A Comparative Study of Vehicles' Routing Algorithms for Route Planning in Smart Cities

Vi Tran Ngoc Nha, Soufiene Djahel and John Murphy
Lero, UCD School of Computer Science and Informatics, Ireland
{vi.tran-ngoc-nha, soufiene.djahel, j.murphy}@ucd.ie

***Abstract*—Vehicle routing problem (VRP) is a generic name referring to optimization problems in transportation, distribution and logistics industry. They mainly focus on serving a number of customers by a number of vehicles. Route planning techniques is one of the main tasks of VRP which aims to find an optimal route from a starting point to a destination on a road map. As road traffic conditions may change during the car journey (e.g., increase/decrease of the congestion level, road incidents etc), the optimal route should be re-evaluated as soon as an update in traffic conditions is available. Choosing an appropriate route planning algorithm among the existing algorithms in the literature to apply it in real road networks is an important task for any transportation application. In this paper, we first present a classification of the different route planning algorithms, and then explain how we compare and analyze their performance when they are applied in real road networks. For the purpose of comparison, we simulate the behavior of these algorithms during runtime using Simulation of Urban Mobility (SUMO) package and TRACI. We have chosen Dijkstra, the most well-known shortest path algorithm, to be the first algorithm to be implemented in SUMO. Upon reception of any traffic conditions update that affects the current optimal route of a car, we use TRACI to re-apply the algorithm and change this cars route accordingly. In the near future, our target is to simulate other algorithms and compare their performance based on the quality of the obtained best route.**

***Keywords* – Vehicles' Routing Problem (VRP), Vehicles' Routing Algorithms, Shortest Path, Route Planning, Dijkstra Algorithm.**

I. INTRODUCTION

Computing the shortest path between two locations in road networks is a challenging task in vehicles routing area and related transportation, distribution and logistics industry. Choosing a suitable route planning algorithm from the numerous algorithms proposed in the literature is a key issue in many transportation applications involving real road networks [1]. This is due to the fact that in such networks several dynamic parameters (e.g. traffic congestion level, random incidents, weather conditions etc) affect the efficiency of the applied algorithms. Therefore, those algorithms should be extended in order to take into account these dynamic parameters and update the chosen shortest path accordingly.

Car navigation systems are the most popular applications of Intelligent Transportation Systems (ITS). They provide information about traffic conditions, tourist locations as well as recommending the shortest routes to destinations based on Global Positioning System (GPS) and digital road map databases [2]. In order to realize the full benefits of car navigation system, the software components have to be able to operate in a real-time environment or a dynamic network. Dynamic road network is a network in which road conditions change overtime. Therefore, the algorithm applied for this network should react to changes in network by updating the previously chosen route so that it still guarantees optimal properties under new conditions [3]. The selection of such algorithm requires deep investigation of the appropriate evaluation criteria that we may use to assess its effectiveness in different road scenarios and under different road network scales.

In this paper, we present a literature review of the most significant algorithms for route planning in real-time environment and classify them into three categories according to the utilized approach for finding the shortest path. Afterwards, we propose a novel approach to enhance the effectiveness of these algorithms in dynamic road networks. This approach consists in defining new inputs for vehicles routing algorithms along with an interactive scheme to update the chosen route according to the changes during the vehicle journey. Besides, new evaluation metrics are also proposed in order to meet the needs of various categories of car drivers and owners (e.g. experienced drivers, new drivers, freight and transport companies, etc) which usually have different perceptions of the best route to be taken.

To compare the performance of the classified algorithms according to these metrics, we plan to choose one specific algorithm in each category to simulate. To this end, the open source microscopic road traffic simulator (SUMO) [4] is used to implement these algorithms. However, the current version of SUMO supports only simulating fixed routes which cannot be changed during simulation runtime. To change the route during simulation runtime, we propose to use TRACI (Traffic Control Interface), an interface to communicate between our program and SUMO. Currently, we are implementing Dijkstra for dynamic shortest path finding. The preliminary results show that it is possible to calculate the route via Dijkstra algorithm and the vehicles route can be updated if the set of the links constituting it are affected due to any event (e.g. car crash, stalled vehicle, congested road segment etc). Based on these results, we plan to extend our implementation on other algorithms to analyze their performance.

The remainder of the paper is organized as follows. Section II gives an overview of the existing vehicles routing approaches and classifies them in different categories. In section III, we describe the proposed inputs for improved vehicles routing algorithms and highlight the different evaluation met-

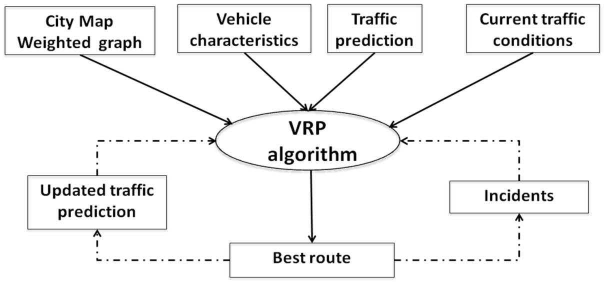

Figure 1: Vehicles' routing algorithms: main inputs and update

rics for both the best routes and the chosen routing approach. In section IV, we describe briefly how we have implemented Dijkstra algorithm using SUMO and TRACI to calculate and update vehicles routes during the vehicle journey. Finally, the conclusion and some directions of future work are given in section V.

II. Literature overview and classification

In this section, we explain briefly the key principle of the most significant vehicles routing approaches in the literature; then we classify them into three categories.

A. Vehicles routing approaches

There are a number of route planning algorithms applied in road networks. The main approaches proposed in the literature are briefly described below.

1) Dijkstra algorithm: Classical Dijkstra algorithm is a process of finding the path with the lowest cost (i.e. usually refers to the shortest path) from one node to all nodes in a city map. Its computation complexity is O(n2) with n being the number of nodes in network [6]. Dijkstra is one of the optimal algorithms based on labeling method. In addition, other labeling algorithms like Bellman-Ford-Moore, incremental Graph, threshold, topological ordering, etc. are also used to find shortest path. F. Benjamin [7] states that for finding the shortest path from one-to-one problem, it is worthwhile to consider Dijkstra algorithm since this algorithm is terminated as soon as the destination node is labeled, which also means that the shortest path is found. The other algorithms can only find optimal path when full shortest path tree is calculated meaning that shortest paths to all the nodes in the graph are found. Therefore, for searching the shortest path for one-to-all problem, incremental Graph is more efficient.

2) A algorithm:* A* is considered as a variant of Dijkstra algorithm but it uses a heuristic function rather than optimal search mechanism. Hence, A* restricts the search space and reduces the computational time. In traffic application, the search space is restricted to the area where traffic congestion has changed. Examples of A* algorithms extension are RTA* and a LRTA* [8] proposed for real-time applications. They usually use the direct distance between current location and the destination as a heuristic function.

3) Tabu search: Tabu search [9] has been applied to route planning problem by Liao et al in [10]. It is a local search-based meta-heuristic with run through several iterations. During each iteration, the best solution in the neighborhood of the current solution is chosen as the new current solution, even if the solution cost is increased. Hence, a bad local optimal solution is mitigated. A short-term memory, known as the Tabu-list, is required to store attributes of recently visited solutions. This helps to avoid short term cycling. The search stops after a fixed number of iterations or after a number of consecutive iterations without any improvement to the best known solution [11].

4) ANT based colony: This meta-heuristic is inspired from ant nature, when real ants communicate and cooperate with each other to find short paths from their nest to food sources [11]. When one ant finds a path from the colony to a food source, they lay down a chemical compound; known as pheromone on the ground and form a trail. If other ants find such a path, they are more likely to follow it instead of wandering randomly. This eventually leaves more pheromones and leads more ants to follow that path. The idea of ant colony algorithm is to simulate this behavior. When we apply this algorithm to VRP, the ants keep a memory about the visited nodes and the estimated time to reach them. Ant based control approach [12] has been used for searching the shortest paths in VRP as it is able to react to dynamic changes of traffic conditions.

5) Genetic Algorithms (GA): Genetic algorithms are used to solve routing search and optimization problems. GA is meta-heuristics inspired from a natural metaphor. It simulates the way species evolve and adapt to their environment, according to the Darwinian principle of natural selection. In the beginning, a randomly or heuristically population is generated. Then, this cycle is repeated for a number of generations. When applied to vehicle routing problems, the classical GA solution scheme is modified [11]. Since GAs always have routes in a population during a search, it is possible for the route to be re-evaluated in a short time using another route in the population and the constraints regarding all amenities in driving can be reflected in search [2].

6) Hybrid Genetic Algorithms: Kanoh et.al [13] proposed a hybrid approach which combines Genetic Algorithm with Dijkstra to solve a dynamic multi-objective problem. This algorithm finds the solution simultaneously for three objective functions: route length, travel time and ease of driving. In order to apply GA to traffic system, their approach uses Dijkstra to calculate the initial population of high-quality routes. From that initial population, this approach applies GA to generate later routes generations.

B. Classification of route planning algorithms

In this section, we classify the previous approaches into three main categories according to the technique used to explore the solutions space as shown in Figure 2. These three categories are: optimal algorithms based approaches, heuristic based approaches and hybrid approaches.

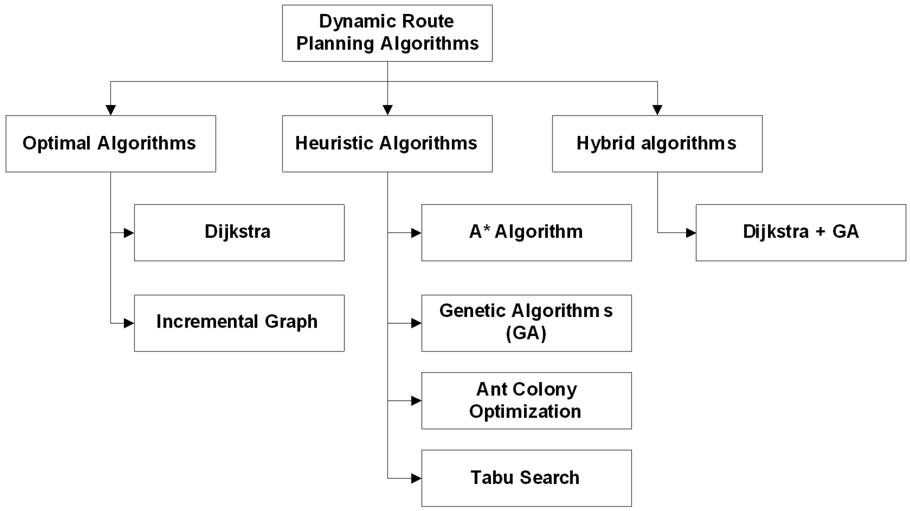

Figure 2: Classification of Dynamic Route Planning Algorithms

- Optimal algorithms guarantee to find the global optimal solution through the exploration of the whole set of available solutions.
- Heuristic based approaches explore a subset of the available solutions and usually find an approximate optimal solution that has qualities close to those of the global optimal one.
- Hybrid approach based algorithms leverage the strengths of both of the previous approaches.

The most common optimal algorithms are Dijkstra and Incremental Graph. They find the shortest path from one node to any other node in the road network. Heuristic based approaches include A*, Genetic Algorithm, Ant Colony Optimization and Tabu search. In order to reduce the computation time during search process, they accept the best route possible under certain constraints (time, search space, etc.) The last category combines both optimal and heuristic based approaches.

III. How Route planning algorithms can be improved?

In this section, we present our vision on how to improve the efficiency of the existing vehicles routing algorithms in road networks. To this end, we first discuss the main inputs which should be incorporated in these algorithms, and then present new metrics that might be used to calculate the best route for a given vehicle. Moreover, we discuss the main criteria used to assess the performance of these algorithms. Finally, we illustrate our route planning approach with real-time traffic updates.

A. Routing algorithms' inputs

In [5], the authors discussed the information that can be used as input for vehicles routing algorithms, which are:

- Road information: current traffic conditions like congestion level, incidents and weather conditions etc.
- Destination information: the purpose of travel
- Mobile information: such as, the remaining fuel, vehicle-related conditions, traveler-related conditions etc.

In addition to these inputs, we propose to incorporate the vehicles characteristics and road traffic prediction information since these two information will influence the best route selection.

We draw a high level overview of how to apply route planning algorithms in real-time system in Figure 1. This figure highlights the main inputs of vehicles routing algorithm, as well as the output which is the calculated best route according to one criterion or a combination of several criteria. The best route should be updated during the vehicles travel towards its destination in case of an incident or any update to the previously predicted traffic state. This is because such incidents may impact the chosen route and/or make it inaccessible for short or long periods of time. Notice that the vehicle characteristics are an important input of the algorithm since the height, weight and type of a vehicle (e.g. truck, car, bus) are important metrics to define the best route due to driving regulations and road infrastructure limited capacities (e.g. bridge, tunnel).

B. Best route selection criteria and algorithms evaluation metrics

In order to identify which route is more appropriate to the driver request, we should first define the metric of the set of metrics to be used for comparing the different available routes from the current location of the vehicle to the destination. Indeed, applying any of the aforementioned algorithms requires a metric or a combination of several metrics to be used by the objective function to measure and identify the optimal route. H. Kanoh et al [13] have applied different metrics for different objectives in their algorithm while Dijkstra only applies one metric (i.e. the travel distance) for calculating the shortest path. In what follows, we present the most significant metrics that vehicles routing algorithms may use as major pillar to search

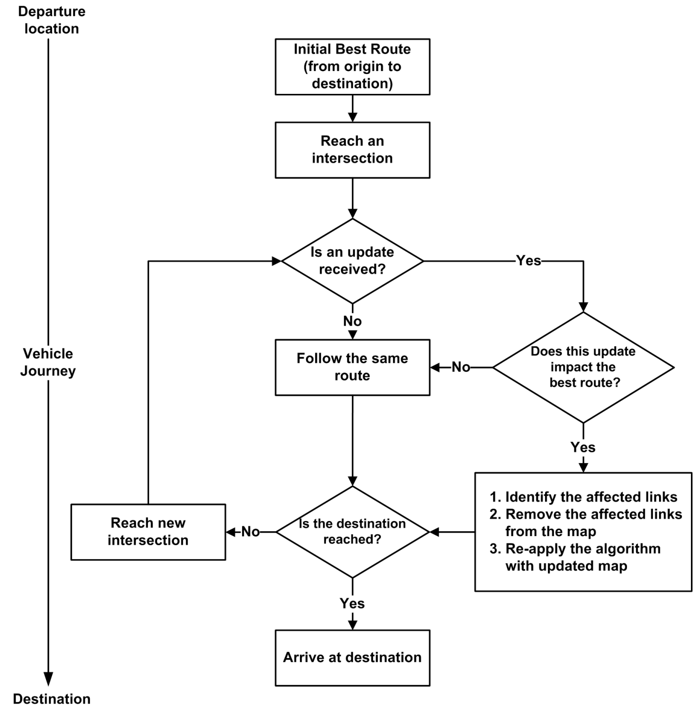

Figure 3: Flowchart illustrating the best route update during the vehicle's journey

the requested route.

- **Travel distance**: this is the basic criterion for shortest path finding. Each road on the map has its associated length value. Finding the shortest path means searching the route from the origin location to the destination through which the vehicle travels the shortest distance.
- **Travel time**: the travel time is another criterion for route planning algorithm. In this case, we consider the fastest route rather than the shortest one. The fastest route is the path through which the vehicle can reach its destination with within minimum travel time. The fastest path might be different from the shortest path due to the traffic constraints like traffic congestion and random incidents as well as the driving regulations like speed limit. Moreover, the fastest path should be updated regularly during the vehicle journey as traffic conditions change rapidly especially in big cities.
- **Easiness of driving**: easiness of driving is mentioned in Kanohs and Chakraborty's study [13], [14]. Car navigation device should provide user information about safer and more comfortable driving route [14]. This is also considered as driver preference reflected by factors such as number of turns or number of signal or width of the road. Based on this metric, route planning algorithm aims to find a vehicle route satisfying drivers preference the most.
- **Travel cost**: the travel cost refers to the number of toll tags in the chosen route as well as the estimated fuel consumption level during the journey. This value is not only affected by the route length but also by the number of stops during travelling, type of vehicle and the type of road. This metric is mainly useful for transport and freight companies as well as any other driver.
- **Combination of two or several metrics**: in this case, several metrics could be combined together to reflect the drivers preferences.

In general, the following three key evaluation metrics are used to measure the performance of vehicles routing algorithms; computation complexity, scalability and quality of the best route.

- **Computation complexity**: this metric is related to computational performance of a given algorithm. An algorithm which finds the optimal route with less travel time but longer computational time might not be the best algorithm, especially when the route is updated during the vehicles journey. Therefore, the computation complexity of each algorithm must be taken into account when comparing them.
- **Scalability**: another factor to take into account for vehicles routing algorithms evaluation purposes is the scalability. The scalability degree of an algorithm reflects the decrease of its performance when the size of the road network gets larger. Therefore, an efficient algorithm in small road network might not be applicable for large scale road networks.
- **Quality of the best route**: this metric is used to compare the different best routes calculated by different heuristics according to same metrics (i.e. travel distance, travel time etc) in order to determine which algorithm is calculating the closest solution to the optimal route.

C. Dynamic route planning framework

In Figure 3, we draw a flowchart describing how route planning algorithms should interact with the dynamic changing environment like road networks and adapt the best routes assigned to a vehicle according to the updates (i.e. change in congestions level, incidents etc) received from the traffic management systems. This framework consists of three main steps which are explained as follows.

- Step 1: calculate an initial best route from the origin location of the vehicle to its desired destination according to a chosen algorithm (e.g. Dijkstra, Genetic Algorithm etc.)
- Step 2: re-calculate the best route due to an update in traffic conditions. In this case, whenever a vehicle reaches an intersection, the traffic conditions are checked for any update. If there is an update impacting at least on link in the best route, the affected links are removed from the map and the route planning algorithm is re-applied to calculate a new best route for the vehicle. Otherwise, the vehicle carries on its journey.
- Step 3: is the destination location reached? If no, the step 2 is repeated until the vehicle reaches its last intersection and arrives at its desired destination.

IV. Performance evaluation

SUMO is chosen for simulating the studied algorithms in this paper since it offers an open source package, which is highly portable, applicable for microscopic road traffic to manage and monitor each vehicle in the network [15]. Moreover, SUMO supports Traci interface which provides the way to change vehicle route during runtime [16]. Firstly, we create a scenario with a specific road network. Figure 4 depicts the chosen road network topology to apply Dijkstra algorithm (see Algorithm 1). The number on each vertex represents its location. Based on the location information, we define nodes and edges properties in the configuration files named dijkstra.nod.xml and dijkstra.edg.xml. Secondly, we convert the map to SUMO format via netconvert application. The new map after conversion is shown in Figure 5.

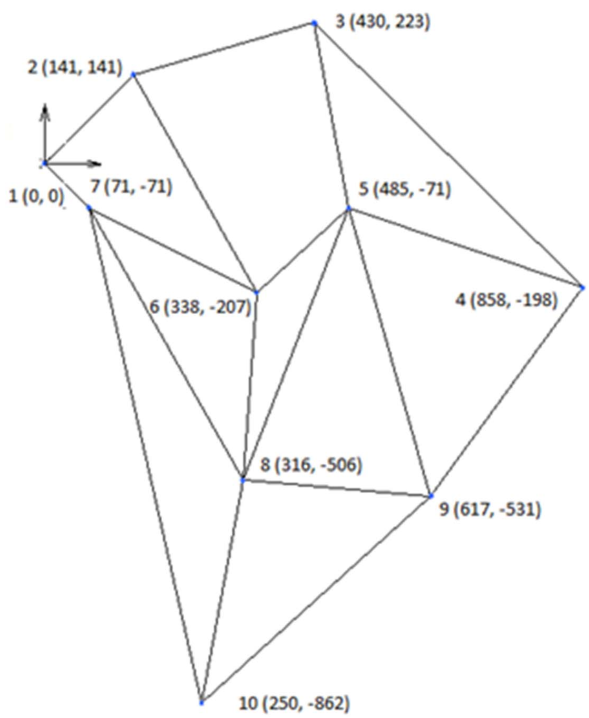

Figure 4: Network map to apply Dijkstra

In our simulation, we find an initial route for each vehicle from original node to the requested destination. The route is then assigned for this vehicle which will follow it as it is simulated by SUMO. When the vehicle reaches an intersection, the program checks the events variable, if any road segment (link) in the route is blocked because of traffic incident, Dijkstra algorithm is applied again to re-calculate the route from the current location of the vehicle to the destination or find an alternative link to replace (overcome) the blocked one. This process is repeated until the vehicle reaches its target. To do so, our program uses TRACI interface to inform SUMO to change the route which is already assigned to a vehicle. In this way, we can update a vehicles route during runtime.

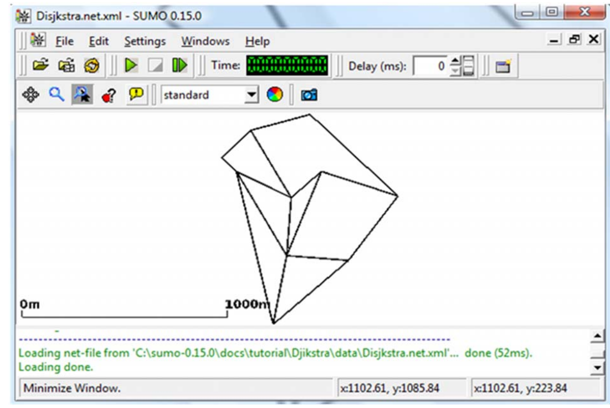

Figure 5: Network map after converting to SUMO network

Algorithm 1 Dijkstra algorithm main function in Python

```
def Dijkstra(graph,start,end=None):
final_distances = {}
predecessors = {}
est_distances = priorityDictionary()
est_distances[start] = 0
for vertex in est_distances:
final_distances[vertex] = est_distances[vertex]
if (vertex == end) then
    : break
end if
for edge in graph[vertex]:
path_distance = final_distances[vertex] + graph[vertex][edge]
if (edge in final_distances) then
    if (path_distance ¡ final_distances[edge]) then
        raise ValueError,
    else
        if ((edge not in est_distances) or (path_distance ¡ est_distances[edge])) then
            est_distances[edge] = path_distance
            predecessors[edge] = vertex
        end if
    end if
end if
return (final_distances,predecessors)
```

The program is written in Python. Below is a part of our code in which Dijkstra is a function to find shortest path from one node (start) to all nodes in the map; shortestPath is a function that calls Dijkstra function in order to calculate the shortest path from start to end. In main function (see Algorithm 2), after finding the route by calling shortestPath, this route is created in SUMO and is assigned to vehicle1 using TRACI API.

V. Conclusion

In this work, we have studied the most significant vehicles routing approaches in the literature and classified them into

Algorithm 2 Example of the main function of our implementation

```
def shortestPath(graph,start,end):
final_distances,predecessors = Dijkstra(graph,start,end)
path = []
while 1:
path.append(end)
if (end == start) then
    : break
end if
end = predecessors[end]
path.reverse()
return path
def main():
traci.init(8813)
route = shortestPath(graph,s,v)
#create a new route for vehicle
traci.route.add("1",route)
#assign the new route for vehicle with id vehicle1
traci.vehicle.add("vehicle1","1",-2,0,10.0)
```

three main categories according to the mechanisms used for searching the best routes. Moreover, we have introduced new inputs and metrics which might be used by those algorithms to improve their effectiveness in road networks. Furthermore, the criteria used to assess the performance of these algorithms are discussed. We also illustrated framework to update the best route during vehicle's journey. However, the implementation of the classified algorithms using SUMO and TRACI is still in progress. We are simulating the efficiency of Dijkstra algorithm in dynamic road networks in which the best route of a vehicle should be updated during its journey due to the variable congestion level and random traffic incidents. In this case, the best route is re-calculated and the vehicle follows the new route. In the future work, we plan to simulate other algorithms to measure their performance under specific road conditions and various road network scales. Then, we compare their performance according to the metrics discussed in this paper.

VI. Acknowledgement

This work was supported, in part, by Science Foundation Ireland grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Centre (www.lero.ie).

References

[1] F. Zhan, Three fastest shortest path algorithms on real road networks: Data structures and procedures, Journal of Geographic Information and Decision Analysis, vol. 1, no. 1, pp. 70-82, 1997.3

[2] H. Kanoh, Dynamic route planning for car navigation systems using virus genetic algorithms, International Journal of Knowledge-based and Intelligent Engineering Systems, vol. 11, pp. 65-78, Jan. 2007.

[3] B. C. Dean, Continuous-Time Dynamic Shortest Path Alorithms. PhD thesis, MIT, 1999.

[4] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, SUMO - Simulation of Urban MObility - an Overview, in SIMUL 2011, The Third International Conference on Advances in System Simulation, (Barcelona, Spain), pp. 55-60, 2011.

[5] G. Ghiani, F. Guerriero, G. Laporte, and R. Musmanno, Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies, European Journal of Operational Research, vol. 151, pp. 1-11, Nov. 2003.

[6] M. Fu, J. Li, and Z. Deng, A practical route planning algorithm for vehicle navigation system, in Fifth World Congress on Intelligent Control and Automation, vol. 6, pp. 5326-5329, IEEE, 2004.

[7] F. B. Zhan and C. E. Noon, Shortest Path Algorithms: An Evaluation Using Real Road Networks, Transportation Science, vol. 32, pp. 65-73, Feb. 1998.

[8] R. E. Korf, Real-time heuristic search, Artificial Intelligence, vol. 42, pp. 189-211, Mar. 1990.

[9] F. Glover and M. Laguna, Tabu Search, July 1997.

[10] T.-Y. Liao and T.-Y. Hu, An object-oriented evaluation framework for dynamic vehicle routing problems under real-time information, Expert Systems with Applications, vol. 38, pp. 12548-12558, Sept. 2011.

[11] M. Gendreau, J.-Y. Potvin, O. Braumlaysy, G. Hasle, and A. L kketangen, Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography, in The Vehicle Routing Problem: Latest Advances and New Challenges (B. Golden, S. Raghavan, and E. Wasil, eds.), vol. 43 of Operations Research/Computer Science Interfaces, pp. 143-169, Boston, MA: Springer US, 2008.

[12] B. Tatomir, L. J. M. Rothkrantz, and A. C. Suson, Travel time prediction for dynamic routing using ant based control, in Winter Simulation Conference, pp. 1069-1078, Dec. 2009.

[13] H. Kanoh and K. Hara, Hybrid genetic algorithm for dynamic multiobjective route planning with predicted trafic in a real-world road network, in Proceedings of the 10th annual conference on Genetic and evolutionary computation - GECCO 08, (New York, NY, USA), p. 657, ACM Press, July 2008.

[14] B. Chakraborty, T. Maeda, and G. Chakraborty, Multiobjective route selection for car navigation system using genetic algorithm, Proceedings of the 2005 IEEE Midnight-Summer Workshop on Soft Computing in Industrial Applications, 2005. SMCia/05., pp. 190195, 2005.

[15] G. Kotusevski and K. Hawick, A Review of Traffic Simulation Software, tech. rep., Institute of Information and Mathematical Sciences, Massey University, Albany, Aukland, New Zealand, 2009.

[16] A. Wegener, M. Piorkowski, M. Raya, H. Hellbr uck, S. Fischer, and J.-P. Hubaux, TraCI, in Proceedings of the 11th Communications and Networking Simulation Symposium on - CNS 08, (New York, NY, USA), p. 155, ACM Press, Apr. 2008.