

# Genetic Algorithm-based Dynamic Vehicle Route Search using Car-to-Car Communication

Byonghwa OH, Yongchan NA, Jihoon YANG, Sungyong PARK, Jongho NANG, Jihwan KIM

Department of Computer Science and Engineering  
1 Shinsoo-Dong, Mapo-Ku, Seoul 121-742, Korea  
{mrfive, ycna, yangjh, parksy, jhnang, kimjihwan}@sogang.ac.kr

**Abstract**—Suggesting more efficient driving routes generate benefits not only for individuals by saving commute time, but also for society as a whole by reducing accident rates and social costs by lessening traffic congestion. In this paper, we suggest a new route search algorithm based on a genetic algorithm which is more easily installable into mutually communicating car navigation systems, and validate its usefulness through experiments reflecting real-world situations. The proposed algorithm is capable of searching alternative routes dynamically in unexpected events of system malfunctioning or traffic slow-downs due to accidents. Experimental results demonstrate that our algorithm searches the best route more efficiently and evolves with universal adaptability.

**Index Terms**—Adaptive System, Genetic Algorithm, Traffic Information System

## I. INTRODUCTION

The demands for car navigation systems, which assist individual drivers by calculating the most efficient driving route and providing the route information between two geographical points based on a graphical user interface, have drastically increased in recent years. However, the existing systems have also faced challenges in providing information that reflects the fluctuations of dynamic real-time traffic and road situations, since they function merely on static data such as the distance between two points on a map.

In an attempt to deal with such a limitation, new technology was recently introduced which enables each navigation terminal to receive and incorporate real-time traffic information collected by a centralized data center, such as TPEG (Transport Protocol Experts Group) [1] information embedded in DMB (Digital Multimedia Broadcasting) [2] signals, and to use it to work out the optimal route.

Nonetheless, the technology may be very costly to run in terms of the amount of time spent on collecting and broadcasting traffic information, and the geographic scope of the information is also significantly restricted. Moreover, it is possible to cause another type of traffic congestion because it broadcasts identical routing information to all users, leading them to the so-called "optimal route".

To overcome these limitations, it is necessary to design a new form of car navigation system which can reflect the dynamic traffic data and calculate and provide the optimal route and other information on a real-time basis. With this consideration, we have designed a new approach which can be used to collect the data of one's own routes (e.g. distance,

GPS data, average velocity, etc.), communicate such data between neighboring terminals, and provide the optimal routes based on the collected data. While each car receives minimal static information (e.g. initial routes) from the data center, it can obtain additional real-time data from its neighbors to compute the optimal routes dynamically. Figure 1 depicts the infrastructure of such a navigation system by car-to-car communication.

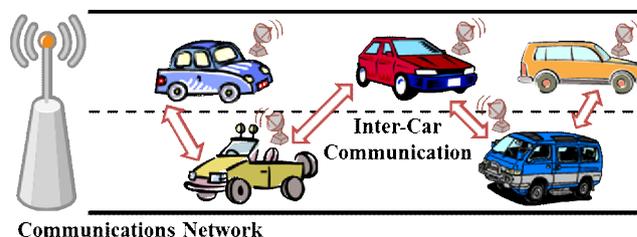


Figure 1. Navigation system based on inter-car communication

In this paper, we propose a dynamic vehicle route search algorithm based on a bio-inspired optimization method, the Genetic Algorithm, under the infrastructure shown in Figure 1. Each car continuously communicates traffic data on the roads just passed by its driver with neighboring navigation terminals to update its traffic database. Then it rectifies the static route determined a priori by applying the Genetic Algorithm with the updated information. The feasibility of our proposed approach is verified by experiments with real traffic data and visualization tools displaying the results.

The rest of the paper is organized as follows: Section 2 introduces related work on route search based on the Genetic Algorithm and its application to our navigation system. Section 3 describes our dynamic vehicle route search algorithm, Alternative Route Search. Section 4 explains the data, experimental environment and setup, and the results. Section 5 concludes with a summary and discussion of some directions for future research.

## II. ROUTE SEARCH BASED ON GENETIC ALGORITHM

The Genetic Algorithm (GA) [3] is an adaptive heuristic search method premised on the evolutionary ideas of living organisms, and is one of the approaches for probabilistic search, learning and optimization. While maintaining the population that describes candidate solutions, the entities (or individuals) in the population evolves using genetic operators of selection, crossover, and mutation. The size of the population is maintained, however superior individuals

This work was supported by the Special Research Grant of Sogang University to Jihoon Yang, the corresponding author.

Digital Object Identifier 10.4316/AECE.2010.04013

are preserved and the inferior are replaced by new individuals reproduced during evolution. The algorithm has been shown to draw good solutions for a variety of problems.

In this paper, we design a route search algorithm based on the approach in [4] where GA was used to find the shortest path. It defines and uses variable-length chromosomes for encoding the problem, in which each gene represents a place in the map. The crossover operation exchanges partial chromosomes at independently chosen two crossing sites, and the mutation operation makes changes to randomly determined sub-paths between the source and the destination. Such genetic operations may cause invalid chromosomes in terms of the representational requirements for the given task of route search (e.g. there does not exist real connecting links among the places in a candidate solution). However the algorithm fixes invalid chromosomes using the repair function (More detailed information of this function is in [4]).

Our dynamic vehicle route search algorithm, named as Alternative Route Search (ARS), is based on GA as described above. However, ARS is different from the previous work in that it considers dynamically determined costs between places based on the current road conditions. In other words, ARS applies GA to dynamically changing traffic situations and produces the best routes continuously rather than a fixed, static route as in [4]. Figure 2 displays an example of finding dynamic routes by ARS (Detailed descriptions on ARS are given in Section 3).

### III. ARS ALGORITHM

#### A. Overview

In a car navigation system, dynamically adjusting the fixed route is intuitively appealing. Given a predetermined route from the source to the destination, users of the

navigation system are constantly interested in finding better partial routes (from where they are currently to the final destination) that minimizes the remaining travel time, rather than sticking to the initial fixed route. This is because, even though the fixed routes might be the best choice in terms of the distance, those do not guarantee the shortest travel time. Our ARS algorithm is proposed for this purpose of dynamic adjustment of routes.

The predetermined route might be chosen either by discretion or a fast algorithm (e.g. A\* algorithm [5]). Also it might be chosen based on a predetermined and memorized optimal route between the source and the destination. All-source all-destination routes can be acquired either by running the entire route search algorithm offline or simply from the navigation system manufacturers. In this research, we assume that such all-source all-destination routes are pre-computed based on the average commute time between each point on the map and stored in the navigation system.

For evaluating an entire route, we use GA to reflect the dynamic real-time traffic data in order to alter a certain part of it. As a result, a partial congested route is replaced by an alternative partial route. Figure 2 depicts this ARS method, showing an example of replacing the partial route of the original path (brown line) with newly generated partial route (yellow-red dotted line), by considering the partial search range (blue circled area) from the rerouting point. In our experiments, we set the partial search range as a circle with a radius of 5 (in terms of connected points) centered at the rerouting point in order to define a reasonably small and manageable region.

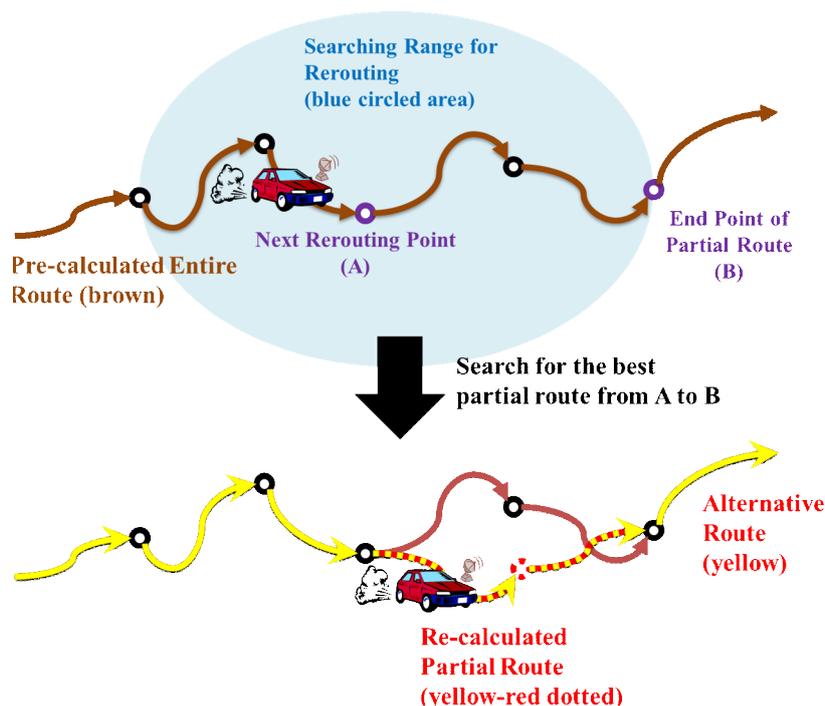


Figure 2. Dynamic route generation by ARS

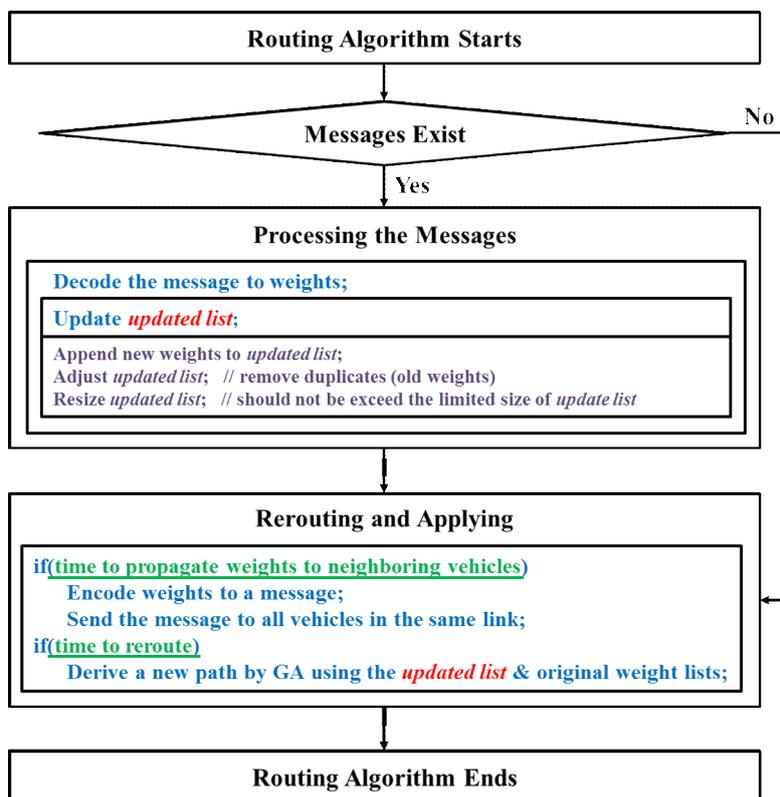


Figure 3. The process of ARS algorithm

### B. Algorithm Description

Regarding the entire route as a graph that consists of individual edges among the vertices of places, the weights of an edge can be defined as the time consumed to run the certain sector, which simplify the entire route search problem to drawing the minimal time consumed running the source and the destination. According to the instruction from Korea Ministry of Land, Transport and Maritime Affairs, we now define each edge as *link* and each vertex as a *node*.

The navigation system on each vehicle runs the algorithm at multiple points of time. Each vehicle has two weight lists: One is a list of (link, weight) pairs where the weight corresponds to the average observed travel time of the link (*original list*); and the other is such a list for the links with newly observed weights (*updated list*). Since *updated list* needs to be up to date, the vehicle regularly deletes old elements (not up to date or collected before a certain period of time) if any.

The algorithm operates as follows. First, it checks if there is any new message about the commuting time (weight information) of links on one navigation system, transmitted from other vehicles' systems. If there is any, it constructs *updated list* ordered by timestamps, eliminating all the old elements as described above. Considering the limited capacity of the memory (say  $n$  elements) in the navigation system, *updated list* is trimmed to  $n$  most recent elements.

Then, the car navigation system conducts rerouting at a regular interval. The navigation system collects data on traffic speed or commuting time from other systems, and transmits its data to them every time it passes by the other systems. The data are propagated to all the systems in the cars on the same link. Rerouting is performed at a certain time interval to prevent the deterioration of the service

quality due to the excessive computing time required during a single routing process. The time interval is set a little longer than the actual calculation time.

Figure 3 shows the process of ARS in pseudo-code. In Figure 3, the underlined parts are contingent on the variable parameters such as the size of *updated list*, propagation interval of the weights, rerouting interval, and the partial search range.

## IV. EXPERIMENTS

### A. Overview

The actual road data used in this research consist of nodes and links as described by Korea Ministry of Land, Transport and Maritime Affairs. The data were collected on the links in the city of Seoul, every 10 minutes from the 11th of September through 10th of December 2006, and the actual traffic speed and the average commute time were used to determine the weights of each link.

### B. Simulation Environment

We first designed the urban roads based on the actual traffic data and generated multiple vehicles running on the roads to validate the usefulness of the algorithm for various parameter settings. To this end, we built virtual urban roads and sorted the roads into normal roads, three-way intersections and four-way intersections. We also added traffic control signals with proper settings.

In order to simulate real traffic situations (e.g. congestions, incidents, constructions, etc.), we considered the degree of saturation which is defined as the number of vehicles divided by the maximum number of vehicles on a link. It increases when more vehicles are running on a link and it causes the reduction of the overall traffic speed. The

traffic speed may be influenced by multiple factors such as the gap between each vehicle and traffic regulations, and is modeled as a cost equation in traffic engineering. In this research, we used the following common BPR equation (1) from Bureau of Public Road of the United States [6] to simulate real traffic situations by evaluating the expected commute times of links as

$$c_a(f_a) = t_a^0 \left[ 1 + k \left( \frac{f_a}{u_a} \right)^\beta \right], \forall a \in L \quad (1)$$

In (1),  $f_a$  represents the volume of traffic on link  $a$  (the number of vehicles),  $u_a$  the capacity of link  $a$  (the maximum number of the commuting vehicles),  $t_a^0$  the average commuting time on link  $a$ , and  $k$  and  $\beta$  are constants that vary according to the regional road conditions. Korea Research Institute for Human Settlements suggests 0.91 and 3 for  $k$  and  $\beta$ , respectively [7]. In our simulation, this expected commute times of links are computed at every second.

We also designed a visualization program to show the results by generating a log of the simulation outputs. Figure 4 displays a snapshot of a sample run of the visualization program that delivers an overall picture of the dynamic traffic conditions. Having chosen a vehicle on the visualization program (at the very left part of Figure 4), the program shows the predetermined optimal route and the real-time driving route in red and blue respectively (the upper right pane). Its simulation time graph (the bottom left pane) shows how fast the vehicle arrives at its destination. In addition, the program also shows the movements of other

cars which affect our model vehicle (the bottom right panes).

We considered three algorithms in our experiments. As mentioned previously, we generated multiple virtual vehicles on a given route to see how our system produces dynamic routes for changing traffic conditions and to compare with existing navigation systems that extract shortest routes from the predetermined shortest routes between nodes.

The first algorithm is the shortest route algorithm of Dijkstra [8]. It is guaranteed to find the shortest route, but it requires significant amount of computing time. Therefore, we just used pre-computed static routes. The second is a pseudo TPEG algorithm where the data on entire roads are updated every five minutes. (Note that the actual data are available only on the limited number of the roads). We, however, assume that there is approximately two minute data latency because of the data transmission process (since data must go through several information providers). Under the assumption, we scheduled each vehicle to receive new data every five minutes to see how it applies the updated data to searching for new routes and how the application influences the traffic condition. Again, the vehicles in our simulation used Dijkstra's algorithm to reroute. Because it takes more time to perform the calculation on mobile devices, we added one extra minute for the computation. Therefore, this process may cause a worst case possibility that a route is drawn from the data produced eight minutes earlier. In addition, we set the maximum number of routes drawn at 500 links, since the weight information is cast only for the major 10 percent of the links on the roads in Seoul in TPEG. The last is our ARS algorithm described in Section 3.

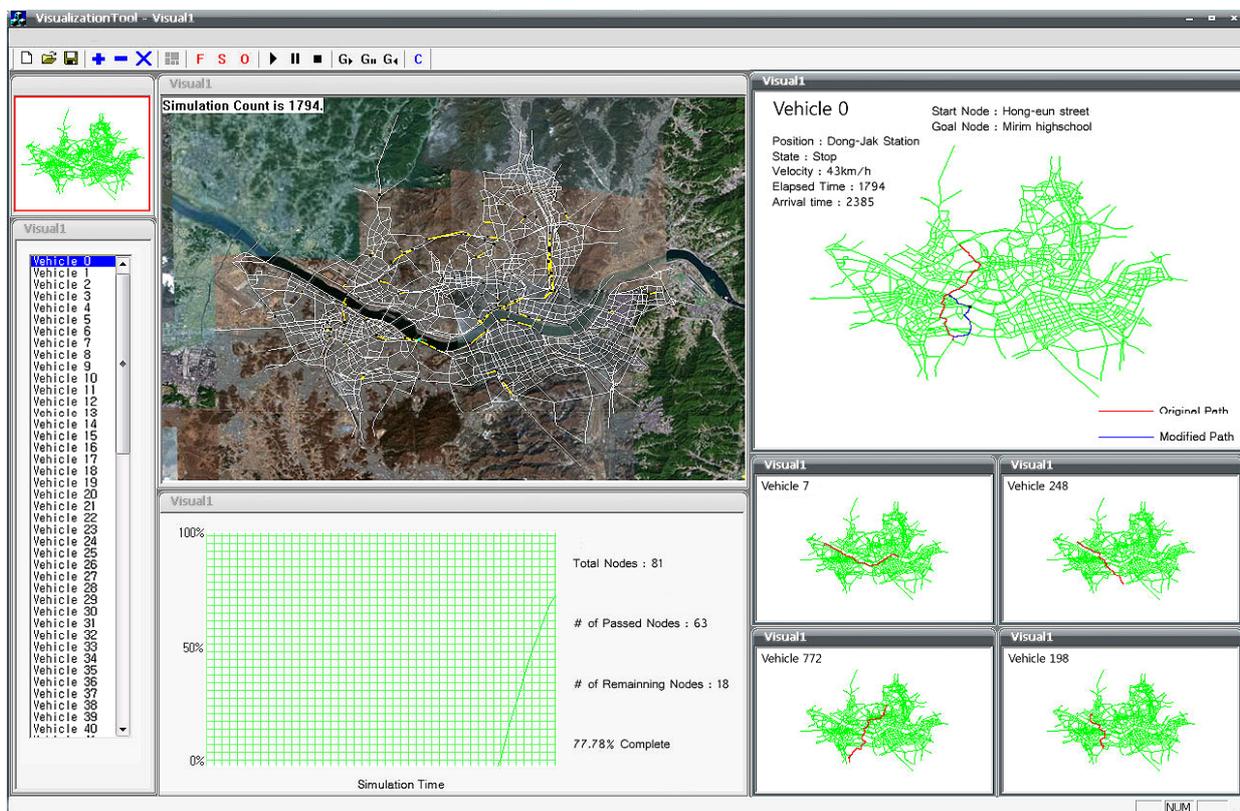


Figure 4. Sample run of visualization program

In addition to the route finding capability, we also considered the adaptability and the survivability of the algorithms. To confirm the adaptability we created virtual accidents on links. On the other hand, we examined the survivability by creating situations in which our vehicles are disabled and cannot perform bio-inspired route search to see how each system affects the others.

We compared the three algorithms based on the numbers of vehicles that managed to reach the end point (i.e. destination). All the vehicles are given the source and destination, and when a vehicle reaches the destination it goes back to the source and starts its journey again. By doing this we created the continuous traffic volume and compared the performance of the algorithms.

For the parameters of ARS, we set the size of the *updated list* of 1,000, weight propagation interval of 15 seconds, rerouting interval of 60 seconds, the size of the partial graph of within five links from the center point, and the data deleting term of 30 minutes when adjusting the size of the list.

### C. Results

Table 1 displays the results of our experiments. It shows the simulation time in seconds and the number of vehicles that successfully reached at the destination by each algorithm. As shown in Table 1, more cars managed to travel the routes by ARS than the other two algorithms, which implies that ARS can accelerate the traffic stream.

TABLE I. THE NUMBER OF VEHICLES REACHING AT DESTINATION

Situation	Environment		Algorithm		
	Simulation Time	# of vehicles	Dijkstra	TPEG	ARS
Normal	10,000	4,900	11,866	14,260	15,606
Survivability	8,000	10,000	16,928	19,163	21,591
Adaptability	10,000	4,900	11,696	13,790	15,181

Figure 5 shows the performance of each algorithm under normal traffic conditions. As shown, ARS performs less efficiently at the beginning of the simulation since there is little information propagated among systems. However, it surpassed other algorithms as simulation continued. This implies that ARS evolves and adjusts to the dynamic situations successfully.

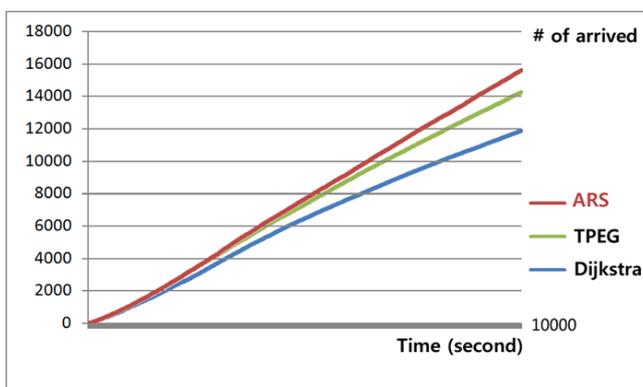


Figure 5. Performance in normal situation

Figure 6 depicts the number of vehicles that survived to the destination in the situation where 10% of the navigation systems failed in communication and the rest 90% was

functioning. It tells us that ARS guarantees better survivability from system failure because it enables each vehicle to actively collect data and act accordingly, so the remaining functioning systems assist the failed ones to survive despite the failure.

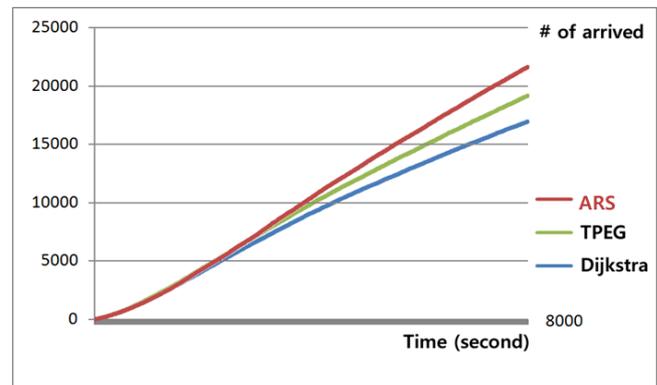


Figure 6. Survivability comparison

Figure 7 compares the number of vehicles that managed to reach the destination within a certain time frame in the event of significant increase of traffic congestion level because of sudden accidents. As shown, ARS handled a greater number of vehicles. This implies that ARS has higher adaptability toward unexpected traffic condition changes. The high survivability and adaptability proves the robustness of ARS.

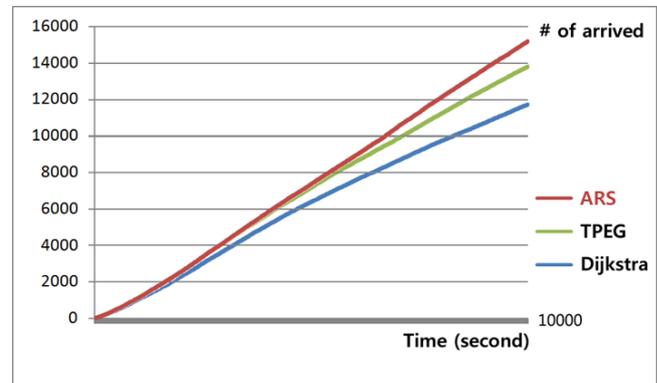


Figure 7. Adaptability comparison

## V. CONCLUSION

In this paper, we proposed a new GA-based route search algorithm, ARS algorithm, which finds the best routes dynamically considering road conditions. ARS showed its superiority over the general shortest-route search algorithm and centralized data broadcasting systems such as TPEG. It is also survivable and adaptable in a variety of dynamic traffic situations. By using ARS-based navigation systems, data can be propagated instantly and the service coverage can be expanded without going through the expensive process of centralized data collection from all the streets. Considering this, we believe that our algorithm will be able to produce a navigation system that reflects the real-time traffic data and entails reduced commuting time.

We expect to validate the usefulness of our algorithm by further adjusting parameters in more sophisticated ways and

enlarging the scale of simulations. We also expect that improved simulation models which better describe the actual road conditions, shall be feasible with data containing more detailed information. In addition, we are developing a new hybrid (e.g. Genetic Algorithm + Reinforcement Learning [9]) approach to find good routes in short time, expecting to improve our current algorithm. In the end, we hope to produce a new navigation system incorporating our ARS and to deploy in Korea.

#### ACKNOWLEDGMENT

The authors are grateful to Hyunjin Oh for his help in the design of the simulation tool and the visualization program.

#### REFERENCES

- [1] Explanation of Transport Protocol Experts Group, Dec. 2, 2008, Available : [http://www.ebu.ch/en/technical/projects/b\\_tpeg.php](http://www.ebu.ch/en/technical/projects/b_tpeg.php).
- [2] E. Moon, "The Introduction of Digital Multimedia Broadcasting in Korea," July 2006, Available : [http://ericemoon.googlepages.com/Introduction of DMB in Korea.pdf](http://ericemoon.googlepages.com/Introduction%20of%20DMB%20in%20Korea.pdf).
- [3] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Upper Saddle River, NJ: Addison-Wesley, 1989.
- [4] C. W. Ahn, R. S. Ramakrishna, and C. G. Kang, "A New Genetic Algorithm for Shortest Path Routing Problem," IEEE Transactions on Evolutionary Computation, Vol. 6, No. 6, Dec. 2002.
- [5] J. Pearl, Heuristics: Intelligent Search Strategies for Computer Problem Solving. Upper Saddle River, NJ: Addison-Wesley, 1984.
- [6] A. Nagurney and Q. Qiang. "Robustness of transportation networks subject to degradable links," A Letters Journal Exploring the Frontiers of Physics, 68001-p1-p6, 2007.
- [7] I. H. Jung, D. H. Son, "Survey of Highway Business Effect," Korea Research Institute for Human Settlements, 1995.
- [8] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," Numerische Mathematik, Vol. 1, pp. 269–271, 1959.
- [9] Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, 1998.
- [10] G. Harik, E. Cant' u-Paz, D. E. Goldberg, and B. L. Miller, "The Gambler's ruin problem, genetic algorithms, and the sizing of populations," Evol. Comput., Vol. 7, No. 3, pp. 231–253, 1999.
- [11] M. Munetomo, Y. Takai, and Y. Sato, "A Migration Scheme for the Genetic Adaptive Routing Algorithm," IEEE International Conference on Systems, Man, and Cybernetics, pp. 2774-2779, 1998.
- [12] J. Inagaki, M. Haseyama, and H. Kitajima, "A Genetic Algorithm for Determining Multiple Route and Its Applications," Proceedings of IEEE International Symposium on Circuits and Systems, pp. 137-140, 1999.
- [13] Y. Leung, G. Li, and Z. B. Xu, "A genetic algorithm for the multiple destination routing problems," IEEE Trans. Evol. Comput., Vol. 2, pp. 150–161, Nov. 1998.
- [14] Z. Xiawei, C. Changjia, and Z. Gang, "A genetic algorithm for multicasting routing problem," in Proc. Int. Conf. Communication Technology (WCC-ICCT 2000), pp. 1248–1253, 2000.
- [15] N. Shimamoto, A. Hiramatsu, and K. Yamasaki, "A dynamic routing control based on a genetic algorithm," in Proc. IEEE Int. Conf. Neural Networks, pp. 1123–1128, 1993.
- [16] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley.
- [17] X. Hue, "Genetic algorithms for optimization: Background and applications," Edinburgh Parallel Computing Centre, Univ. Edinburgh, Edinburgh, Scotland, Ver 1.0, Feb. 1997.
- [18] Sebastian Zander, Thuy Nguyen, and Grenville Armitage, "Automated traffic classification and application identification using machine learning," in Proc. of IEEE Conf. on Local Computer Networks, pp.250-257, 2005.