Hybrid Genetic Algorithm for Dynamic Multi-objective Route Planning with Predicted Traffic in a Real-World Road Network

Hitoshi Kanoh
Department of Computer Science
Graduate School of Systems and Information
Engineering
University of Tsukuba
Tsukuba Ibaraki 305-8573 Japan
kanoh@cs.tsukuba.ac.jp

Kenta Hara
Graduate school of Sciences and Engineering
University of Tsukuba
Tsukuba Ibaraki 305-8573 Japan
haraken@kalab.cs.tsukuba.ac.jp

ABSTRACT

Car navigation equipment in practical use has treated a route planning problem as a single-objective problem. In this paper, we formulate the problem as a dynamic multi-objective problem and show how it can be solved using a GA. There are three objective functions to optimize simultaneously in this problem: route length, travel time that changes rapidly with time, and ease of driving. The proposed method gives the Pareto-optimal set by using both the predicted traffic and a hybrid multi-objective GA (GA + Dijkstra algorithm) so that a driver can choose a favorite route after looking at feasible ones. We give the results of experiments comparing the proposed method with the Dijkstra algorithm and the single-objective GA in applications with a real road map and real traffic data in wide-area road network.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic methods*, *Scheduling*.

General Terms

Algorithms, Experimentation.

Keywords

Multi-objective optimization, genetic algorithm, Dijkstra algorithm, hybrid, dynamic, planning, route, prediction, traffic, road network, real-world, transportation.

1. INTRODUCTION

Car navigation equipment is the most widely used form of information terminal for Intelligent Transportation Systems. In Japan there are more than 28 million vehicles equipped with these terminals, two thirds of which use real-time information on traffic

and can inform drivers not only the location of traffic congestion but also the amount of time needed for passing through it.

In the context of car navigation equipment, a route-planning problem is a multi-objective optimization problem (MOP) for finding feasible routes from an origin to a destination on a road map so that a driver can choose a favorite route. There are three objective functions to optimize simultaneously in this problem: route length, travel time, and ease of driving (as reflected by factors such as the number of turns and the width of the roads). The travel time for a route changes rapidly with time and the other functions do not, and we assume here that it can be evaluated by considering the predicted traffic calculated from real-time information on traffic. The traffic at a certain place can thus be given as a function of time, but the search space in this problem is larger than that in conventional static route-planning problems. Furthermore, in the MOP the shape of the Pareto front is complex because the ease of driving is expressed by constraints.

A typical solution to conventional route planning problems is the Dijkstra algorithm (DA) [1], which has been used in road networks [2, 3] and in dynamic environments [4] to find a route minimizing the travel time from an origin to a destination. The DA generates a single solution from each run, so several runs using different parameters are required in order to generate several elements of the Pareto optimal set. Genetic algorithms (GAs), in contrast, deal with a set of feasible solutions in a single run and are less sensitive to the shape of the Pareto front.

Many studies using GAs to solve multi-objective route planning problems have been reported, which include a review of recent issues [5] and applications to mobile robots [6], personal navigation [13], tourist sight-seeing itinerary [14], and car navigation system [15]; but none has applied one to dynamic problems. There are also papers treating dynamic environment [16-19], but no example applied to route planning in wide-area road network is found.

In this paper we formulate a dynamic route planning problem as a multi-objective problem with predicted traffic and show how it can be solved using a genetic algorithm. A GA-based route planning algorithm in a dynamic environment has already been reported [7], but it gives only one compromised solution because its objective function is the weighted sum of the three objective

functions. It also very rapidly generates detours using other routes whenever traffic conditions change, so the route it finds cannot be the global optimum. The method we propose here gives the Pareto-optimal set by using both the predicted traffic and a multi-objective GA. Furthermore, the routes calculated by the DA are used in generating the initial population that consists of a small number of high-quality routes.

In Section 2 of this paper we formulate the problem and explain our strategies for solving it, in Section 3 we describe the algorithm of the proposed method in detail, and in Section 4 we give the results of experiments comparing the proposed GA with the DA and the single-objective GA in applications with a real road map and real traffic data in wide-area road network.

2. PROBLEM DESCRIPTION AND STRATEGIES

2.1 Background

Vehicle detectors such as ultrasonic sensors, optical beacons, video cameras, and loop detectors are installed at more than 20,000 locations along principal roads throughout Japan. They measure the average travel time of vehicles and the number of vehicles passing through specific road intervals (links) in 5-minute intervals. Most of the car navigation equipment currently available in Japan uses this traffic data for route planning but does not take predicted traffic into account. As a result, a driver whose route includes a congested road that was not congested at the time the route was planned will arrive after the time originally estimated. Discrepancies between actual and computed travel times also occur when links not equipped with detectors become congested because the travel times for those links are calculated using standard times.

These problems cannot be solved without taking predicted traffic into account when planning routes. A method for making spatial interpolations of traffic data has already been proposed [8]. It can estimate traffic on links not equipped with detectors from the traffic on links equipped with detectors, and in this paper we describe a method for taking the predicted traffic into account when planning routes.

2.2 Formulation of Problem

In this section we extend the formulation for conventional real-world route-planning problems [7] and give one for multi-objective route planning problems with predicted traffic. We suppose a road network *Net* with a set of nodes $N = \{N_1, N_2, \ldots\}$ and a set of links $L = \{L_1, L_2, \ldots\}$. Each node has either a *with signal* or *without signal* property, and each link has *length of link*, *road class*, and *number of lanes* properties. Examples of road class are listed in Table 1. We treat a problem formulated by (*Net*, *O*, *D*, *F*, *C*), where *O* and *D* are an origin and a destination, *F* is a set of objective functions, and *C* is a set of constraints.

The constraints can be classified into *hard constraints*, corresponding to traffic regulations, and *soft constraints* (Table 2), corresponding to the ease of driving. Hard constraints must be satisfied, but soft constraints can be violated at the cost of a penalty. We regard the route with the lowest total penalty as the easiest one for drivers.

Consider the set of objective functions

Table 1. Examples of road class.

Category	Road class
1	Expressway, National highway, Principal local road
2	Prefectural road
3	Municipal road

Table 2. Constraints and their penalties.

Object of constraint	Constraint	Penalty
Node	To reduce the number of signals	P_{Sig}
Link	To select a arterial road	P_{Art}
	To select a wide road	P_{Wide}
	To avoid traffic jam	P_{Jam}
Route	To reduce the number of turns	P_{Turn}

$$F = \{f_l, f_T, f_C\}, \tag{1}$$

where f_l and f_T are respectively the length and the travel time of the route, and f_C is the total penalty. Let l_i and $T_i(t)$ respectively be the length and the travel time at time t of link L_i along the route $(O, N_1, \ldots, N_n, D)$. Figure 1 shows the relationships between these quantities, so functions f_l and f_T can be written as follows:

$$f_l = \sum_{i=1}^{n+1} l_i \tag{2}$$

$$f_T = \sum_{i=1}^{n+1} T_i(t_{i-1}), \tag{3}$$

$$\text{where,} \quad t_{i-1} = \begin{cases} t_0 & \text{for } i = 1 \\ t_{i-2} + T_{i-1}(t_{i-2}) & \text{for } i > 1 \end{cases}$$
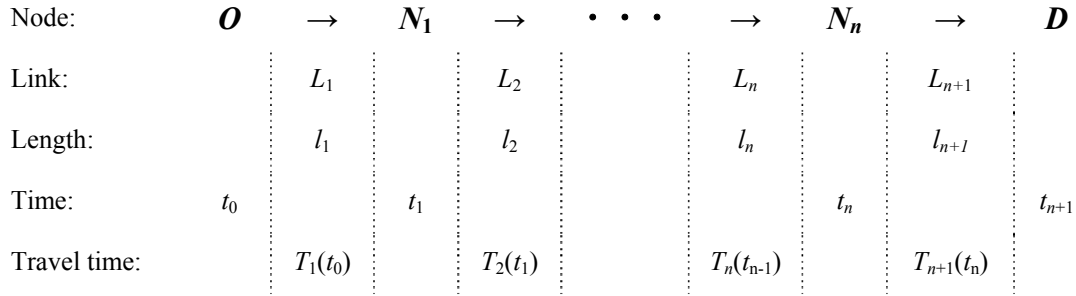
Function f_C is

$$\begin{aligned} f_C = Sum(P_{Sig}) + Ave(P_{Art}) + Ave(P_{Wide}) \\ + Sum(P_{Jam}) + Sum(P_{Turn}) \end{aligned}, \tag{4}$$

where P_{Sig}, P_{Art}, P_{Wide}, P_{Jam}, and P_{Turn} are the penalties listed in Table 2 and $Sum(X)$ and $Ave(X)$ are respectively the sum and the average of the penalty along the route.

2.3 Strategies for solution

Among the currently available overviews of evolutionary algorithms to solve MOPs are those analyzing the state of the art [9], that are tutorial [10], and that provide a historical view [11]. The method that we will describe in this paper is different from those algorithms in that it is specialized for the problem. Wanting

Node:	O	$\rightarrow$	N_1	$\rightarrow$	$\cdots$	$\rightarrow$	N_n	$\rightarrow$	D
Link:			L_1		L_2		L_n		L_{n+1}
Length:			l_1		l_2		l_n		l_{n+1}
Time:	t_0			t_1				t_n	t_{n+1}
Travel time:			$T_1(t_0)$		$T_2(t_1)$		$T_n(t_{n-1})$		$T_{n+1}(t_n)$

Figure 1. Relationship between node, link, length, time, and travel time along route (O, N_1, …, N_n, D).

our algorithm to provide the quasi-Pareto optimal set rapidly, and we used the following two strategies.

- To get a set of feasible solutions in a single run, we use a multi-objective GA whose objective functions are the length of a route, the travel time, and the ease of driving.

- We assume that predicted traffic for all links in a target map is given. It can be obtained by using the previously reported method [8].

To generate an initial population that consists of a small number of high-quality individuals (routes), we use the virus GA described in reference [7] and the hybrid GA which is specialized for this problem.

- A part of an arterial road is regarded as a partial solution (we call it virus). Only routes that include viruses are generated as the initial population.

- The routes calculated by the DA are used in generating the initial population. This is intended to favor the use of non-arterial roads and provide stable solutions.

Finally, we use the following strategy to generate nondominated solutions in successive generations.

- Offspring are generated from individuals with the best value of each objective function in the population.

3. PROPOSED METHOD

3.1 Chromosome Representation

We regard a route from O to D as an individual and express it by a variable-length sequence of intersection symbols. Let R_k be the k-th individual in the population. Here $R_k = (R_k(0), R_k(1), …, R_k(n_k), R_k(n_k+1))$, n_k is the number of nodes, $R_k(i)$ is the i-th node, $R_k(0) = O$, and $R_k(n_k+1) = D$. We also regard arterial roads on the target map as viruses and express them in the same manner. Let $V_j = (V_j(1), …, V_j(m_j))$ be the j-th virus in the population of viruses, where m_j is the number of nodes on V_j. Viruses do not need to include an origin and a destination, while individuals must include them.

3.2 General Procedure

The general procedure of the proposed method is given below. The prediction and the interpolation have already been reported [8], and mutation is not used in this method. The other operations are described in detail in the following sections.

```
procedure main() {
    input an origin and a destination;
    input a map that includes the origin and
     the destination;
    input traffic data;
    perform prediction and interpolation of
     the travel time for all links;
    generate a population of viruses;
    generate a population of individuals
     using the Dijkstra Algorithm;
    repeat until upper limit of generation {
        calculate objective functions of the
         individuals;
        selection and crossover;
        infection;
    }
    select nondominated solutions from the
     population;
    output them as recommended routes;
}
```

3.3 Initial Population

An initial population of individuals (routes) is generated using the roads of categories 1 and 2 (see Table 1). Since links on these roads have road names such as Route 17 and are numbered serially on maps, the links can be taken as separate roads. The proposed method should not be used if all the roads in the target map are municipal roads (category 3) because no real-time information about traffic on these roads is provided. For them, only conventional techniques are necessary.

The procedure of generating a population of viruses and an initial population of individuals is given below. Steps 1 to 3 correspond to the population of viruses and steps 4 to 5 correspond to the population of individuals. Here the distances $d(O, D)$, $d(O, V_j)$, and $d(D, V_j)$ are respectively those from the origin to the destination, between the origin and the node on V_j that is the one nearest to the origin, between the destination and the node on V_j that is the one nearest to the destination. And DA(length) and DA(time) are respectively the Dijkstra algorithms whose cost

functions are the length of a route (f_l) and the travel time calculated with traffic when a car starts (f_{T0}). Here

$$f_{T0} = \sum_{i=1}^{n+1} T_i(t_0) \cdot \qquad\qquad (5)$$

In addition, the computational time for DA(time) is vanishingly small because that algorithm does not use predicted traffic (note the difference between expressions (3) and (5)).

[Step 1] Let all the roads of categories 1 and 2 within the rectangle on the map with a diagonal that links the origin and the destination be a set of viruses.
[Step 2] Viruses such that $d(O, V_j) + d(D, V_j) > d(O, D)$ are removed from the set of viruses.
[Step 3] Let this set of viruses be a population of viruses $\{V_j \,|\, j = 1, …, N_v\}$, and for each V_j let the node nearest the origin be the first node $V_j(1)$.
[Step 4] For $j = 1$ to N_v generate both the route from O to $V_j(1)$ and the route from $V_j(m_j)$ to D by using DA(time). The route that combines these two routes with V_j becomes the j-th individual R_j.
[Step 5] Generate two routes from O to D by using DA(time) and DA(length), and add them to the population of individuals.

3.4 Selection and Crossover

We use one point crossover that produces Offsprint-1 and Offspring-2 from Parent-1 and Parent-2. Let a set {Parent-1, Parent-2, Offspring-1, Offspring-2} be called a family, and let $P(t)$ and $P'(t)$ be the populations of individuals and candidate individuals at generation t.

Here are the directions for the process at generation t.

[Step 1] An individual with the smallest value of the objective function f_l is selected from $P(t)$ as Parent-1.
[Step 2] Steps 3 to 5 are applied to all the remaining individuals that have a node in common with Parent-1. If there are two or more common nodes, one will be chosen randomly.
[Step 3] Let an individual be Parent-2.
[Step 4] One point crossover is performed.
[Step 5] If a parent is nondominated by the rest of the family and no identical individual exists in $P(t+1)$, then it is added to $P(t+1)$. If offspring is nondominated by the rest of the

Figure 3. Map of the central part of Tokyo.

family and no identical individual exists in $P'(t+1)$, then it is added to $P'(t+1)$.
[Step 6] The objective function f_l in step 1 is changed into f_T, and steps 1 to 5 are performed once again.
[Step 7] The objective function f_l in step 1 is changed into f_C, and steps 1 to 5 are performed once again.
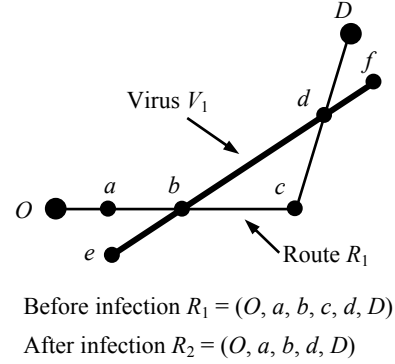[Step 8] Individuals in $P'(t+1)$ that are nondominated by the rest of the individuals are added to $P(t+1)$.

Before infection $R_1 = (O, a, b, c, d, D)$
After infection $R_2 = (O, a, b, d, D)$

Figure 2. Example of infection.

3.5 Infection

Infection is introduced as a genetic operation so as to generate the combination of arterial roads in successive generations. The infection substitutes the nodes of the virus for the nodes of the route if the route has two common nodes with the virus. In Fig. 2, for example, if the route before infection is $R_1 = (O, a, b, c, d, D)$ and the virus to infect is $V_1 = (e, b, d, f)$, then the route after infection is $R_2 = (O, a, b, d, D)$, where b and d are common nodes.

The procedure of infection at generation t is given below.

[Step 1] An individual with the smallest value of the objective function f_l is selected from $P(t)$ as Host-1.
[Step 2] Steps 3 to 5 are applied to all the viruses that have two nodes in common with Host-1. If there are three or more common nodes, two will be chosen randomly.
[Step 3] Let a virus be Virus-1.
[Step 4] Infection is performed between Host-1 and Virus-1.
[Step 5] The individual after infection is added to $P'(t+1)$.
[Step 6] Individuals in $P'(t+1)$ that are nondominated by the rest of the individuals are added to $P(t+1)$.
[Step 7] Let $P'(t+1)$ be empty.
[Step 8] The objective function f_l in step 1 is changed into f_T, and steps 1 to 7 are performed once again.
[Step 9] The objective function f_l in step 1 is changed into f_C, and steps 1 to 6 are performed once again.

4. EXPERIMENTS
4.1 Experimental Methods

The performance of the proposed method was evaluated using the historical traffic data measured by vehicle detectors and the standard map whose format was developed and established by the Navigation System Researchers' Association [12]. The map of the central part of Tokyo that was used in the experiments, which includes 19,963 nodes and 58,222 links, is shown in Figure 3. The

traffic data on June 17, 2003 was used, and the values of the penalties applied are listed in Table 3. Expressways are considered congested when the speeds on them are less than 40 km/h, and ordinary roads are considered congested when the speeds on them are less than 10 km/h.

The multi-objective GA without predicted traffic (MOGA) and the single-objective GA without predicted traffic (SOGA) were compared with the proposed method. The set of objective functions of the MOGA and the fitness function of an individual of the SOGA are as follows (see expressions (4) and (5)).

$$F^{MOGA} = \{f_l, f_{T0}, f_C\} \tag{6}$$

$$F^{SOGA} = Norm(f_{T0}) + Norm(f_C), \tag{7}$$

where

$$Norm(x) = 1 - \frac{x - \min(x)}{\max(x) - \min(x)}.$$

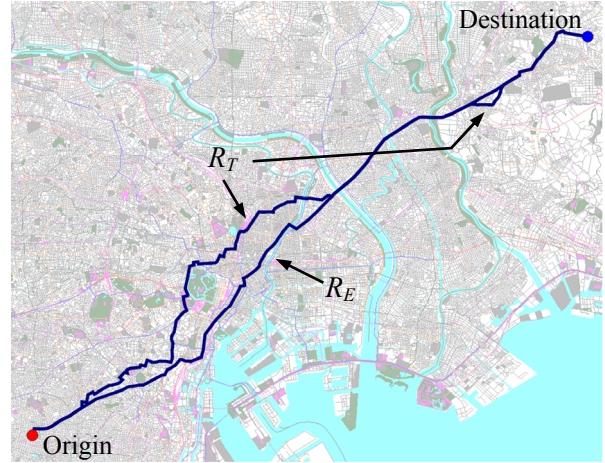Here $\max(x)$ and $\min(x)$ are respectively the maximum and the minimum values of x in the population.

Figure 4. Examples of routes generated by the proposed method.

Figure 5. Example of routes generated by the SOGA (the elitest individual).

Table 3. Values of penalties applied.

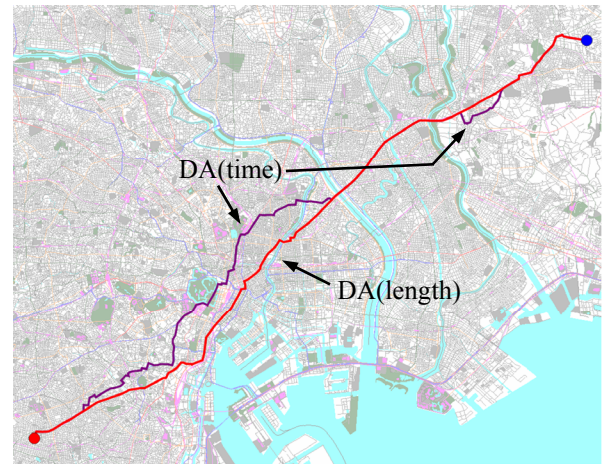| Penalty	Condition	Value
P_{Sig}	One signal	2
P_{Art}	Expressway	0
	National highway	0
	Principal local road	0
	Prefectural road	2
	Municipal road	4
P_{Wide}	Six-lane road	0
	Four-lane road	0
	Two-lane road	1
	One-lane road	3
P_{Turn}	No turn	0
	Left-hand turn	3
	Right-hand turn	7
	U-turn	20

Figure 6. Examples of routes generated by DA(time) and DA(length).

Table 4. Objective functions and performance indices of routes shown in Figs. 4 to 6.

| Objective function and performance index	Proposed method		SOGA	DA(time)	DA(length)
	R_T	R_E			
Travel time (s)	6303	6478	6478	6439	6657
Route length (m)	43282	39135	39135	43252	39104
Total penalty	1553	903	903	1795	1023
Number of signals	152	140	140	156	144
Number of right turns	35	15	15	35	16
Number of left turns	35	17	17	34	18
Arterial road ratio	0.46	0.83	0.83	0.43	0.81
Congestion length (m)	528	652	652	1716	890
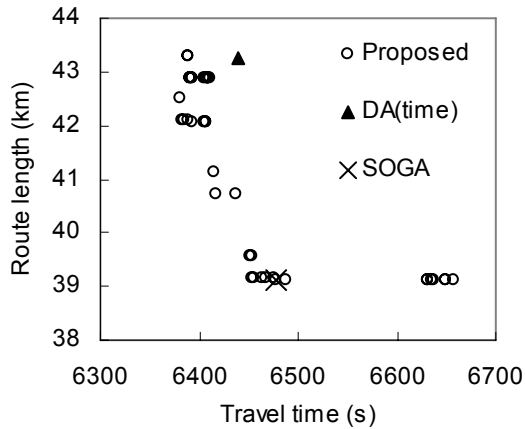
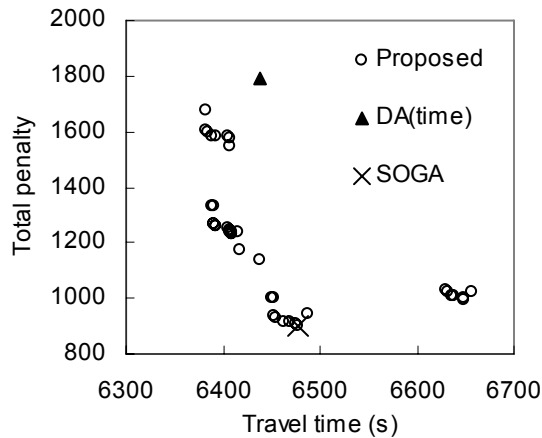**Figure 7. Distribution of solutions between travel time and route length.**

Figure 8. Distribution of solutions between travel time and total penalty.

4.2 Examples of routes

Figures 4 to 6 show examples of routes generated by the proposed method, the SOGA, and the DAs when a car starts at 12:00. In Fig. 4, R_T and R_E respectively indicate the routes with the shortest travel time and the lowest total penalty in the set of solutions. The performance indices of the routes shown in Figs 4 to 6 are listed in Table 4, where the arterial road ratio is the ratio of the total length of roads belonging to category 1 or 2 to the length of the whole route. Examining the results listed in Table 4, one sees the following:

- R_T is better than the route found by DA(time) in terms of travel time and congestion length, but the other indices of these routes are almost same.
- R_E is exactly the same as the route found by the SOGA.

In addition, the route found by DA(length) was the shortest and was included in the set of routes found by the proposed method.

4.3 Distribution of solutions

Figure 7 shows the distribution of solutions between the travel time and the length of a route. These solutions were generated in single runs of the proposed method, the SOGA, and DA(time). Figure 8 shows the corresponding distribution for the travel time and the total penalty. Note that the proposed method gave only nondominated solutions in the population (see section **3.2**). We can see from Figs. 7 and 8 that the set of solutions generated by the proposed method is widely distributed in the Pareto front.

4.4 Effectiveness of Predicted Traffic

To examine the effectiveness of introducing predicted traffic into the route planning, we compared the proposed method with the MOGA and DA(time). Here, DA(time) may be the route-planning method used in commercial car navigation equipment (the details of that method are undisclosed).

Table 5 shows the minimum value of the travel time for each method (each entry listed in Tables 5, 6, and 7 is the average of 20 trials). The first row indicates the time a car starts out. One sees from the data listed in Table 5 two things:

- The proposed method is better than the MOGA at all times, and the effectiveness of predicted traffic can be confirmed.

- The MOGA is better than DA(time) at all times. This means that in a dynamic environment the length of a route and the ease of driving should be considered even when priority is given to the travel time.

The minimum values of the total penalty for the proposed method and the MOGA are listed in Table 6, from which one can see that the proposed method is better than the MOGA at all times. One can see why this is so from the performance indices listed in Table 7: the congestion length of the proposed method was only about 74% of that of the MOGA and the proposed method is effective for the avoidance of congestion.

5. CONCLUSIONS

Car navigation equipment in practical use has treated the route-planning problem as a single-objective problem, and users of this equipment have had to decide the priority of objective functions before planning. In this study the problem was described as a multi-objective problem so that a user can chose a favorite route after looking at feasible ones. Furthermore, the route planning in three-dimensional travel time space including predicted traffic was performed in a wide area road network.

The evaluation presented in this paper used only one map and the data for one day. More systematic experiments should be done in future work.

6. ACKNOWLEDGMENTS

7. REFERENCES

[1] Golden, B., Shortest-path algorithms - a comparison -, Operations Research, Vol. 24, No.9, pp. 1164-1168, 1976.

[2] Chan, E. P. F. and Zhang, N., Finding shortest paths in large network systems, Proceedings of the 9th ACM international symposium on Advances in geographic information systems, pp. 160-166, 2001.

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

[4] Pattanamekar, P., Park, D., Rilett, L.R., et al., Dynamic and stochastic shortest path in transportation networks with two components of travel time uncertainty, Transportation Research Part C 11, pp. 331-354, 2003.

[5] Maria, J., Pangiliana, A., and Janssens, G. K., Evolutionary algorithms for the multiobjective shortest path planning problem, International journal of computer and information science and engineering, Vol. 1, No. 1, pp. 54-59, 2007.

[6] Castillo, O., Trujillo, L., and Melin, P., Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots, Soft computing – A Fusion of foundations, methodologies and applications, Vol. 11, No. 3, pp. 269-279, 2007.

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

[8] Kanoh, H., Furukawa, T., Tsukahara, S., Hara, K., Nishi, H., and Kurokawa, H., Short-term traffic prediction using fuzzy c-means and cellular automata in a wide-area road network, IEEE International Conference on Intelligent Transportation Systems, ITSC'2005, pp. 984-988, 2005.

[9] Van Veldhuizen, D.A and Lamont, G.B., Multiobjective evolutionary algorithms: Analyzing the state-of-the-art, Evolutionary Computation, Vol. 8, No. 2, pp. 125-147, 2000.

[10] Zitzler, E., Laumanns, M., and Bleuler, S., A tutorial on evolutionary multiobjective optimization, In Xavier Gandibleux et al, editor, Mathematics for Multiobjective Optimizationpp.3-37, Berlin, Springer. Lecture Notes in Economics and Mathematical Systems Vol. 535, 2004.

[11] Carlos, A., Evolutionary multi-objective optimization: a historical view of the field, IEEE Computational Intelligence Magazine, February, pp. 28-37, 2006.

Table 5. Minimum values of travel time (s) (average of 20 trials).

Start time	9:00	12:00	15:00	18:00	21:00
R_T	6113	6380	6523	5512	4648
MOGA	6309	6382	6871	5516	4815
DA(time)	6476	6439	7078	5521	4815

Table 6. Minimum values of total penalty (average of 20 trials).

Start time	9:00	12:00	15:00	18:00	21:00
R_E	947	903	974	751	691
MOGA	987	996	1012	754	716

Table 7. Performance indices (average of 20 trials).

Performance index	R_E	MOGA
Total penalty	903	996
Number of signals	140	144
Number of right turns	15	16
Number of left turns	17	18
Arterial road ratio	0.83	0.81
Congestion length (m)	652	890

[12] Navigation System Researchers' Association, http://www.naviken.jp/.

[13] Shiraishi, T, et al., A personal navigation system with a schedule planning facility cased on multi-objective criteria, International conference on mobile computing and ubiquitous networking (ICMU 2005), pp.104-109, 2005.

[14] Huang, B., Yao, L., and Raguraman, K., Bi-level GA and GIS for multi-^objective TSP route planning, Transportation planning and technology, Vol. 29, No. 2, pp.105-124, 2006.

[15] Chakraborty, B., Maeda, T., and Chakraborty, G., Multiobjective route selection for car navigation system using genetic algorithm, IEEE Mid-Summer workshop on soft computing in industrial application (SMXia/05), pp. 190-195, 2005.

[16] Farina, M., Deb, K., and Amato, P., Dynamic multiobjective optimization problems: test case, approximations, and applications, IEEE Transactions on evolutionary computation Vol. 8, No. 5, pp. 425-442, 2004.

[17] Hatzakis, I. and Wallace, D., Dynamic multi- objective optimization with evolutionary algorithms: A forward-looking approach, Genetic and evolutionary computation conference (GECCO'06), pp. 1201-1208, 2006.

[18] Lui, C. and Wang, Y., Dynamic multi-objective optimization evolutionary algorithm, International conference on natural computation (ICNC'07), Vol. 4, pp. 456-459, 2007.

[19] Zhou, A., et al., Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization, Lecture note in computer science, Vol. 4403, pp. 832-846, 2007.