

Software Solution for Monitoring Street Traffic and Generating Optimum Routes using Graph Theory Algorithms

M. Moise, M. Zingale, A.I. Condea

Maria Moise, Marilena Zingale, Alexandru Ioan Condea
Romanian American University, Romania
E-mail: maria.moise@rau.ro, zingale@un.org, zander.aq@gmail.com

Abstract: Nowadays, big cities are facing traffic jams, generated by the great number of automobiles in regard to the limited infrastructure capacity. Drivers are being presented with these problems: increased time spent between areas of interest, a higher risk of having an accident and of course stress suffering. In order to solve the urban traffic-jam problem a number of solutions have been developed. One of these is TomTom , which offers, free of charge, the possibility to generate navigation indications for a route. Unfortunately, the traffic monitoring service is limited to a few countries, but some countries are not on their coverage area. At this time there isn't a complete method to calculate the optimum route from a destination to another, taking into account street traffic. The only way to get relevant information is represented by the drivers personal expetraffic jams a series of solutiontraffic jams a series of solutionrience and the news on TV/radio. Thus, the choice for generating an optimum route is up to the driver/client and, as a consequence, this method is not a scientific one, being certified only empirically.

In this context, the paper presents a software solution, which determines the optimum route, taking street traffic into regard, thus contributing to a substantial reduction of time spent in traffic by drivers. The information needed for the application regarding the state of the street traffic can be supplied by the agents that check all available information sources(news bulletins, radio, police announcements) and the mobile agents that patrol the streets. The aim of this paper is to present a solution for determining the optimum route choice for cars. The solution is composed of two applications: First is MapMaker, which designs a street map. Second is BestRoute which can add traffic coefficients to streets and calculate the optimum route. In order to choose the best route two criteria are used: the minimum distance and the street traffic coefficients. The data regarding the streets map and the traffic situation is taken from a MySQL database; the optimum route from destination A to the destination B is calculated using a modified Dijkstra algorithm.

Keywords: optimum routes, graph theory, Dijkstra algorithm.

1 Introduction

Today most big cities are facing traffic jams that are generated by the exponential increase of automobiles versus the original limited traffic flow design. In order to solve the urban traffic jams a series of solutions have been proposed and used worldwide. One of these is TomTom , which offers free of charge the possibility to generate a navigation path on their website. Unfortunately, the traffic monitoring service is limited to a few countries, and Romania is not on their supported list.

In Romania such a solution has not been developed yet; existing solutions in Bucharest are limited to the synchronization of the traffic lights and the existence of a reduced number of intelligent traffic lights.

At this time a complete solution to calculate the optimum route from a destination to another, with the traffic taken into account, does not exist. The only information available represents the personal experience and/or the news on TV or radio. Thus calculating the optimum route is up to the driver(client) and as a consequence it is certified only empirically.

In this context, the paper presents a software solution, which determines the optimum route, taking street traffic into regard, thus contributing to a substantial reduction of time drivers spend in traffic. The information needed regarding the state of street traffic can be acquired either by agents that check all the available information sources or by agents that patrol the streets.

2 Solutions description

The solution for determining the optimum route is composed of two applications. First is MapMaker a map designer with the ability to use a background satellite image for easier plotting. For example, it could use a selection of satellite images from Bucharest, covering the Doamna Ghica area and the adjacent neighborhoods. After the background image is selected the map is designed node by node with the use of the mouse. The streets are made by connecting the desired nodes. The resulting map can be saved both on the local hard-disk and on an external MySQL database. The second application is BestRoute, used to set traffic coefficients and calculate optimal routes. The necessary data is taken from a MySQL database.

The application has two modes of usage:

- the client mode, which is used to calculate the optimal route based on the selection of two nodes. The result is shown both graphically and in text mode;
- the admin mode, which is used for altering the traffic coefficients of a street. In this mode, you can add, modify and delete user accounts. This mode also provides a set of reports about customers or streets.

The BestRoute application uses elements and algorithms from the graph theory. A modified Dijkstra algorithm has been used in order to calculate the minimum distance path between two nodes of the graph. The features of the algorithm's Dijkstra are the following:

- there are established a list of distances, a list of previous nodes, a list of visited nodes and a current node;
- the values from the list of distances are initialized with an infinite value, excluding the home node, which is set with value "0";
- all values in the list of visited nodes are set with value "false";
- all values in the list of previous nodes are initialized with the value "-1";
- the home node is set as the current node;
- the current node is marked as visited;
- the distances are updated based on nodes that can be viewed immediately from the current node;
- the current node is updated to the unvisited node, which may be visited by means of the shortest path from the home node;
- to be repeated (from point f) until all nodes are visited.

Formalization of the Dijkstra algorithm consists of:

Step 1. Initialization of the initial peak having a value of 0: $w(X_0) = 0$;

Step 2. Setting up the lot A comprised of the initial node: $A = X_1$;

Step 3. Analysis of nodes outside the lot A, namely:

i. If the nodes can be reached by direct arcs from nodes in A, then for these nodes, we calculate:

$$w(X_i) = \min(W(X_j + V(x_j, x_i))) \text{ for problems of minim} \tag{2.1}$$

$$X_j \in A$$

$$\exists(X_j, X_i)$$

And it is added to the lot A only that node for which the minimum value is obtained, then step 4 is initialized.

Step 4. Analysis of A multitude:

i. If $X_n \in A$, then its value represents the value of the optimum value path from X_1 to X_n ; in order to find this path we start backwards from the final node x_n and we find the node $X_{k_1}, X_{k_2}, \dots, X_{k_r}$ q which form the path searched for, where $X_{k_1} = X_n$,

$X_{k_r} = X_1$, and each other index k_{i+1} is the one for which:

$$w(\cdot) + v(X_{k_{i+1}}, X_{k_i}) = w(X_{k_i});$$

STOP.

ii. If $X_n \notin A$, the algorithm is resumed from step 3.

$$w(\cdot) + v(X_{k_{i+1}}, X_{k_i}) = w(X_{k_i});$$

STOP.

iii. If $X_n \notin A$, the algorithm is resumed from step 3.

The application BestRoute uses data from public sources and its own agents. On its execution the program queries the MySQL database and retrieves the map and its traffic coefficients, then it shows a color coded visual representation of the traffic map. Upon requesting a route from address A to address B a modified Dijkstra algorithm is used. It is also possible to select whether one wants the fastest route (traffic wise) or the shortest route (geographically). This application also has the possibility to generate reports on customers (useful for the administrator) and on the state of the streets. The reports generated by the application can be exported in Excel worksheets and/or PDF document format. For generating reports Microsoft Report Viewer component within the NET framework was used and the reports were created using the Report Wizard plus a manual tweak for better presentation of data.

3 Solution output presentation

3.1 MapMaker

In fig. 1 we have the main screen.

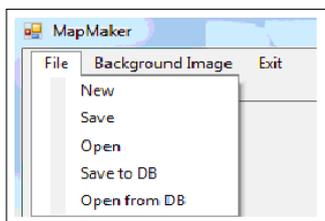


Figure 1: Main screen of module MapMaker.

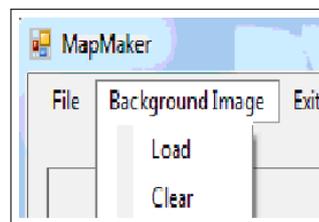


Figure 2: The menu Background Image.

By clicking on File we can:

- Create a new map - New;

- Save the existing map - Save;
- Open a saved map - Open;
- Save the map to the database - Save to DB;
- Open a saved map from the database - Open from DB.

Selecting a background image from the hard-disk is done by clicking on Background Image (fig. 2).

To design the nodes, we use the buttons from fig.3, which enable:

- Add a new node to the map - Nod nou;
- Show details about a selected street - Detalii muchie;
- Delete a selected node - Sterge;
- Clear the map scale - Clear Scara.

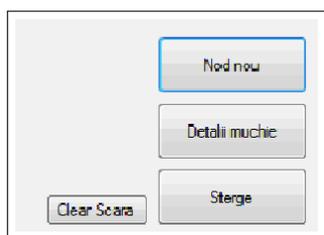


Figure 3: Buttons for creating, modifying and deleting nodes and clearing the maps scale.

Also MapMaker uses a system which automatically determines the scale for the map. It is necessary to introduce the distance between nodes only the first time. The calculation is done by the ratio between the initially introduced distance and the actual pixel distance between the two points.

3.2 BestRoute

The applications main screen is illustrated in fig. 4.

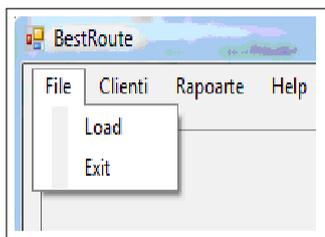


Figure 4: Options of the File menu.

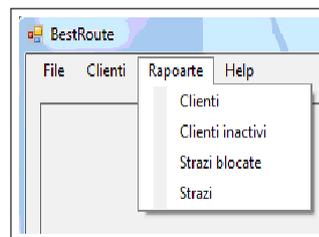


Figure 5: Options of the Reports menu.

By clicking on File we can download map and user data from the server or quit.

By clicking on Clienti while in administrator mode one can add new users, modify old ones, delete them or set whether a specific user is restricted or not.

By clicking on Reports (fig.5) one can request reports on clients, restricted clients and street traffic.

On opening the application, the system displays the map in img. 6, and selecting the start node is illustrated in fig. 7.

Generating a non-optimized route in fig.8;

After selecting the destination node the application calculates the shortest path available.

Optimized route provided by the application is illustrated in fig.9.

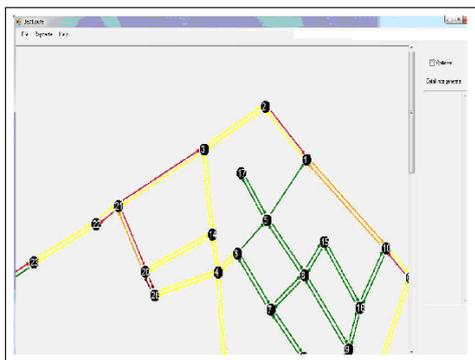


Figure 6: Initial state of the Map on opening the application.

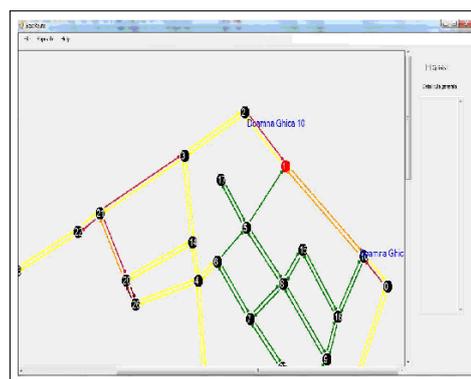


Figure 7: Selecting the start node.

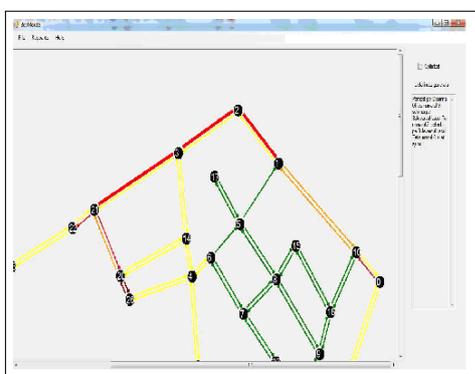


Figure 8: Image (red one) of non-optimized route.

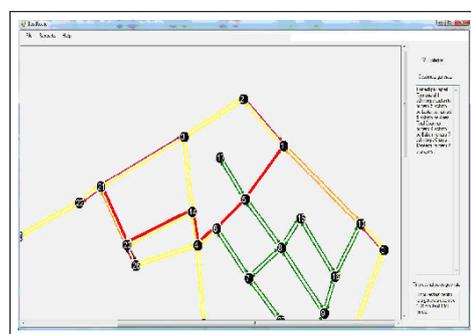


Figure 9: Image of optimized route, enabling avoiding crowded streets.

If the optimize option is checked the application will generate an optimum route to avoid heavy traffic areas. Also a time estimate is offered.

For calculating the optimal route, the following distance modifiers based on traffic coefficients were used: 1, 1.25, 1.66, 2.50, 5 and 999 for closed streets.

To calculate the estimated time, the following maximum speeds were assumed: for excellent traffic conditions a speed of 50kph, reducing speed by 10kph to the minimum of 10kph for very bad traffic. Thus, a relatively accurate time can be calculated for the generated route. For the best estimation possible, a interval of plus/minus 20% is assumed.

On both generating options a street by street text solution is also supplied.

4 Conclusions

Using such an application by drivers leads to avoiding stress and fatigue generated by traffic jam, thus reducing car crash risks. Also, the application provides information on routes unknown to drivers thus leading to a better awareness of the city. Implementing the application for a taxi or courier company offers an edge on competitors, generating a shorter delivery or reply time. On the whole, large-scale use of the application in partnership with the town hall and police may lead to general fluidization of traffic, equally improving the environment.

Bibliography

- [1] Moise, M., Zingale, M., Condea, A., *Informatics application which determines the optimum routes for the cars*, in Proc. of the E-COMM-LINE 2009 Conference, Section V 35, pp. 5.
- [2] Moise, M. *Data base informatics systems*, Prouniversitaria Publishing House, 2008, Bucharest
- [3] *** <http://msdn.microsoft.com> - Microsoft Developer Network
- [4] *** <http://stackoverflow.com/> - StackOverflow
- [5] *** <http://dev.mysql.com/usingmysql/dotnet/> - Documentation MySQL