Best Route Determination in Real World Traffic Using Temporal Statistics of Roads and Vehicle-Specific Characteristics

Mohammad A. M. Asmaran, Sulieman Bani-Ahmed

IT Department, A-Balqa Applied University
Prince Abdullah Bin Ghazi Faculty Of Science and IT, Al-Balqa Applied University, Al-Salt 19117, Jordan
m_asmaran@hotmail.com, sulieman@case.edu

***Abstract*— Route planners that identify the *best* step-by-step driving directions between two locations have rapidly become a part of our everyday life. However, the output *quality* of route planners depends critically on the current status of roads in order to calculate the best (*shortest, fastest, or cheapest)* route from a starting point to a destination. Maintaining accurate and updated statistics about the status of roads has been made possible due to the integration of navigational technologies. These technologies are the global positioning system (GPS), database technologies such as geographic information system (GIS) and communication technology. This paper is to propose an efficient self learning mechanism or framework to help rout planners find out *best* routes based on the *current status* of roads in addition to the static road lengths. This helps customizable best rout identification. With the proposed framework, it is possible, for instance, to choose the best route that the minimal time in addition to minimal fuel consumption, which is a very critical factor due to the significant increase in fuel prices in the last few years. An advantage of the proposed framework is that is does *not* require any extra streets equipments. Further, the proposed framework promises a system that requires minimal administration.**

***Keywords*— Route planners; Best path algorithms; GPS; Traffic tracking; Shortest Path.**

I. INTRODUCTION

Many real life applications relying on finding the *best* way to reach specific destination point in a GIS map. This implies using a best path algorithm that can find out a best path according to certain measures [1]. Visualizing the GIS map as a graph, these measures are usually referred to as weights of the edges of the path, where an edge is the link between two linked points in the GIS map. These weights actually control guide the way that best path algorithm can run and identify the best route. The weights represent the lengths of the edges and, thus, the best route can be thought of as the shortest path [2].

Depending on the criteria represented by the weights of the edges, the best path can be viewed as the one with the shortest overall travelling time or the one with shortest path length. Other optimization aspects are also possible, such as the total fuel cost. The best route identified may vary depending on the obstacles that vehicle could face in its way to the destination. These obstacles could be jams, accidents, traffic lights, or maintenance operations in streets. In some cases, these obstacles would make the shortest path in terms of length to consume a lot of time and/or more fuel cost when compared to following a longer alternative route [3]. Consequently, the weight of the path or the edge should be computed according to more factors such as jams that are one of the most affecting factors in term of time. Another factor is the number of traffic lights in roads and the stopping times of traffic lights. In practice, traffic lights are categorized in to four categories according to the way they are controlled [4]:

A. *Fixed time traffic lights*

In this case traffic lights are controlled by a timer that indicates when traffic light should allow vehicles to move or stop. This doesn't mean that a given traffic light assign the same open/close times for each direction on the same cross of streets. Main streets usually assigned longer time period in green lights and shorter time period in red lights because of the fact that they contain more vehicles moving on. Of course the opposite is true regarding sub-streets. In some cases this opening/closing time may vary depending on the time of the day in order to handle peak hours. Such type of traffic light relies on some statistical measures that are used by traffic agencies to assign proper opening/closing times for each traffic light. This type of traffics is very effective in terms of time measurements when used by best path algorithms because that time of traffic is predictable and deterministic [4].

B. *Intelligent traffic lights*

In this case traffic lights are controlled by an intelligent control system that controls the traffic according to some facts read from sensors mounted within the traffic. Sensors provide some information to the control system in order to control traffic time. For example, sensors can provide the control system with the number of vehicles that currently stopping in each street lane, which is an indicator of the size of the jam on a certain street lane. Another approach is to use image processing algorithms to find out the jam size on a certain traffic light. In this type of traffic lights, time can not be predicted even if the system is connected to main traffic control system because it depends on real time data that can't be predicted before [4].

C. Manual traffic lights

In this type of traffic lights, traffic lights are controlled by human operator who is sitting in a control room and watches real traffic using cameras attached on the traffic lights. These cameras allow him/her to determine which street(s) to be allowed to go and for how long. The final goal is to reduce traffic jams. This type of traffic lights is very effective in real time traffic system because of the intelligence of people but its main problem is that no one can expect or determine the time of traffic light [4].

D. Multi-Control Traffic light:

In this type of traffic lights traffic agency operator would gain the control according to some circumstances in order to solve some huge jams or emergency situations [4].

Actually traffic agencies could use all the above types in the same city or country which could lead to inaccurate time calculations and predictions. Furthermore, accidents, maintenance operations, or other obstacles would consume unpredictable time intervals. Predicting these time intervals is a very critical input to route planners as we said earlier. Real time measurement of these time intervals requires using sensors, cameras, or humans. These traditional techniques are not usually available in every street or situation and they may have expensive installation and maintenance costs [3].

II. Related Work

Many algorithms are proposed to manage streets congestions to provide efficient traffic system [5]. These algorithms such as RFID tags approach [6]. This approach evaluates congestion level using some sensors to evaluate street congestion using speed facts and car counting according to RFID sensors that sense the RFID tag attached to each car. This approach is used to control traffic lights in order to minimize the number of cars moving in a specific street and, thus, minimize the overall waiting time.

Another approach proposed in [7] to measure congestion using image processing techniques. This approach is used to control traffic lights by measuring congestions using cameras that capture images. These images are feed to an algorithm to be analysed in order to get car counts that are considered to be valid measure of the congestion level. These algorithms involve installing equipments in the streets in order to collect data from specific locations. This approach increases the overhead of maintenance and administration. Furthermore, required equipments are subject to errors according to some inefficient working conditions such as weather conditions (e. g. cameras would provide unclear pictures in case of rain or dust storms).

Many algorithms are proposed to forecast traffic volumes. The major concept used in these algorithms is to use historical data statistics in the same conditions (i.e. same dates, times, and weather circumstance) in order to predict what will traffic look like in future [8, 9]. The main problem of such algorithms is that they may fail in unpredicted situations such as accidents. Using statistical data provides clear general view according to same conditions in previous situations. But if some accidental situation occurs, statistics results will be very far from actual traffic situation. In addition to the above drawbacks, this approach implies the administration overhead of collecting data in order to determine accurate statistical results.

In this paper, short-term intelligent statistical view of moving vehicles is used for efficient *current* edge weight computation. The proposed framework promises more accurate and customizable route determination. This is achieved through using GPS systems mounted to vehicles currently moving in a given road to report statistics that will be later used for edge weight computation. This way, the *current* experience of vehicles is used for computing up-to-date edge weights. The vehicles themselves are used as *agents* to improve the performance of the route identification algorithm for the benefit of all the vehicles in the traffic system.

Another approach has been proposed to predict travel time interval and congestions using ANT Colony algorithms [10]. In [10] edge probabilities are represented by the time the vehicles spend in order to go to the destination. ANT algorithm provides efficient way to decrease time; however, it calculates travel time using sensors attached in every path. This implies higher cost and maintenance overheads. Moreover, calculating edge weights is a blind process and, thus, fails to take into consideration *irregular* vehicles movements if used with real world traffic. Another use of ANT Colony based algorithms is available in [11] which provides more efficient way to take into consideration the weather conditions. Weather conditions and past experience both provide more accurate expectations. Similar to the work in [11], accidental situations are not considered.

In this paper, our proposal provides a complete solution in order to use vehicle equipments (i.e. Global Positioning System (GPS) Navigator). Such devices are already provided in most of modern cars. Moreover, our proposal provides an efficient way to consider fuel consumption in computing edge weights. The fuel consumption factor is very important as fuel prices increase almost every day. In our approach, fuel consumption is not only a function of time and distance. Rather, the type of street (be it highway or not) and the total number of vehicle stops both affects the fuel consumption especially in hybrid cars.

Another algorithm is introduced in [12] to generate cost function using topological and historical data in order to get best cost estimation. This algorithm assumes that statistical data is available for various day time periods in addition to streets topology. The performance of this algorithm is sensitive to real time changes such as accidental situations and is negatively affected by the lack of statistics in some regions and streets. In this algorithm no human interaction is needed except in defining very basic information that is available in vehicle specification manuals and streets connections which is obvious to be available for such algorithms.

III. Proposed algorithm

Computing the expected travelling time of a car in a given road can't be computed in a very accurate way. However, estimating this time is a very important factor in weighting edges of the path. This paper proposes a solution in order to find more accurate measures of the weights that could properly guide best path algorithms in determining the best path.

The proposed solution depends on measuring weights according to distance, obstacles, and traffic situations. In order to do that, obstacles would be represented as a weight value that indicates how much the obstacle would affect the edge weight. This value is computed according to the time.

In our proposal, travelling time is considered as an indictor to the presence of the obstacles in the street or edge as follows: if the moving vehicles that have already entered a specific road are wasting a lot of time in moving through that road, the corresponding edge in the graph is assigned a higher weight. Travelling time of a road is highly affected by the number and severity of obstacles in that road. Consequently, edges with longer travelling times are to be assigned higher weights.

We propose the following approach in order to dynamically compute the current travelling time of roads. Travelling time is divided into two components, namely; the "waiting time" and "walking time" components. Waiting time is stopping time of the vehicle. This is the total time during which that vehicle is *not moving*. Practically, "*not moving"* means that the vehicle speed is near to zero or below a certain threshold. In our implementation, this threshold is assumed to be 1 km/h. This threshold depends on multiple factor and may be changed, some of which is the nature of the country or the city that this algorithm is applied to. The second type of time is the walking time which is the total time during which the vehicle is moving (its speed is above the given threshold).

A. *Edge Weight Computations*

Each edge has three weight values to be computed according to the following equations:

1) *Length Weight:* This weight is computed according to the speed low so that length = (Average Speed of the Vehicle) X (Walking Time).

2) Cost Weight: This weight is computed according to manufacturer specifications of the vehicle according to the following equation:

Cost = (length weight) X (Car Fuel Consumption Rate Per Unit of Length) + (Waiting Time) X (Car Fuel Consumption Per Stopping Unit of Time).

The above Two factors (Car Fuel Consumption Rate Per Unit of Length and Car Fuel Consumption Per Stopping Unit of Time) are manufacturer specifications specified in the car specifications manual. Car Fuel Consumption Rate Per Unit of Length has three values according to the type of street that vehicle is moving through. These values differ for Highways, Semi Highways, and none Highways [13, 14, 15].

3) Time Weight: This weight is computed according to the total time that vehicle elapsed in order to go through the edge which is Total Time = Walking Time + Waiting Time.

So, each edge is assigned three weights (Length, Cost, and Time). If the user of the best path system needs to go through shortest length path, best path algorithm will use length weight for each edge. If the intended path is the lowest cost path, cost weight will be considered, and the same in case of lowest time where time weight will be considered. So, weight of the edge will vary according to user need in order to match his requirement to get best solution.

B. *How to measure edge time*

According to previous description regarding traffic lights systems, it is clear that time can't be predicted, and emergency obstacles in streets such as accidents, maintenance workshops, etc. are not predictable either. So, the best way is to measure the current travelling time of individual roads. Time is measured by using two components as we said earlier: one to measure stopping time and another one to measure walking time. Notice that this time is measured for the vehicles that are currently in the road. This means that computed time values are very late for the vehicles already in the road. However, these values will be used by other vehicles welling to go through the same edge.

The proposed algorithm is designed to use data collected from currently moving vehicles to weight edges and provide guidance to any vehicle going to pass a given edge at any certain moment.

C. *Speed Measurement:*

Speed can be measured using normal speed equation by finding the difference between two vehicle locations using GPS system and divide it by time period that it spent in order to move between them. In fact, proposed solution considers the use of special tracking equipments that can send car speed in addition to the GPS location to the server [16]. Server should compute the average speed according to the number of reads along edge.

D. *Algorithm Steps:*

Step 1: Each vehicle should attach a processing unit that communicates with server in order to send information such as speed and GPS position. All map information, mainly edges information are stored on the server as shown in Table I.

TABLE I. EDGE REPRESENTATION DATABASE TABLE

Edge ID	Highway Level

Edge ID is a unique identifier of the edge and Highway Level is value that describes type of the edge whether it is Highway, Semi Highway, or None Highway. Highway Level column is used to help calculates Cost Weight.

Moreover, vehicle information should be stored such as its fuel consumption rates as shown in Table II.

TABLE II. EDGE REPRESENTATION DATABASE TABLE

Vehicle ID	Highway Fuel Consumption Rate	Semi Highway Fuel Consumption Rate	None Highway Fuel Consumption Rate	Stopping Fuel Consumption Rate

Using Table II system will be able to compute fuel cost of the edge as described before in cost weight equations where Vehicle ID is a unique identifier of the vehicle. Consumption rates are found in the specifications manual of the vehicle or they can be known by experience.

Step 2: Server collects data from vehicles and registers this information with calculated stopping and walking times depending on the vehicle speed as described before. All this information is stored as shown in Table III.

TABLE III. EDGE PASSING VEHICLES VALUES

Edge ID	Vehicle ID	Stop Time	Walking Time	Average Speed	Time	No Of Familiar Vehicles

As shown in Table III each vehicle passes an edge will has a record that contains its measured statistics in addition to the time in which this record is stored.

Step 3: Statistical data are computed every predefined period of time and based on the Table III data and stored as shown in the Table IV.

TABLE IV. STATISTICS TABLE

Edge ID	Average Stop Time	Average Walking Time	Average Speed	Calculation Time

Table IV contains average statistical data computed every period of time which is assumed to be in minutes (e. g. 10 minutes).

Step4: Edge weights are computed according to Table IV values using average stopping and walking times in addition to the average speed. Table II data are used to calculate Cost Weight in addition to the values computed in Table IV.

E. Average Calculation Details:

As seen from above steps, average values should be calculated every period of time and they are calculated based on the values registered during last interval period only. But in real life traffic, some vehicles will be odd in term of their walking or stopping because of the nature of their journey or the special conditions they have met. So some errors will arise if their values are considered in average calculations. These values should be ignored. In order to ignore them, algorithm defines threshold values that must not be exceeded between time and speed average values calculations. Consider values as shown in Table V.

TABLE V. SAMPLE STATISTICAL COMPUTATIONS

Edge ID	Vehicle ID	Stop Time	Walking Time	Average Speed
1	1	20	100	43
1	2	18	98	45
1	3	23	105	40
1	4	200	300	15

All values in the table look like to be close to each other except values determined for vehicle number 4. It is clear that this vehicle has none logical values comparing to other vehicles values. So, its values shouldn't be considered in average calculations. In order to do that, threshold value is defined for every column as the following example:

Stop Time Threshold Value = 10
Walking Time Threshold Value = 15
Average Speed Threshold Value = 10

Values differences should not exceed above threshold values for every column (i.e. Stop Time, Walking Time, and Average Speed).

When new vehicle values are received, they are checked across all interval vehicles. If the absolute value of the difference doesn't exceed threshold value. No of Familiar Vehicles column should be incremented by one for the corresponding vehicle and new vehicle.

Initially "No Of Familiar Vehicles" should be one. At the end, table should contain each vehicle value and the number of familiar values related to it in the same interval. Table VI, VII, VIII, and IX show steps calculation of the previous example:

TABLE VI. FIRST VEHICLE VALUES DURING INTERVAL

Edge ID	Vehicle ID	Stop Time	Walking Time	Average Speed	No Of Familiar Vehicles
1	1	20	100	43	1

TABLE VII. SECOND VEHICLE VALUES DURING INTERVAL

Edge ID	Vehicle ID	Stop Time	Walking Time	Average Speed	No Of Familiar Vehicles
1	1	20 **(diff. is 2)**	100 **(diff.. is 2)**	43 **(diff. is 2)**	2
1	2	18	98	45	2

TABLE VIII. THIRD VEHICLE VALUES DURING INTERVAL

Edge ID	Vehicle ID	Stop Time	Walking Time	Average Speed	No Of Familiar Vehicles
1	1	20 **(diff. is 3)**	100 **(diff. is 5)**	43 **(diff. is 3)**	3
1	2	18 **(diff. is 5)**	98 **(diff. is 7)**	45 **(diff. is 5)**	3
1	3	23	105	40	3

TABLE IX. FOURTH VEHICLE VALUES DURING INTERVAL

Edge ID	Vehicle ID	Stop Time	Walking Time	Average Speed	No Of Familiar Vehicles
1	1	20 **(diff. is 180)**	100 **(diff. is 200)**	43 **(diff. is 28)**	3
1	2	18 **(diff. is 182)**	98 **(diff. is 202)**	45 **(diff. is 30)**	3
1	3	23 **(diff. is 177)**	105 **(diff. is 195)**	40 **(diff. is 25)**	3
1	4	200	300	15	1

As shown in the above example vehicle 4 is neglected because of its irregular values.

Once the interval end is reached, average is calculated according to the vehicle having the maximum "No Of Familiar Vehicles" column value. This is done by finding the familiar vehicle values related to it and dividing their summations by "No Of Familiar Vehicles" column value. Table X shows above example results.

TABLE X. FINAL STATISTICAL COMPUTATIONS

Edge ID	Average Stop Time	Average Walking Time	Average Speed
1	(20+18+23)/3 = 20.3	(100+98+105)/3 = 101	(43+45+40)/3=42.7

F. Complexity

Average calculation process complexity is O(n) where n is the number of values read during interval.

Insertion process complexity is O(m) where m is the number of previously inserted values from the beginning of the interval. [17]

G. Determining "Highway Level" column

"Highway Level" column varies according to interval average speed so that if it is more than 80 km/h, then Highway Level is 3. If the interval Average Speed is between 50 km/h and 80km/h, then Highway Level is 2. Otherwise Highway Level is 1.

Speed ranges depend on the nature of traffic system in the city or country.

H. Multithreading problems

This algorithm involves multi access operations and updates at the same time on the same vehicle values. So, average calculation and insertion process is defined as atomic functions. Or, it is left to be handled by database management system locking mechanism if the system is implemented using database.

I. Measure Enhancements

In order to get more accurate results edges shouldn't contain any cross or sub turn with another street regardless of crossing street size.

J. Implementation

Implementation of the proposed algorithm is done using object oriented to encapsulate vehicle data and functions. A special function is defined to add records and increase time index every interval end to initiate next time interval and other weights calculation functions. As another implementation, database is used to utilize database management systems fast access algorithms.

To demonstrate the superiority of the proposed framework consider figure 1 as an example. Figure 1 shows two different routes between two points in Salt area in Jordan. Few statistics will be presented next. Those statistics are based on real observations. One of the routes is via a street with a possible average speed of 96 km/h (red colored in figure 1). This route is longer than another possible route by 6 kilometers with a total length of 11 kilometers (blue colored in figure 1). The average travel time for the longer route was 430 seconds. This is because that this route contains a single traffic light that stops traffic for a while. Walking time average was 413 seconds but average stop time was 17 seconds. Based on the above dynamically calculated statistics; this street (in the longer route) is considered a highway by our system compared to the other shorter route. The other street length is 5 kilometers and consists of 3 traffic lights in addition to a congested cross in the middle. The average speed was 41 km/h and average travel time was 673 seconds. Average walking time was 439 seconds and average stop time was 234 seconds. Dijkstra algorithm can be used to find out *best* path in terms of routes length. However, with our proposed framework, it is possible to consider the time as well as fuel cost factors. For instance, time factor produces the first route as an optimal one although second one is shorter in terms of length. Moreover, in case of using a Toyota Camry car, fuel cost of the second path will be 639 cc (cubic centimeters) of fuel while the cost of longer route is 792 cc of fuel. This example gives a flavor of how valuable our framework can be as it is customizable and dynamic.

Implementation is done using manual stop watches used by volunteer guys who take snap velocity reads and enter them to an excel sheets on their laptops. These excel sheets are migrated to the database of special implementation and applying average calculation process in order to get results.

Calculation time interval length is used to control the sensitivity of the algorithm to sudden small congestions. In case of very sensitive algorithm, time interval should be small interval so that sudden accidental situations are sensed. In case of low sensitivity to sudden small congestions, time interval is set to be long so that sudden congestions participation vehicles are treated as irregular reads in average calculations so that they are excluded.

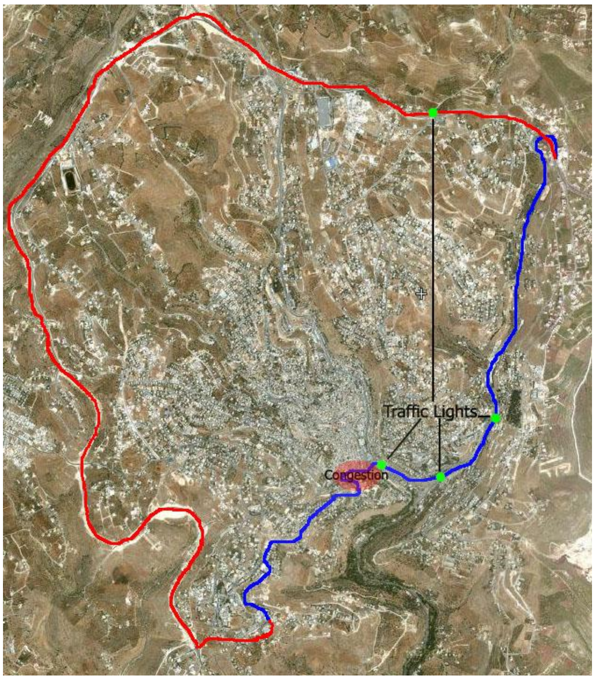

Figure 1. Implementation roads paths

IV. RESULTS AND CONCLUSIONS:

The main advantage of our proposal is that it is self learning and self maintaining. Consequently, the proposed framework is capable of providing its customizable best route identification based on multiple possible optimization factors such as travel time, fuel cost, and distance. Our framework also implies less overhead of administration (except the initial step of defining edges and vehicles information). Moreover, weights can be used by Traffic Agencies in order to achieve Intelligent Transportation System that control streets limited speeds and traffic lights to minimize congestions and derive efficient traffic flow. On the other hand, traffic agencies do *not* have to install any extra equipments or sensors in roads because that moving vehicles are used as moving agents that provide the server of travel agencies with the current statistical data regarding the status of the roads. The availability of such statistics travel agencies can find out obstacles or accidents especially in critical situations. The main disadvantage of the proposed solution is the complexity of insertion process and the high level of process synchronization needed in the algorithm. But this delay could be over came or neglected because that system will use last computed averages and weights till new ones are available so that system won't stop or wait. In order to achieve very high accuracy, GPS navigation equipment should be attached to significant number of the vehicles passing through streets. In fact, most of new model vehicles have embedded GPS navigation equipments already installed from their vendors. In future, new research is planed to minimize time needed to perform calculations and algorithm steps as a try to reach as much as possible to the optimality.

REFERENCES

[1] Dijkstra, E. W. (1959). "*A note on two problems in connexion with graphs*", *Numerische Mathematlk*, vol. 1, pp. 269–271, Nov. 1959.

[2] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Ed, *Prentice Hall*, 2009

[3] (2010) FHWA Arterial Management Website, latest information on traffic signal operations [Online]. Available: *http://ops.fhwa.dot.gov/arterial_mgmt/*

[4] Wikipedia (2011) Traffic light page on Wikipedia [Online]. Available: http://en.wikipedia.org/wiki/Traffic_light

[5] Wikipedia (2011) Congestion pricing page on Wikipedia [Online]. Available: http://en.wikipedia.org/wiki/Congestion_pricing

[6] Khalid A. S. Al-Khateeb, Jaiz A.Y. Johari and Wajdi F. Al-Khateeb, "Dynamic Traffic Light Sequence Algorithm Using RFID" Journal of Computer Science, vol. 4, pp. 517-524, 2008.

[7] P. Pongpaibool, P. Tangamchit,K. Noodwong, and Klong Luang NECTEC, "Evaluation of road traffic congestion using fuzzy techniques" In Proc. TENCON, 2007, paper 10.1109/TENCON.2007.4429119, p. 1

[8] Yan-hong Tang and Bao Xi, "Dynamic forecasting of traffic volume based on quantificational dynamics: A nearness perspective", Scientific Research and Essays, vol. 5, pp. 389-394, Feb. 2010.

[9] Tom Thomas, Wendy Weijermars, and Eric van Berkum , "Variations in urban traffic volumes", EJTIR, vol. 8(3), pp. 251-263, September 2008.

[10] Ronald Kroon, "Dynamic vehicle routing using ant based control", Master of Science thesis Delft University of Technology, Delft, Nederland, May 2002.

[11] Bogdan Tatomir, Adriana-Camelia Suson and Leon Rothkrantz, "Dynamic Routing and Travel Time Prediction with Ant Based Control," in Proc. 6th International Conference ANTS, 2008, paper 10.1007/978-3-540-87527-7, p. 404.

[12] O. Aloquili,A. Elbanna, and A. Al-Azizi , "Automatic Vehicle Location Tracking System Based on GIS Environment", IET Softw., vol. 3(4), pp. 255-263, 2009.

[13] Honda (2011) 2011 Honda Accord Sedan Specs. [Online]. Available: http://automobiles.honda.com/accord-sedan/specifications.aspx

[14] Mercedes-Benz (2011) 2011 Mercedes-Benz C300 Specs. [Online]. Available: http://www.mbusa.com/mercedes/vehicles/explore/specs/class-C/model-C300W

[15] Chevrolet (2010) 2010 Chevrolet COBALT COUPE Specs. [Online]. Available: http://www.chevrolet.com/cobalt/coupe/features-specs/

[16] (2010) STARTCOMS Systems website. [Online]. Available: http://www.starcomsystems.com/

[17] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, 3rd Edition,Cambridge, Massachusetts London, England: *The MIT Press*, 2009