EFFICIENT ROUTE FINDER SYSTEM

Sanjay Kumar Yadav¹, Madhavi Singh²

¹Assistant. Professor, Department of Computer Science and IT, SHIATS, Allahabad, U.P, India  
²M.Tech. Scholar, Department of Science and IT, SHIATS, Allahabad, U.P, India

ABSTRACT

The work in this paper is to design and implement a Route Finder System that not only try to minimize the number of re-computations needed for computing shortest route or alternate route from source to destination but also models and process the data automatically upon loading of road network graph into memory of the system. We compared the performances of our Route Finder System with Mumbai navigator [8, 9], and we find that our proposed system has much less space and time complexities as compare to Mumbai navigator. We performed the simulation and experiments by using GeoServer, Java, PostGIS and PostgreSQL database.

Key Words: Transportation, Transport Management System (TMS), Intelligent Transport System (ITS), Traveller Information System (TIS), and Route Guidance System.

1. INTRODUCTION

One of the major challenges of transportation is continuous growth in traffic, in terms of number of vehicles. This enormous and rapid rise in vehicle population with increased urban land use has generated considerable travel demand as well as numerous transport problems. The need of Traffic management has changed the manual traffic lights into automated traffic lights on city streets, but the development and implementation of sophisticated integrated applications based on Intelligent Transport Systems (ITS) is much needed and has grown apace in recent years, as a result of successful research and technological advances. Various ITS products and services are already at work throughout the world, significantly improving transportation safety, mobility, and productivity [1].Still the deep and wide innovations of applications in intelligent systems represent a true revolution in the field of ITS. The main impetus behind any Route Finder System in this area is to analyze and provide a way for development of facilities that enhances the working of existing transport management system which will ultimately lead to enhancement of nations’ economy with decreased inflation. The performance of transport systems is of crucial importance for individual mobility, commerce and for the welfare and economic growth of all nations.
Figure 1 shows the rapid rise in vehicle population along with increased urban land use. This increased rate has generated considerable travel demand as well as numerous transport problems [2]. The congestion, safety and environmental problems increased to such an extent that it has become increasingly difficult to navigate due to the combined effects of rapid motorization and urbanization. Since IT enables elements within the transportation system—vehicles, roads, traffic lights, message signs, etc.—to become intelligent by embedding them with microchips and sensors and empowering them to communicate with each other through wireless technologies. Hence it can be emphasized that merging IT in the transport industry might lead to a new era where the best use of existing road infrastructure can occur.

2. MERGING OF INFORMATION TECHNOLOGY IN TRANSPORT MANAGEMENT SYSTEM

Imagine knowing real-time traffic conditions for virtually every road in the country and having that information available both in-vehicle and out [6]. Imagine driving down an expressway with a navigator device that combines GPS with real-time traffic information and alerts you that you are approaching a blind curve or traffic congested road and that you need to brake immediately or probably that can display real-time traffic information and optimize your route accordingly. Information technology (IT) has already brought great revolutions in many industries, and now appears poised to transform countries’ transportation systems. Indeed, IT has emerged like an info structure medium that brings new wireless technologies and devices which allows consumers to connect to real time information on regular basis.

Figure 2 shows the estimated increase in transport wireless devices from year 2006 to 2011. Thus, it can be concluded that consumer’s travel demand is heavily diverted towards using the Intelligent Transport Systems (ITS) (Fig 3) technologies and its application.
Intelligent transport systems include a wide and growing suite of technologies and applications such as real-time traffic information systems [7], in-car navigation (telemetric) systems, vehicle-to-infrastructure integration (VII), vehicle-to-vehicle integration (V2V), adaptive traffic signal control, ramp metering, electronic toll collection, congestion pricing, fee-based express (HOT) lanes, vehicle usage-based mileage fees, and vehicle collision avoidance technologies. With the development of Intelligent Transport System (ITS) and Geographic Information System (GIS), the increasingly intensive demand of route guidance system in real time has coincided with the increasing growth of roads in real world [4]. In the leading nations in the world, ‘ITS’ bring significant improvement in transportation system performance, including reduced congestion and increased safety and traveler convenience.

Figure 3: Intelligent Transport System

To achieve the solutions ITS [5] enables the help of Route Guidance Systems. A route guidance system helps to tackle many of the transportation problems by minimizing congestion and ensuring uniform utilization of the road network. For this purpose modeling of real road network into digital map format is necessary that requires large amount of pre-processing time and human effort. Thus the objective of ITS can be summarized as follows [6]:

- To improve traffic safety
- To relieve traffic congestion
- To improve transportation efficiency
- To reduce air pollution
- To increase the energy efficiency
- To promote the development of related industries

ITS try to improve the performance of a country’s transportation system by maximizing the capacity of existing infrastructure, reducing to some degree the need to build additional highway capacity.
3. RELATED WORK

Mumbai Navigator

Mumbai navigator is a route finder system that produces a travel plan [8, 9], with an estimate of the total journey time including the waiting time when origin and destination are given. The Mumbai navigator software requires a computation each time when the origin and destination of the travel is entered into the system. For example, the software requires 2000 execution computations for 2000 pair of origin and destination. Mumbai navigator generates travel plan on the basis of bus stops and bus routes while not on the basis of actual routes so, there may be a possibility that there exist a shortest path between origin and destination, but that path might not be included in the travel plan generated by Mumbai navigator since that may not be a bus route. Thus Mumbai navigator works only on the basis of shortest bus route and not on the basis of shortest route.

4. PROPOSED ROUTE FINDER SYSTEM

![Figure 4: Architecture of Route Finder System](image)

The proposed system is an application in the field of ITS with the help of GIS to provide efficient and safe road transport. Figure 4 shows overall architecture of proposed system, where real road network map (spatial data) is converted into Geography Markup Language (GML) file that is transformed into SQL file for loading into PostGIS and PostgreSQL database.

The details of components of the proposed architecture are as follows:

A. **ShapeFile**: A shapefile is a digital vector storage format for storing geometric location and associated attribute information. Shapefiles spatially describe features: points, lines, and polygons, representing, for example, water wells, rivers, and lakes. Shapefile stores nontopological geometry and attribute information for the spatial features in a data set. A Shapefile consists of a main file (.shp), an index file (.shx), and a dBASE table (.dbf).

B. **PostGIS & PostGreSQL**: PostGIS is a spatial database extender for PostgreSQL object-relational database. It adds support for geographic objects allowing location PostGIS "spatially enables" the PostgreSQL server, allowing it to be used as a backend spatial database for geographic information systems (GIS), much like ESRI's SDE or Oracle's Spatial extension. PostGIS follows the OpenGIS "Simple Features Specification for SQL" and has been certified as compliant with the "Types and Functions" profile queries to be run in SQL.
C. GeoServer & Open Layers: GeoServer - an open-source server written in Java - allows users to share, process and edit geospatial data. OpenLayers is a JavaScript library for displaying maps in the browser. GeoServer uses a web application framework known as Wicket for its user interface. Wicket differs from most Java web frameworks in that it is component based rather than JSP template based.

D. Netbeans: A Java integrated development environment for application development. The NetBeans Platform is an extensible framework which you can use to build professional OS-independent desktop applications.

4.1 Working of Route Finder System

The proposed system uses Floyd Warshall for computing all to all pair (A2A) shortest route (See Table 1) [3].

```
Floyd Warshall (W) Pseudo Code // W is weight matrix, initially adjacency matrix
1 { 
2 n = rows (W) 
3 D(0) = W 
4 for k = 1 to n 
5 { 
6 for i = 1 to n 
7 { 
8 for j = 1 to n 
9 { 
10 d_{ij}(0) = \min (d_{ij}(0-1), d_{ik}(0-1) + d_{kj}(0-1)) 
11 } 
12 } 
13 } 
14 return D(n)
15 }
```

Table 1: FW’s Pseudo Code for A2A Computation

Performing route finding techniques on a large graph slows down the processing of any algorithm. To increase the efficiency of these algorithm researchers do manual modeling and preprocessing of graphs [5]. This requires too much of preprocessing time, human effort and manual working. To minimize this computation time, the proposed system not only models and process the road network automatically but also provide different route finder options as follows:

- Shortest Route Finder (Based on constraints such as time, distance or traffic etc.)
- Alternate Route Finder (In case, there is change in topology of the shortest path)
- Facility Based Route Finder (In case, there is a route based on facilities such as hospital, petrol pumps, police station etc.)
- Hierarchical Based Routing (In case, there is a route based on hierarchies such as highways, expressways, main roads or mini roads) (See Fig 5)
- Multimodal Route Finder (In case, there is a route based on multiple modes of travel such as road and railways etc.)
4.2 Finding Shortest Route in Proposed System

Finding shortest route in old approach requires executing algorithm every time a shortest route finder query is encountered. The following steps are performed when a shortest route finder query is encountered:

(a) If shortest path table in database is empty, compute route by executing algorithm and store source, destination and path into database.

(b) If shortest path table in database is not empty then

- Extract all routes from table,
- Search whether source node “s” and destination node “t” both is present in any route. If so, extract substring between “s” and “t” which is shortest route.
- If no route contains source and destination node both, compute route by executing the shortest route finder algorithm.

Table 2: Pseudo Code for shortest route Computation

Suppose, for verification purpose we have a route 1→2→3→4→5→6→7→8→9→10 stored in database table as a shortest route between source node “1” and destination node “10” or source node “10” and destination node “1” in reverse direction (as all edges are bi-directional), then for all subset of cardinality 2 of set {2,3,4,5,6,7,8,9} where number denotes source or destination nodes, we don’t need to either compute route from algorithm nor we have to compute routes to be stored in the database table. Therefore, by storing a route with “n” nodes, we can save the space required for storing (n*(n-1)) routes.
The formulae for calculating total time taken for computing shortest route using normal approach and using our approach is shown in table 3. To prove the correctness, randomly 2000 nodes were selected and equated using the equations in table 3.

<table>
<thead>
<tr>
<th>Table 3: Formulae Used in the Proposed System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation time using normal approach = Computation time for query by executing algorithm Mumbai Navigator</td>
</tr>
<tr>
<td>Computation time using our approach = Database access time + searching time (if found), Database access time + searching time + execution time for algorithm + storing time (if not found)</td>
</tr>
</tbody>
</table>

5. SIMULATION AND RESULTS

Figure 6 shows shape file used for simulation and experimental purpose. For simulating the algorithms the above shape file is expressed in terms of graph containing 4000 nodes. Since for computation of path by Floyd Warshall algorithm requires memory matrix creation of size 4000*4000 which is not supported by JAVA therefore we have converted it into a sparse matrix. By doing so, computation time increases drastically. The Sparse Matrix used here is internally represented as a hash map where combination of row and column makes a key and matrix content is the value. So, to find a value of (row, column) = (i, j), we have computed the key corresponding to i and j and then shows the value that this key maps. We have randomly taken 500 source and destination pairs for computing shortest route and then taken the average of their time of computation for comparison.

Figure 6: Shape File for Simulation and Experiment

The space and time complexities of our proposed Route Finder System are compared with the space and time complexities of Mumbai navigator [8, 9].

230
Figure 7: Time complexity in Mumbai Navigator and Route Finder System

Figure 7 shows how average time complexity varies in our route finder system whereas it is constant in Mumbai navigator. From graph it is clear that our route finder system is better than Mumbai navigator. As, in our route finder system, after certain number of routes generated and stored in database, there is no need for computing path from algorithm hence the average time of computation restricts to route accessing and searching from database.

Figure 8: Space complexity in Mumbai Navigator and Route Finder System

Similarly, Figure 8 shows that for random 2000 queries normal approach would require 2000 route generation whereas our approach requires only 798 routes to be generated and stored. The remaining routes could easily be obtained from the stored ones thereby making space complexity less than that of Mumbai navigator approach.

5. CONCLUSION AND FUTURE WORK

The route finder systems in our approach minimized the number of computations needed for computing shortest route or alternate route from source to destination and the space and time complexities of Mumbai navigator are much greater than that of the our proposed approach Route Finder System. Hence our Route Finder System is efficient than Mumbai navigator. Future works includes GPS based data to be integrated into our system. This will help to understand peak times when the traffic is much congested thereby enabling less utilized road to be maximum utilized. It also includes density based transport management system approach to be integrated into the system.
REFERENCES


