Shortest Path problem in the context of Route Guidance Systems

Benjamin Vincent¹, Philippe Lacomme², Ren Libo³, Nikolay Tchernev⁴

¹ Université Blaise Pascal (IMOBS) Campus des Cézeaux, 63177, Clermont-Ferrand (France)
² benjamin.vincent@isima.fr
³ Université Blaise Pascal (LIMOS UMR 6158) Campus des Cézeaux, 63177, Clermont-Ferrand (France)
⁴ placomme@isima.fr
⁵ Université d’Auvergne (LIMOS UMR 6158), Campus des Cézeaux, 63177, Clermont-Ferrand (France)
⁶ ren@isima.fr, ⁷ Nikolay.Tchernev@udmall.fr

Abstract - The shortest path problem is a well know routing problem which received a considerable amount of attention for several decades. In a mobile environment, specific constraints must be taken into account to define an efficient system that is targeted to the field of pedestrian mobility in an urban environment. Such a definition consist in defining new algorithms taking into account for example the computational time constraints which is, in client-server system, a real issue to promote a service quality. This is also closely connected to the architecture of the web service which must be defined according to some basic SOA well-known rules. The architecture we tune for the project is based on an active monitoring system which dynamically requires new shortest path using a SOAP API defined on a server. This research is the first step in definition of efficient routing algorithms tuned for mobility environment.

I. INTRODUCTION

Optimization is a key paradigm for modeling and solving in Operations Research (OR) and in related aspects of engineering, science, economics, and business. There are numerous optimization problem classes including planning, scheduling, routing, etc. All these problems can be solved by means of different methods like linear programming or metaheuristics as well as using latest Information Technology (IT). As stressed by Mike Trick in his widely read blog (http://mat.tepper.cmu.edu/blog/?p=301), wrote, “Within OR, we often don't track IT concepts such as Service Oriented Architectures or business intelligence, but we should: it can have a great effect on how our work is used in organizations.”

A. Context of shortest path

In recent years there has been a renaissance of interest in the shortest path problem for use in various transportation engineering applications. Without any doubt it could be directly attributed to the recent developments in Intelligent Transportation Systems (ITS), particularly in the field of Route Guidance System (RGS) and real time Automated Vehicle Dispatching System (AVDS). In both cases, there is a definite need to find the shortest paths from an origin point to a destination one in a quick and accurate manner.

In the above applications, the traditional optimal shortest path algorithms cannot be often used because they are too computationally intensive to be feasible for real-time systems.

A number of heuristic search strategies have been developed for increasing the computational efficiency of shortest path search.

Most of these heuristic search strategies came from the artificial intelligence (AI) field including but not limited to [1], [2], [3], [4], where the shortest path problem is used as a mechanism to validate the effectiveness of these heuristics.

The current RGS field in both North America and Europe have generated renewed interest in using heuristic algorithms to find shortest paths in a traffic network for real-time vehicle routing operations. [5] discussed how heuristic search methods could be used in vehicle navigation system. [6] debated applications of an A* algorithm (called the force driven method in his paper), a bi-directional search method, and a hierarchical search one.

Since then, an huge number of researchers have followed the track and tried to introduce an worldwide strategy for improving the efficiency of the shortest path search process. These efforts have resulted in a large literature including a wide spectrum of search strategies and mechanisms.

B. Context of web services

Web Services connect servers (computers) and clients (computers or devices) together using the Internet to exchange data and combine data in new ways. Web services can be defined as software objects that can be assembled on the Internet using standard protocols to perform functions or execute business processes.

A web service could also defined as software running on a server, providing a service and exposing a set of functions or methods. The description must be formal and explicit in such a way that the client can easily interact with the system. The customer must be able to discover and understood the service description to interact with the web service according to its definition using XML (eXtensible Mark-up Language) based message conveyed by Internet Protocol. Web services are characterized by their great interoperability and extensibility thanks to the use of XML, and they can then be combined in a loosely coupled way in order to achieve complex operations [7]. During the last decade several technologies have been used including: XML-RPC (Remote Procedure Call), REST...
(REpresentational State Transfer [8]) and lately SOAP (Simple Object Access Protocol [9]).

Basically, a web service must accept request, perform execution of one or several functions according to the request received and return a result. The definition of a web service cannot be achieved without a careful analysis of the Security (illegal access, abnormal and non-reasonable use of the service...), analysis of the Quality of Service (QoS) and Management.

As stressed by [7] “Basically, the security problems that are likely to affect Web services are the same as those that have affected the conventional Web-based systems.” After a literature review the authors summarized that the security is critical to the adoption of Web services by enterprises.

II. PROPOSAL FOR A ROUTE GUIDANCE SYSTEM (RGS)

A. System architecture for conception

The definition of the system architecture is based on the couple Netbeans/JEE for the conception and uses the well known Glassfish application server for deployment. The Java server implementation takes advantage of the EJB (Entreprise Java Bean) and of the SOAP Web Service definition which is compliant with the JEE standard development.

The SOAP web service exposes five methods:

- Demarrer_Guidage which is an asynchronous method which starts the shortest path computation on the server;
- TesterEtatDemande which permits to check the state of the method Demarrer_Guidage returning 1 if the job is in progress for example;
- RecupererEtatGuidage which permits to download the shortest path as an ordered set of (latitude/longitude) positions;
- LireCodeRetour which permits to obtain a message from a code returned by the other methods;
- RecupererGraphe which permits to download a graph around a given position as a set of vertices and a set of arcs.

![Figure 1. Iterative shortest paths execution over time](image)

B. API restriction

A public API is a provided and a required definition of a strict security policy and a well-tuned control of usage trying to avoid non rational use and by induction trying to guaranty fairness between users. One commonly basic trend consists in conditioning API access to a key delivered after filling out a detailed questionnaire where an email address is mandatory in order to send a mail with the key requested by a scientist who wants to use the API.

This solution has been kept and a key can be request at:

III. PROPOSAL FOR AN EFFICIENT GUIDANCE SYSTEM DEFINITION

A. Creation of a graph to model the real live graph

The first condition to compute a shortest path is the possibility to obtain a map or to obtain the part of the map required for computation. The second condition consists in mapping the map into a dedicated graph with a well suited data structure favoring computation efficiency of algorithms. The API provided by OpenStreetMap has the great advantage to be free of charge and to provide both png map and data on the road map.

B. Definition of a dedicated shortest path

The solution we promote takes advantages of three considerations:

- Branch pruning techniques are considered and consist in discarding nodes from the scan eligible node list after proving to be located out of the assumed feasible solution area. The branch pruning algorithm maintains a strongly limited set of eligible nodes as compared to A* and provided a better complexity.
- It has been commonly recognized that the computational effort required for solving a generic search grows faster than the size of the problem. However if the original problem can be decomposed into small sub-problems a substantial computational time saving can be achieved.
- A sub-goal based method can be defined into a sub-problem especially for shortest path problem in a road traffic network where goals could be nodes or links located between the origin and destination location.

With such an approach it is possible to obtain a dynamic system where shortest paths are consecutively computed depending on the user progress in the road map and taking into account efficiently road hazard including but not limited to inaccessible routes and user errors.

Let us consider the graph of figure 1 where an end-user required assistance to route from node 43 to node 8. First the Euclidean distance is computed including the direction which is represented in blue on figure 1 and second, the algorithm extract a sub-graph G’ limited to node 43-44-35-36 and third compute a shortest path from 43 to node 36. When the end-user reaches the node 36, a new shortest path is computed into the next sub-graph.

C. Integration of the shortest path algorithm

The web method Demarrer_Guidage is asynchronous: it closes the communication immediately and launches a long time process into a thread. The main advantage is that a client can continue its execution in very short delays.

The thread is linked to the client and more precisely to the API Key which is used as session identification. Into the thread the following operations are achieved:

D. Android demonstration

With the objective to validate the principle and the performance of the tuned shortest path algorithm in the real life situation, an android Application (figure 2) has been created and it can be downloaded at:

http://www.isima.fr/~lacomme/ORWebServices/GPS4pedestrian/source/Gps_android.apk

The application is compliant with any smartphone equipped with Android 3.0 or later and equipped with classical wide spread sensor including GPS.

Fig. 1. The RGS in a Wiko Cink Five smartphone

IV. SHORTEST PATH API DESCRIPTION

Using multilayered architecture, a Java client can be created using NetBeans with JEE plugin. A Java application encompasses a SOAP layer which is automatically generated by NetBeans, a business layer and a basic layer interface. Similar remarks hold for any conception except that numerous tasks in current professional environment are not so integrated and required more attention and time for conception. The
business layer can be divided into 3 steps to make a demonstration of the different method executions:

- **Step 1.** The client has to create an instance of the class RecuperationDonnee and to define a port assigned with getRecuperationDonneePort.
- **Step 2.** The client can access to the asynchronous DemarrerGuidage method providing a valid API key, and two (latitude/longitude) positions.
- **Step 3.** The client can periodically investigate if the result is available or not using the RecupererResultatGuidage method providing only the key which is the session identification.

The API provided offers to the community a free execution on a server and the possibility to reuse the methods for a very small programming effort as stressed in figure 3.

```java
webservice.WebServiceJobShop_Service service = new webservice.WebServiceJobShop_Service();
webservice.WebServiceJobShop port = service.getWebServiceJobShopPort();

double lat1 = 45.758451, long1 = 3.105476;
double lat2 = 45.776234, long2 = 3.891571;
String key = "7780458c95b922qbd215";
String rlt = port.demarrerGuidage(lat1, long1, lat2, long2, key);
jTextArea1.setText(jTextArea1.getText() + "resultat = " + rlt+"\n");

do{
    rlt = port.recupererResultatGuidage(key);
    try{
        Thread.sleep(1000);
    }catch(Exception E){
        break;
    }
}while (rlt.contains("#4#"));
jTextArea1.setText(jTextArea1.getText() + "resultat = " + rlt+"\n");
```

Fig. 3. Shortest path API usage in Java

V. NUMERICAL EXPERIMENTS OF THE RGS PERFORMANCES

A. Comparative study from the optimal shortest with the shortest path proposed

The objective consists in considering a theoretical situation where the end-user is not submit to any route hazard and so respects the indication and applies exactly the shortest path proposed iteratively by the system. As stressed in figure 4, the RGS computed only part of the path and each partial path permits to propose a global solution to the end-users.

To evaluate the overall performance in this theoretical situation, 20 instances have been generated using randomly selected positions in the city of Clermont-Ferrand.

The average gap is about 5% and can be considered as acceptable since 5% of path of average length about 3000 meters is not more than 150 meters.
Fig. 4. Shortest path iteratively computed in a mobile device Samsung Galaxy S3

<table>
<thead>
<tr>
<th>Instance</th>
<th>Shortest Path Dijkstra's Algorithm</th>
<th>RGS Shortest Path</th>
<th>Gap %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3629.80</td>
<td>3812.60</td>
<td>5.04</td>
</tr>
<tr>
<td>2</td>
<td>4509.11</td>
<td>5613.92</td>
<td>24.50</td>
</tr>
<tr>
<td>3</td>
<td>1639.40</td>
<td>1639.40</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>4047.20</td>
<td>4496.63</td>
<td>11.10</td>
</tr>
<tr>
<td>5</td>
<td>1866.19</td>
<td>1882.40</td>
<td>0.87</td>
</tr>
<tr>
<td>6</td>
<td>1245.82</td>
<td>1289.56</td>
<td>3.51</td>
</tr>
<tr>
<td>7</td>
<td>3278.60</td>
<td>3388.53</td>
<td>3.35</td>
</tr>
<tr>
<td>8</td>
<td>2737.98</td>
<td>2737.98</td>
<td>0.00</td>
</tr>
<tr>
<td>9</td>
<td>1525.79</td>
<td>1699.48</td>
<td>11.38</td>
</tr>
<tr>
<td>10</td>
<td>2417.21</td>
<td>3079.71</td>
<td>27.41</td>
</tr>
<tr>
<td>11</td>
<td>5440.46</td>
<td>5659.44</td>
<td>4.03</td>
</tr>
<tr>
<td>12</td>
<td>1789.43</td>
<td>1943.73</td>
<td>8.62</td>
</tr>
<tr>
<td>13</td>
<td>3383.98</td>
<td>3741.73</td>
<td>10.57</td>
</tr>
<tr>
<td>14</td>
<td>2750.92</td>
<td>2770.76</td>
<td>0.72</td>
</tr>
<tr>
<td>15</td>
<td>2529.81</td>
<td>2529.81</td>
<td>0.00</td>
</tr>
<tr>
<td>16</td>
<td>1260.14</td>
<td>1260.14</td>
<td>0.00</td>
</tr>
<tr>
<td>17</td>
<td>2074.18</td>
<td>2137.08</td>
<td>3.03</td>
</tr>
<tr>
<td>18</td>
<td>2588.66</td>
<td>2588.66</td>
<td>0.00</td>
</tr>
<tr>
<td>19</td>
<td>3033.87</td>
<td>3094.42</td>
<td>2.00</td>
</tr>
<tr>
<td>20</td>
<td>3103.75</td>
<td>3703.95</td>
<td>19.34</td>
</tr>
<tr>
<td>Avg.</td>
<td>2742.62</td>
<td>2953.50</td>
<td>6.77</td>
</tr>
</tbody>
</table>

TABLE II. RGS PERFORMANCE WITH DIFFERENT PROBABILITY OF WRONG TURNS
The crossroads of optimization research community and the web service community expectations.

**REFERENCES**


