GIS Enabled Travel Planner System with TSP Implementation


Abstract — A travel planning system which offers facility to find optimum route to (1) reach a known tourist attraction and (2) find optimum route to move around that area visiting required locations is proposed in this study. The conventional Dijkstra’s algorithm and Traveling salespersons problem (TSP) algorithms are applied to a geo-enabled transportation network to find the optimum paths. Main functionalities are implemented as a web GIS application based on free and open source technologies. The travel plan once designed can be accessed via mobile Internet and as an Android application also. A linear map is also provided to support the mobile user. The implementation is optimized to give a faster response to the route queries. Comparison of results with available travel planning systems is also presented. TSP implementations and linear maps are not popular in existing travel planning applications. Prototype implementation is done based on spatial data of Anuradhapura area of Sri Lanka.

Keywords — Geographical Information System, optimal path, road network, algorithms, spatial data, travel planner system

I. INTRODUCTION

Planning a travel route before starting the travel is a common requirement of a tourist. This becomes a necessity where more and more people tend to travel to unfamiliar places in local and foreign countries. Maps become a vital component of travel planning.

This study provides a novel solution to assist travelers to plan their journey beforehand, where users are able to use a geo-enabled web interface to,

1. View a full detailed map.
2. Obtain optimal routes.
3. Search for a known place or path.
4. Obtain location information on supportive services such as ATM, Banks, Hospitals, Accommodation and Restaurants.
5. View linear maps.

Travelling to a tourist attraction can be understood based on two tour segments; (1) travel to the destination and (2) travel within the destination. Travelling within the destination may involve visiting several places of interests, restaurants etc which needs to be visited in one tour. The destination can be reduced to a city for easy interpretation. For example if a person in Colombo(C) wants to visit Anuradhapura (A), his journey can be divided into two parts; tour from C to A and tour within A. Both these segments are considered in this study.

Additionally a facility to generate a linear map is provided, which is more readable than a conventional map while on the move. The web interface can be used before the travel for planning and then the plan can be accessed through mobile phone while travelling. To cater for wider audience, two versions, mobile web based version which aimed at mid range phones and Android application targeting smart phone users are designed. Mobile phone is considered as the most suitable mode to reach the general public due to its wider acceptance in general public.

Available on-line travel planning systems lack the facility to address the issue of planning the travel within destinations. They do not provide facility to generate linear maps as well.

A prototype application is developed based on Anuradhapura, a popular travel destination of Sri Lanka. Local data sets are used which avoid the binding of the system to external data providers. The system is implemented with free and open source software tools including map servers, web map interfaces and spatial databases.

II. BACKGROUND OF THE STUDY

A Geographic Information System (GIS) integrates hardware, software, and data for capturing, managing, analyzing, and displaying all forms of geographically referenced information. GIS allows us to view, understand, question, interpret, and visualize data in many ways that reveal relationships, patterns, and trends in the form of maps, globes, reports, and charts [1].

Several Algorithms exist for finding shortest paths between two known locations, based on different cost factors. Shortest path algorithms such as Dijkstra’s algorithm [2], and BellmanFord algorithm [3].
Traveling salespersons problem (TSP) [4] is one of the main research problems of the day that carry significant value to the transportation networks. The exact solution of a Traveling salespersons problem is not feasible. There are some good approximation algorithms that can provide an approximate solution. Having the TSP solved for transportation networks, enables to optimize many transportation tasks. Some potential applications of TSP in transportation context are postal delivery, tourist visits, garbage collection and pizza delivery.

In this study, Dijkstra’s algorithm and an approximation algorithm to solve TSP are used to get the optimum path between locations with real world spatial data.

III. METHODOLOGY

A web-based GIS is selected as the main mode for implementation as then the system would be accessed online by any web user. However the system is allowed to be accessed via three different mediums, web interface, mobile access through mobile web and mobile access through android smart phones.

Fig 1 illustrates the different type of users that are involved with the system. The main components of the system are as per Fig 2. It has four main layers.

a. Data Layer – stores spatial and non-spatial data relates to road networks and service points.
b. Server Logic Layer – Web server and map server
c. Application Logic Layer – optimal path generating algorithms, searching and other core functionalities are handled by this layer.
d. User interaction handling Layer – deals with three types of the user that are involved with the system. Each user interaction mode is powered by the application logic layer.

A stack of free and open source and low cost software tools is used for the implementation of the system. The database is implemented using PostgreSQL 8.4.12 with PostGIS 1.5 spatial extension. The core system is designed using Dreamviewer CS4 and written using javascripts. UMN MapServer [5] is used as the map server to render spatial data into map data. OpenLayers javascript API is used as the map viewer in web applications. Smart phone application is implemented using Android 4.0. Mobile web implementation uses HTML and CSS. Topology building, network creation and data cleaning and correction are done using Quantum GIS 1.6.

The data set used is of 1:250000 scale and is in WGS 84 coordinate reference system. Following section explains the implementation of different components of the system.

A. Data Layer

Data layer contains the implementation of database which has the capability of manipulating spatial data. Spatial data is a special kind of data that define a location. These are in the form of graphic primitives that are usually points, lines, polygons or pixels [6]. Not like normal integer or decimal data, geographical information is usually stored as coordinates or topologies [7].
Fig 3 illustrates the adopted procedure in database implementation process. It is a step by step approach to handling spatial data which were used in important situations like algorithm implementation.

**B. Server Logic Layer**

UMN MapServer is used to generate the required map by rendering map layers which are written in the map file. Map file fetches data from the database and format them using the WMS standards. The MapServer application sends the rendered output through Apache web server to the map displayer component.

Steps involved in generating the final map output:
- Create the mapfile for MapServer application by defining the area of the map and where to output images. Define basic map layers, including their data sources, projections, and symbology.
- Using runtime substitution [6], parameters are sent to the map file and the map file is modified using WMS standards.
- By using OpenLayers with same projection in all layers, implement basic GIS capabilities like zooming and panning.

**C. Application Logic Layer**

This particular layer contains the implementation of optimal path generating algorithms. When combining spatial data with selected algorithms to generate optimal paths, there are several factors that have to be considered.

a. Creating a road network out of available road data – normal road data which can be considered as a set of line data cannot be manipulated using an algorithm. For that data has to be converted into a set of nodes and edges forming a graph.

That can be achieved by converting road data into a road network. Fig 4 illustrates the difference between normal road data and a road network.

b. Topological correction of data - In a road network type data set, having a topology is must. Because topology will model how points, lines and polygons share geometry. For example, adjacent features, such as two cities will share a common edge. Without having topology, it is difficult to maintain the relationship between two adjacency roads.

I) **Optimal path to reach the destination.**

This study uses Dijkstra’s algorithm to provide the optimal path to reach a destination. Dijkstra’s algorithm is a graph algorithm that simultaneously finds the shortest path from a single vertex in a weighted graph (requires nonnegative edge weights) to all other vertices in the graph [8].According to this study weight of the graph varies according to users’ choice of cost factor. If users’ concern is about distance, weight becomes length of the edges, if users’ concern is about traffic or road condition, weight changes accordingly. In order to provide geospatial routing functionality PostgreSQL/PostGIS database is extended using pgRouting [9].

II) **Optimal path to travel within a main city covering all user specified locations.**

Once user specifies his interested places the implemented system generates the order of places to be visited.

In this implementation the system gets a set of places from user and stores it in an array. Then by using
run time substitution, a sql query is executed and the resulted new layer is displayed on the map.

Travelling Salesman Problem (TSP) algorithm is used to generate the optimal path to travel within a main city covering all user specified locations. Goal of TSP algorithm is to find the shortest tour that visits each node in a given list exactly once and then returns to the starting node. A tour can be represented by a cyclic permutation $p$ of $\{1, 2, \ldots, n\}$ where $p(i)$ represents the city that follows city $i$ on the tour. To find a permutation $\pi$ that minimizes the length of the tour is denoted by:

$$\sum_{i=1}^{n} c_{\pi(i)}$$

In order to implement TSP algorithm, topologically corrected data is stored in PostgreSQL database and with the pgRouting extension TSP algorithm is executed. In this implementation length of the edges is not required. However corresponding longitude and latitude values of each point are required [10,11].

D) User interaction handling layer

I) Web Interface

The main system provides the web interface which allows the users to view the map, search for known locations and paths, get optimal paths to travel and save the map of the planned journey. The resulting spatial information is displayed on the map as highlighted routes and points.

II) Mobile web based application

Any mobile phone having web browsing facility can access the application using mobile web client. This component is designed with limited features for the reason that it should address the issues like low display resolution and limited bandwidth of mobile phones. It inherits all the basic functionalities from the main web application and uses the same back end processes. Furthermore, users can view the planned journey which is saved and stored using the main web interface, while they are travelling.

III) Android smart phone based application

Purpose of implementing this component is to make it possible for the users to view the planned journey offline. Users can download the planned journey as an image and that image is combined with the downloaded android application. Along with the android application a liner map is provided which is much more efficient when travelling in a vehicle. Linear map illustrates main cities and distance between them of the routes from Colombo to Anuradhapura.

IV. RESEARCH FINDINGS/RESULTS

The results of the implemented system were verified and validated using several methodologies. Initially the system was compared with the results given by Google maps for the same queries. Then the results for the same queries were produced based on personal views and ideas using a printed map issued by Survey Department of Sri Lanka. Next the results were manually calculated using mathematical formulas and compared them with system results.
Fig 6: Optimal path based on distance provided by the implemented system

Table 1: Evaluation of Algorithms

<table>
<thead>
<tr>
<th></th>
<th>Results Given By the System</th>
<th>Google Map Results</th>
<th>Personal Opinion/ Manual Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dijkstra’s Algorithm</td>
<td>When user provides the source and the destination, the optimal path to travel based on either distance, traffic condition or road condition is displayed on the map. Fig 6 shows the optimal path generated by the system based on distance, from Kandy to Anuradhapura.</td>
<td>Google map shows three alternatives to travel from Kandy to Anuradhapura. Fig 7 shows the shortest route suggested by Google maps to travel from Kandy to Anuradhapura. The optimal path generated by the system based on distance and the shortest path suggested by the Google maps is almost the same.</td>
<td>A printed map was analyzed in order to come up with the shortest path to travel from Kandy to Anuradhapura. Since there are several alternatives, total distance of each road is calculated and the route with the smallest total was selected as the shortest path. The result matched with the result produced by the system.</td>
</tr>
<tr>
<td>TSP Algorithm</td>
<td>The system allows the user to select the places and the system generates the optimal path to travel covering all user specified places. Fig 8 shows the generated optimal path to travel covering Sri Maha Bodhiya, Jetavanaramaya, Thuparamaya, Ruwanweliseya, Mirisawatiya, Isurumuniya.</td>
<td>Google or any other similar kind of GIS applications that were studied does not use the implementation of TSP algorithm.</td>
<td>The selected points are plotted in a graph according to their X Y coordinates. Then the optimal path is produced manually by looking at the scattering of the points on the grid. Fig 9 shows the manual calculation of TSP algorithm.</td>
</tr>
</tbody>
</table>

Table 1 describes how the implemented algorithms were evaluated for their level of accuracy.

Fig 5 demonstrates the map viewing page which contains the generated map. In order to enhance the visual clarity of the map different colors, labels and symbols are used.
V. CONCLUSION

The implemented application proved that it is capable of providing sufficient detailed information for a trouble free tour planning. In a real world implementation, the system can be improved in several ways. The system can be developed in a way so that it can track the current location of the user using GPS technology. Tracking the current location of the user is very important to generate more accurate optimal paths and to display supportive services. When the user enters the origin and the destination of the journey, the system generates the optimal path to reach the destination. The system can be further developed to give the necessary directions as well.
REFERENCES


