowledge Discovery (CyberC). IEEE, 2016, pp. 386–389.

4. E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische mathematik, vol. 1, no. 1, pp. 269–271, 1959.

5. M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, "Anytime search in dynamic graphs," Artificial Intelligence, vol. 172, no. 14, pp. 1613 – 1643, 2008.

6. N. G. Marasinghe Arachchige, "Heuristic search algorithms with applications to path planning on uneven terrains," Ph.D. dissertation, The Hong Kong Polytechnic University, 2016

7. N. Ganganath, C.-T. Cheng, and C. K. Tse, "A constraint-aware heuristic path planner for finding energy-efficient paths on uneven terrains," IEEE Transactions on Industrial Informatics, vol. 11, no. 3, pp. 601–611,2015.

8. Wanmai Yuan, Nuwan Ganganath, Chi-Tsun Cheng, Guo Qing, and Francis C.M. Lau "A Consistent Heuristic for Efficient Path Planning on Mobility Maps" Department of Electronic and Information Engineering, the Hong Kong Polytechnic University (Projects RUWM and GYBKH) and NSF of China (Grant No. 61372095).

9. KazuyaMurata and Takayuki Fujimoto "Proposal of Multiple Travel Scheduling System based on Inverse Operation Method" 978-14799- 8679-8/15/$31.00 copyright 2015 IEEE ICIS 2015 June 28-July 1 2015, Las Vegas, USA.

10. Ivaldir de Farias Junior, Nelson Leitão Júnior and Marcelo M. Teixeira, "Urbis: A Touristic Virtual Guide"

11. https://developer.Google Maps.com/docs/resources/categories

12. https://www.researchgate.net/figure/271458021_fig2_Figure-2-Classification-of-DynamicRoute-Planning-Algorithms

13. Chin-Jung Huang, Ying-Hong Lin,The Approximate Shortest Distance Route Intelligent System for Travelling for Taiwan Innovative Computing Information and Control, 2006

14. K. Ogawa, Y. Sugimoto, K. Naito, T. Hishida, T. Mizuno, "Basic design of a sightseeing recommendation system using Characteristic Words", IPSJ SIG technical reports 2014-MBL-71(14), 16, 2014-05-08.

15. W. Souffriau, P. Vansteenwegen, J. Vertommen, G. V. Berghe, and D. V. Oudheusden. a Personalized Tourist Trip Design Algorithm for Mobile Tourist Guides. Applied Artificial Intelligence, 22(10):964- 985, Oct. 2008.