

An Adaptive Vehicle Path Planning System

Chenn-Jung Huang, Kai-Wen Hu, Yu-Wu Wang, Heng-Ming Chen, Han-Wen Tasi, and Sheng-Yuan Chien

Abstract—Nowadays, most of the road navigation systems planning optimal routes are conducted by the On Board Unit (OBU). If drivers want to obtain the real-time road status, a Traffic Message Channel (TMC) module is needed. However, this module can only provide the current road conditions, instead of planning appropriate routes for users. In this work, the concept of the cellular automata is used to collect the real-time road conditions and derives the appropriate paths for users. Notably, fuzzy logic is adopted for path analysis for each cell established in the cellular automata algorithm. Besides establishing the optimal routes, this research is expected to automatically meet the personal demands of all drivers, reach the load balancing in all road sections to avoid the problem of traffic jams, and allow the drivers enjoy better driving experiences. A series of simulations were conducted to compare the proposed approach with the well-known A* Search algorithm and the latest state-of-the-art path planning algorithm in the literature. The experimental results exhibited that the proposed approach is scalable in terms of the turnaround time to individual users. The practicality and feasibility of applying the proposed approach in real-time environment is thus justified.

Index Terms—Search algorithm, cellular automata, fuzzy logic, optimal path planning.

I. INTRODUCTION

Researchers presented some algorithms to deal with path planning problems in the past. They can be classified to four kinds of shortest path problems, including Single-Pair Shortest-Path (SPSP), Single-Source Shortest-Paths (SSSP), All-Pairs Shortest-Paths (APSP), and Single-Destination Shortest-Paths (SDSP). There are solutions for above-mentioned four kinds of problems, like Bi-directed method [1] can solve SPSP problem; Dijkstra [2] proposed the shortest path algorithm to solve SSSP problem; Floyd [3] and Warshall [4] solve APSP and SDSP problems, respectively.

Ding *et al.* [5] presented a real-time infrastructure-free route guidance algorithm, called Vehicle-to-Vehicle Real-time Routing (V2R2). Although using V2R2 algorithm can reduce the cost of the establishment and maintenance in infrastructure, it is lack of scalability. Besides, load balancing cannot be achieved by this algorithm because it attempts to avoid the disconnection problem by ignoring the routes with no traffic load. Recently, Li *et al.* [6] combined the hierarchy traits of road network with the path planning algorithms of

Graph Voronoi Diagram [7] and HSR-Based Route Planning [8]. This hierarchical Voronoi graph based route planning algorithm [6] utilized the characteristics of the road network hierarchy to reduce searching time and the expanded node number of route searching. However, as mentioned in [9], if a car accident occurred in the route generated by the shortest path algorithm, this planned path cannot be used at all. Road network database was adopted in [9] to store all computed paths in the past. When a car accident occurs, the situation will be immediately updated in the road network database and case-based reasoning is used to locate the path in the database by examining if the origin and the destination are the same as the data stored in the database. However, this method still needs to take consideration of the road status at different time or different locations, or it cannot meet the requirement of path planning for every source/destination combinations.

Cellular automata are made up of cells like points in the lattice, and follow a simple rule [10]. They include large numbers of simple identical components with local interactions. Cellular automata can not only perform complex computation with high degree of efficiency and robustness, but also be addressed as massive collections of simple objects interacting locally with each other [11]. Each cell can assume a state from a finite set, changing its state simultaneously in discrete time steps according to a local transition rule. The new state of each cell depends on the previous state of itself, as well the state of its adjacent cells [12]. Cellular automata have been widely used to develop a wide variety of complex systems in the real world owing to its simple mathematical constructs and distinguishing features. To name a few, a method for spatial electric load forecasting using a reduced set of data was presented in [13]. The method used a cellular automata model for the spatiotemporal allocation of new loads in the service zone. The approach was tested in a real system from a mid-size city showing good performance. Cheng *et al.* [14] proposed a cellular automata-based approach for clustering the interests of car drivers, increasing the lifetime of interest groups, and increasing the throughput in vehicle-to-vehicle environments. Simulation results revealed the strengths of the proposed cellular automata clustering algorithm in terms of increased group lifetime and throughput for vehicular networks. Tonguz *et al.* [15] introduced a new cellular automata approach to construct an urban traffic mobility model. Based on the developed model, characteristics of global traffic patterns in urban areas were studied. Their experimental results showed that different control mechanisms used at intersections such as cycle duration, green split, and coordination of traffic lights have a significant effect on intervehicle spacing distribution and traffic dynamics.

To tackle the above-mentioned problems of traditional static path planning, this research proposed a novel dynamic

Manuscript received June 16, 2014; revised August 20, 2014. This work was financially supported in part by the National Science Council of Taiwan under contract numbers NSC 102-2511-S-259-010-MY3 and 99A058, respectively.

The authors are with the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan (e-mail: cjhuang@mail.ndhu.edu.tw).

optimal path planning method. The concept of the cellular automata is used to collect the real-time road conditions and derives the appropriate paths for users. A local server is established in each region and hierarchical servers are employed to plan routes. The real-time road information is first collected via servers. The information is analyzed and a suitable route is then generated for drivers. Meanwhile, current users' driving paths in the same region are also considered in decision making to prevent most vehicles from moving on the same route. Notably, fuzzy logic is adopted for path analysis for each cell established in the cellular automata algorithm. A series of experiments were performed in this study in order to evaluate the performance of the proposed work. The experimental results exhibit the feasibility and practicality of our proposed work.

The remainder of this paper is organized as follows. Section II presents the architecture of the proposed algorithm. The performance evaluations are given in Section III. Finally, conclusion is made in Section IV.

II. A DYNAMIC PATH PLANNING ALGORITHM

The road map is first divided into different sizes of regions according to the road density, traffic flow, and the area under the hierarchical framework as shown in Fig. 1. Each region is managed by a regional server, whereas an upper-level regional server at a higher hierarchical level is responsible for a region with a larger scope, which is composed of several nearby regions. When a driver requests a path planning, the upper-level regional server will determine which regional servers should be included in this planned route. We also use a path region analyzer module embedded with fuzzy logic to calculate the parameters of each road condition, including average speed, the current position, and the number of vehicles on the road. We work out the path planning problem by comparing an adjustable criterion for road conditions recorded on a traffic condition database to select some appropriate paths and mark them with certain suitability level. Thus, the optimal route can be derived by applying the concept of cellular automata recursively until the source and the destination are connected. Notably, the number of the hierarchical levels for the upper-level regional servers will vary with the length of the route. More higher-level hierarchical servers might be needed to handle the nearby macro-regions with higher traffic densities.

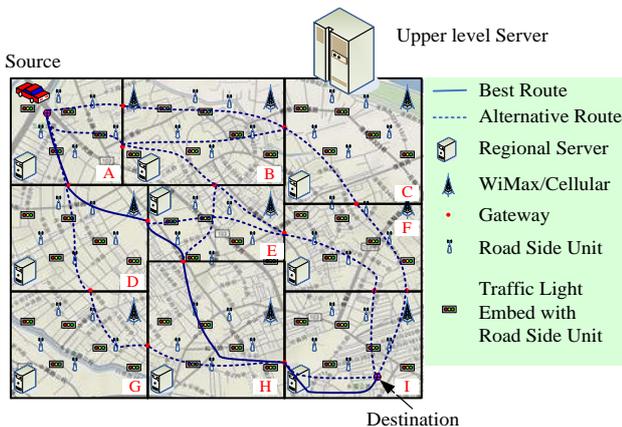


Fig. 1. Illustration of the servers and the areas they govern at different hierarchical levels.

Fig. 2 shows the architecture of the proposed work, which is constructed by the OBUs, regional servers, and upper-level regional servers. Drivers located at a specific region can issue path planning requests to their governing regional server through the OBU. If the destination of a path planning request is out of reach of the governing regional server, a route request will be sent to an upper-level regional server at higher hierarchical level for further processing. Accordingly, each regional or upper-level regional server accepts the requests from not only the users in the area that it governs, but also the upper-level regional servers at upper levels if a route request within the local area that the regional or upper-level regional server governs arrives.

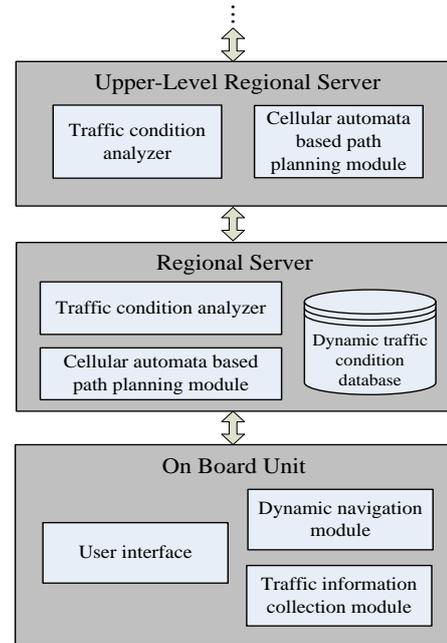


Fig. 2. System architecture of the proposed approach.

The traffic information collection module located at the OBU is responsible for informing its governing regional server of current location and velocity of vehicles, and a dynamic navigation module regularly checks the traffic status and reminds users to change the path based on the update of the traffic status. Each regional server periodically evaluates the traffic status, including the number of the vehicles and the average velocity of the vehicles on reach road, and updates the latest status of all road segments in the real-time road status database, which is established at each region. The regional path planning algorithm is used to draw up the best route based on the local traffic information that the regional server governs while receiving route request within the local region that the server governs.

The flow diagram of our proposed dynamic vehicle path planning algorithm is shown in Fig. 3. Our approach manages several hierarchical servers to make a large-scale path planning. The path region analyzer at the server side is responsible for analyzing the route requested by users, and to determine if the sever that it resides is placed at the highest hierarchical level. If the path region analyzer is at the highest level, it will collect the path planning results received from the regional servers it governs and determine the final path for the vehicle. Otherwise, it will forward the collected data to its upper level server, called an upper-level regional server,

to ask for route planning. All upper-level regional path planning servers receive the path returned from its governing regional or upper-level regional servers and also make the optimal path planning. If it is the highest level server, it will provide the result of path planning to the requesting OBU. Otherwise, it will turn to a higher level server for further

processing. Notably, each regional or upper-level regional server uses the concept of cellular automata to draw up the optimal path planning of its own. Throughout the division and cooperation of each regional or upper-level regional server, a nested path planning of cellular automata is accomplished.

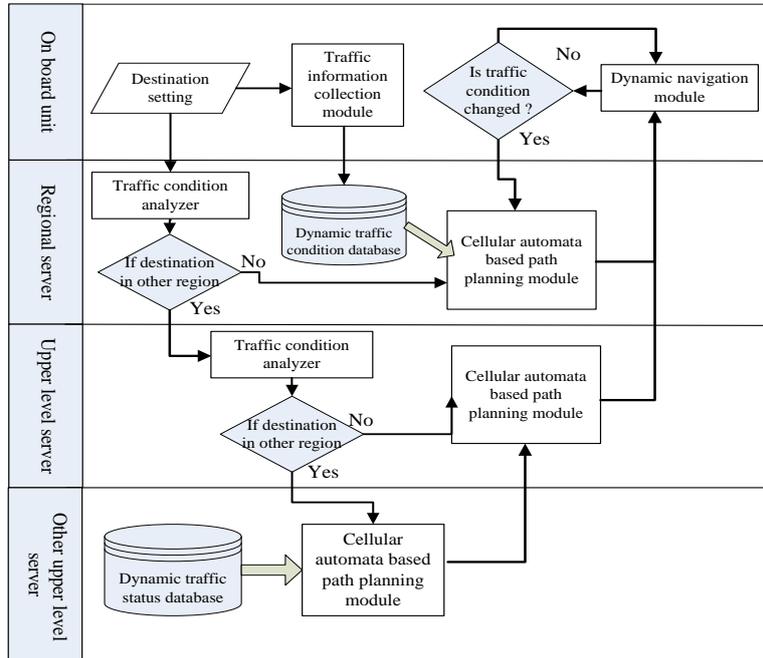


Fig. 3. Flow diagram of the proposed dynamic path planning scheme.

During the real-time navigation, the OBU will monitor the road status of the next section. When the road status of the planned path becomes worse or that of some other path turns out to be better, it will ask the regional server for re-planning each local path for the following road sections. If the new path is superior to the current one, the OBU will remind the driver to substitute the original path.

The database of the real-time path planning in the regional server is used to do the dynamic optimal path planning. At the beginning of the path planning, we have already considered the actual road status along the planned path, and avoided the jammed sections. In addition, our scheme also makes responses of changes timely according to actual road status and suggests the drivers to change the current route in case of substitution of another better path than the original one. Comparing with the traditional static path planning, our method can significantly assist drivers in making considerate path planning.

Notably, when users make a wide range of path planning, the traditional algorithm based on the graph theory needs to compute for every separate section on the route. Thus, the computational complexity will increase dramatically if the coverage area of the routed path is larger. The proposed approach instead makes use of the hierarchical structure and the concept of the cellular automata, so that the computational complexity can be reduced when comparing with the traditional method. In the case of multiple users, the traditional method needs to make different calculations for each user, but the hierarchical structure in the proposed approach can keep the former computation results for the usage of subsequent users, and thus significantly lowers the computation complexity.

A. Operations of an OBU

In the proposed approach, the OBU includes the traffic information collection module, the dynamic navigation module, and the user interface. The traffic information collection module mainly records vehicular position and velocity, and uploads the collected data to the regional server. During the navigation, the dynamic navigation module will regularly look up the current traffic status from the regional server and remind users to change the route according to the latest planned path. While starting using the proposed system, an OBU will regularly request the traffic information from the regional server it resides and communicate with other vehicles. The OBU needs to transfer the information into the regional server it resides, and saves the current traffic information into real-time path database. Additionally, the user interface shows the result with dynamic navigation and path planning.

B. Operations of a Regional/Upper Level Server

A regional server or upper level servers consist of a dynamic traffic information database and a cellular automata based regional path planning module. The dynamic traffic information database stores the current traffic information and provides it to the cellular automata based regional path planning module that finds the optimal path.

1) Real-time traffic data collection and calculation

Regional servers receive the parameters from the OBUs, including average speed, the current position, and the number of vehicles on the roads. As shown in Table I, these data are stored in a real-time traffic information database to evaluate which traffic condition level these sections should be

assigned to. There are different speed limitations for different categories of the pathways. Thus, even two cars with the same speed may be categorized into different traffic conditions depending on the type of pathway that the vehicle resides at. The system will update the latest traffic condition of every road section and establish new dynamic map information in the form of records of road traffic condition description as shown in Table II.

Take Fig. 4 as an example, we can see different traffic levels in the same region are marked with different colors, and the driver can easily query all traffic information immediately.

TABLE I: TRAFFIC CONDITION MAPPING TABLE

Average speed (km/hour)	Category of pathway	Traffic condition
<20	All types of pathway	Poor
20-40	Urban area	Acceptable
20-40	Suburban area	Poor
40-60	Highway	Poor
>60	Highway	Good

TABLE II: RECORD OF ROAD TRAFFIC CONDITION DESCRIPTION

Section ID	Entry node	Number of vehicles	Average speed	Traffic condition level	Exit node
------------	------------	--------------------	---------------	-------------------------	-----------

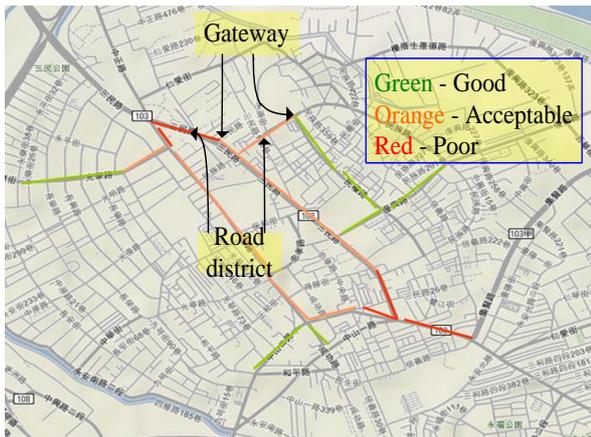


Fig. 4. An illustration of dynamic traffic status update.

2) Cellular automata-based path planning

The concept of the cellular automata [16] is adopted in this work to give the synchronized reports of road conditions and by using regional servers distributed all over the areas, and then determine the best route based on a hierarchical framework. Each road segment forms a basic cell, and some adjacent connected cells can also be seen as an upper-level cell. For example, each road segment under the regional server is a cell, and the selected road segments in the same region can form a long connected path that as well can be seen as a cell in the view of an upper-level server. The operations of a cell are shown in Fig. 5. There are four kinds of statuses for a cell, including “not selected”, “selected”, “selected & connected with the source”, and “selected & connected with the destination”. All of the cells are initially marked as “not selected”. Our algorithm checks whether the traffic condition of a specific cell is better than some preset criterion. The cell is marked as “selected” if the traffic condition is satisfactory. When a “selected” cell is connected to the source, its status will be “selected & connected with the source”. In the same way, the cell will be “selected & connected with the destination” if this cell is connected to the destination. Each “selected” cell will check if the status of its neighboring cell is marked “selected & connected with the source/destination”. If its neighboring cell is connected to the source or the destination, its status will be changed the same as its neighboring cell. The stopping criteria are that a cell has both of the status of “selected & connected with the source” and “selected & connected with the destination”. Consequently, the planned path is finished, and the regional server will provide the result to the upper-level regional server, and then the driver will receive the best planned path from OBU. Notably, the traffic condition is repeatedly forwarded to the upper-level regional servers at different hierarchical level if necessary. Meanwhile, the threshold that determines if the road segment should be marked as “selected” can be dynamically adjusted if the path between the source and the destination cannot be established owing to poor traffic conditions.

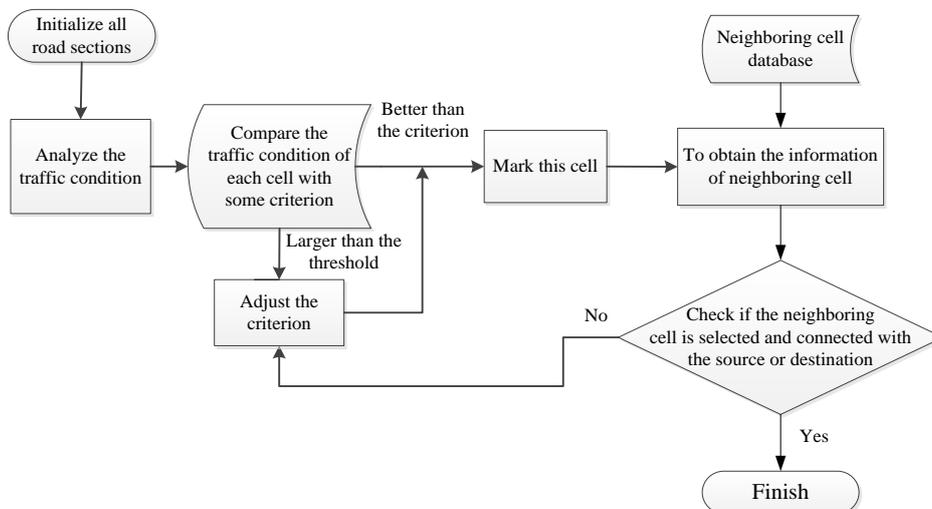


Fig. 5. Operations of each cell in cellular automata algorithm.

3) Fuzzy Logic based Traffic Condition Analyzer

The idea of fuzzy logic was first advanced by Dr. Lotfi Zadeh of the University of California at Berkeley in the

1960s. Fuzzy logic is an approach to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based. Dr.

Zadeh was working on the problem of computer understanding of natural language.

Fuzzy logic was conceived as a better method for sorting and handling data but has proven to be an excellent choice for many control system applications since it mimics human control logic. It can be built into anything from small, hand-held products to large computerized process control systems. It uses an imprecise but very descriptive language to deal with input data more like a human operator. It is very robust and forgiving of operator and data input and often works when first implemented with little or no tuning.

The fuzzy traffic load estimator proposed in this work is composed of the following components as described.

- **Fuzzifier:** The fuzzifier performs the fuzzification function that converts three types of input data from the traffic load estimated scheme into suitable linguistic values which are needed in the inference engine.
- **Fuzzy rule base:** The fuzzy rule base is composed of a set of **linguistic** control rules and the attendant control goals.
- **Inference Engine:** The inference engine simulates

Rule r

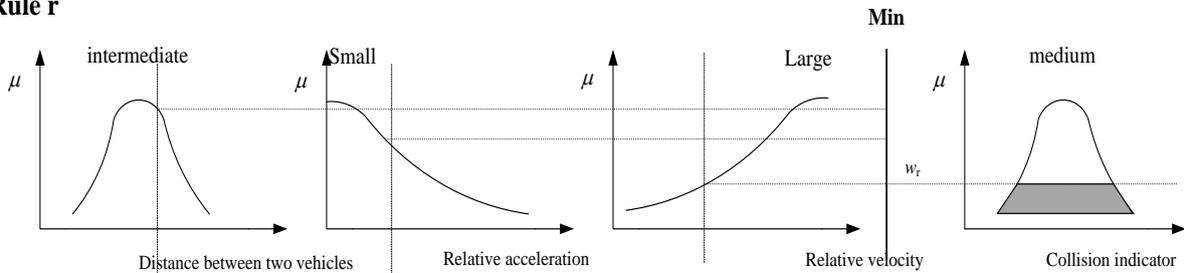


Fig. 6. The reasoning procedure for Madamni defuzzification method.

Fig. 6 illustrates an example of the reasoning procedure. The rule in Fig. 6 is:

IF the distance between two vehicles is “intermediate”, **AND** the relative acceleration between two vehicles is “small”, **AND** the relative acceleration between two vehicles is “large”, **THEN** the collision indicator is “medium”.

The non-fuzzy output of the defuzzifier can then be expressed as the weighted average of each rule’s output after the Mamdani defuzzification method is applied:

$$load_{BW} = \frac{\int_l \mu_A(l)dl}{\int_l \mu_A(l)dl} \quad (1)$$

where $\mu_A(l)$ denotes the aggregated output constructed by the connective “AND” as shown in the above example is satisfied.

III. PERFORMANCE EVALUATION

To evaluate the effectiveness of the proposed algorithm, a series of simulations are conducted to compare the proposed work and the well-known A* Search algorithm [17]. A* Search algorithm was selected for comparison because it is the most effective one, where the optimal path obtained by computing the minimum path length from every node to destination is determined. 10000 road sections were included in our simulation environment. Each path planning was

human decision-making based on the fuzzy control rules and the related input linguistic parameters.

- **Defuzzifier:** The defuzzifier acquires the aggregated linguistic values from the inferred fuzzy control action and generates a non-fuzzy control output, which represents the estimated traffic load of the cell.

Notably, the inputs to the fuzzifier are distance between two vehicles, relative acceleration, and relative velocity, respectively. The membership functions for the distance between two vehicles adopts the linguistic variables “low”, “intermediate” and “high”; The membership function for the relative acceleration between two vehicles is mapped into three linguistic term sets, “small”, “medium” and “large”; and the membership function for the relative velocity is mapped into three linguistic term sets, “small”, “medium” and “large”, respectively. The output parameter of the inference engine is defined as the collision indicator. The fuzzy linguistic variables for the output are “low”, “medium” and “high”, which are represented the measure of the network traffic load.

scheduled from the upper left corner to the lower right corner as shown in Fig. 1.

The comparison of total response time for different numbers of users is shown in Fig. 7. The response time that the traditional A* Search algorithm spent increased rapidly with the increasing of the number of users because each planned path has to be computed in turn. On the contrary, the computation tasks are distributed through the regional servers and upper-level regional servers in the proposed approach. This is why our proposed approach has less total response time.

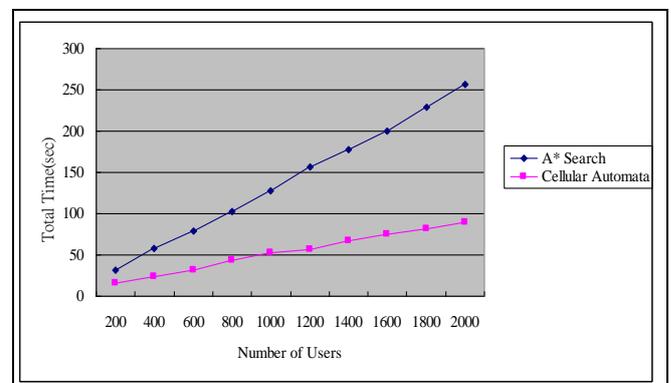


Fig. 7. Comparison of total response time for different number of users.

Fig. 8 shows the influence of the path length between cellular automata based path planning algorithm and A* Search algorithms. As shown in Fig. 8, the total response time

for wide range of road sections for A* Search algorithm would increase with the increase of the path length. However, compared to A* Search algorithm, the proposed cellular automata algorithm cost less time. Therefore, long distance path planning can have better performance in time savings while adopting cellular automata based path planning algorithm.

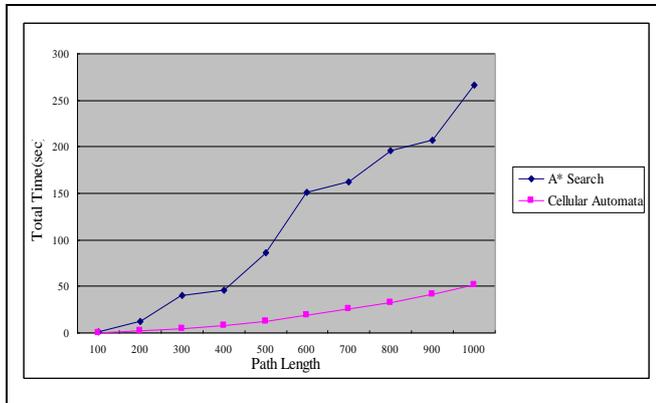


Fig. 8. Comparison of total response time with the increasing of path length.

In the next simulations, instead of using traffic condition analyzer all the time, we retrieved the up-to-date traffic conditions from the dynamic traffic condition database if they had been freshly updated. The total response time for different numbers of users is illustrated in Fig. 9. The A* Search algorithm regards each user as a dependent path planning task, so one's planned result cannot be used for others. Consequently, the total response time would increase with the increasing number of users. On the other hand, the cellular automata approach evaluates the road condition for the whole region at the initial stage, so it only needs to check whether the starting point and the destination have been connected or not while planning a new route. Even the number of users might increase rapidly, the cellular automata approach does not have to repeatedly calculate the road condition, and the planned results can be provided to the following uses as well. Accordingly, our proposed scheme achieve cost less time than A* Search algorithm regardless of the increasing or decreasing number of users.

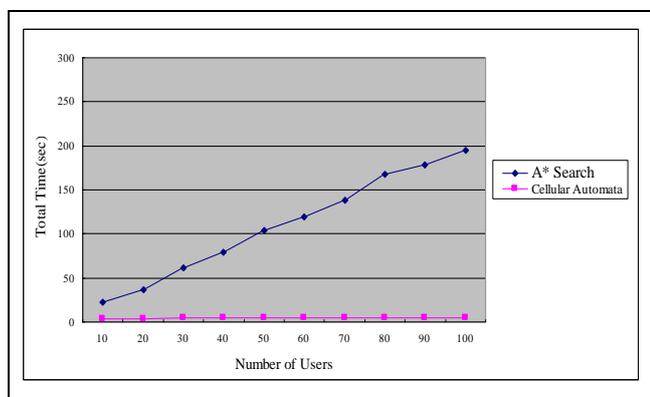


Fig. 9. Comparison of total response time while retrieving the up-to-date traffic conditions from the database.

IV. CONCLUSIONS

In this work, cellular automata based path planning algorithm was proposed to improve the path planning issue.

According to our experimental findings, the proposed approach which deploys regional and macro regional servers distributedly can plan optimal routes in an effective way. With the help of cellular automata algorithm, our scheme can effectively lower the computation complexity, and accelerate turnaround time to fulfill the requirements of real-time path planning. The experimental results show the effectiveness and practicability of our proposed approach, even in the conditions of the increasing of users, a wide range of road sections, and long path length planning. Although the proposed approach needs extra hardware cost for the deployment of regional servers, this expenditure is well worthy because real-time turnaround time for a path planning request is crucial.

REFERENCES

- [1] I. Phol, "Bi-directional search," *Machine Intelligence*, vol. 6, pp. 127-170, 1971.
- [2] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 2, pp. 269-271, 1959.
- [3] R. Floyd, "Algorithm 97: Shortest path," *Communications of the ACM*, vol. 5, no. 6, 1962.
- [4] S. Warshall, "A theorem on boolean matrices," *Journal of the ACM*, vol. 9, no. 1, 1962.
- [5] J. W. Ding, C. F. Wang, F. H. Meng, and T. Y. Wu, "Real-time vehicle route guidance using vehicle-to-vehicle communication," *IET Communications*, vol. 4, no. 7, pp. 870-883, Apr. 2010.
- [6] Q. Li, Z. Zeng, and B. Yang, "Hierarchical model of road network for route planning in vehicle navigation systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 1, no. 2, pp. 20-24, 2009.
- [7] E. Martin, "The graph Voronoi diagram with applications," *Networks*, vol. 36, pp. 156-163, 2000.
- [8] A. Car and A. U. Frank, "General principles of hierarchical spatial reasoning - The case of way finding," in *Proc. the SDH'94*, 1994.
- [9] H. Kanoh and T. Nakamura, "Knowledge based genetic algorithm for dynamic route selection," in *Proc. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, 2000, vol. 2, pp. 616-619.
- [10] E. Fredkin, "Digital machine: A informational process based on reversible cellular automata," *Physica. D*, vol. 45, no. 1-3, pp. 254-270, Sep. 1990.
- [11] N. H. Packard and S. Wolfram, "Two-dimensional cellular automata," *J. Stat. Phys.*, vol. 38, no. 5/6, pp. 901-946, Mar. 1985.
- [12] J. Kari, "Reversibility of 2D cellular automata is undecidable," *Physica D*, vol. 45, no. 1-3, pp. 379-385, Sep. 1990.
- [13] E. M. Carreno, R. M. Rocha, and A. Padilha-Feltrin, "A cellular automaton approach to spatial electric load forecasting," *IEEE Transactions on Power Systems*, vol. 26, no. 2, pp. 532 - 540, 2011.
- [14] S. T. Cheng, G.-J. Horng, and C. L. Chou, "Using cellular automata to form car society in vehicular Ad Hoc networks," *IEEE Trans. on Intelligent Transpor. Systems*, vol. 12, no. 4, pp. 1374-1384, 2011.
- [15] O. K. Tonguz, W. Viriyasitavat, and B. Fan, "Modeling urban traffic: A cellular automata approach," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 142-150, 2009.
- [16] H. Beigy and M. R. Meybodi, "Cellular learning automata with multiple learning automata in each cell and its applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 1, pp. 54-65, Feb. 2010.
- [17] E. P. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. on Systems Sci. and Cybernetics*, vol. 4, no. 2, pp. 100-107, Jul. 1968.



Chenn-Jung Huang received the B.S. degree in electrical engineering from National Taiwan University, Taiwan and the M.S. degree in computer science from University of Southern California, Los Angeles, in 1984 and 1987, respectively. He received the Ph.D. degree in electrical engineering from National Sun Yat-Sen University, Taiwan in 2000. He is currently a professor in the Department of Computer Science & Information Engineering, National Dong Hwa University, Taiwan. His research interests include computer communication networks, data mining, and diagnosis agent for e-learning.



Kai-Wen Hu is pursuing his doctoral degree at the Department of Electrical Engineering, National Dong Hwa University, Taiwan. His research interests include computer communication networks, data mining, applications of machine learning techniques and e-learning.



Han Wen Tsai is pursuing his master's degree at the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan. His research interests include computer communication networks, data mining and applications of machine learning techniques.



Yu-Wu Wang is pursuing his doctoral degree at the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan. His research interests include computer communication networks, data mining and applications of machine learning techniques.



Sheng-Yuan Chien is pursuing his master's degree at the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan. His research interests include computer communication networks, data mining and applications of machine learning techniques.



Heng-Ming Chen is pursuing his doctoral degree at the Department of Electrical Engineering, National Dong Hwa University, Taiwan. His research interests include computer communication networks, data mining, applications of machine learning techniques and e-learning.

Architecture and Software Design

