Using Genetic-Fuzzy Methods To Develop User-preference Optimal Route Search Algorithm

Gyoo-seok Choi* · Jong-jin Park**

Abstract

The major goal of this research is to develop an optimal route search algorithm for an intelligent route guidance system, one sub-area of ITS. ITS stands for Intelligent Transportation System. ITS offers a fundamental solution to various issues concerning transportation and it will eventually help comfortable and swift moves of drivers by receiving and transmitting information on humans, roads and automobiles. Genetic algorithm and fuzzy logic are utilized in order to implement the proposed algorithm. Using genetic algorithm, the proposed algorithm searches shortest routes in terms of travel time in consideration of stochastic traffic volume, diverse turn constraints, etc. Then using fuzzy logic, it selects driver-preference optimal route among the candidate routes searched by GA, taking into account various driver's preferences such as difficulty degree of driving and surrounding scenery of road, etc. In order to evaluate this algorithm, a virtual road-traffic network DB with various road attributes is simulated, where the suggested algorithm promptly produces the best route for a driver with reference to his or her preferences.

※ 본 연구는 청운대학교 교내연구비 지원에 의해서 수행되었음.
* Dept. of Computer Science, ChungWoon Univ.
** Dept. of Internet Computer, ChungWoon Univ.

1. Introduction

As the traffic congestion in a metropolitan area such as Seoul has been serious due to an increasing number of vehicles beyond road capacity, the development of a dynamic route-guidance system that guides an optimal route between the origin and the destination, and helps drivers to avoid a partial traffic jam in a city-road network, is highlighted[1, 2, 4]. But this system does not provide sophisticated route guidance more than simple route guidance on the ground of traveling distance or traveling time, because it does not consider driver's preferences. For instance, beginners are reluctant to drive on a complex route in such as congest market area which needs an experienced driving technique.

Some unoccupied drivers who don't need to rush prefer a scenic route commanding a fine view such as a riverside road rather than a simply fast route. It is preferable for an intelligent route guidance system to provide user-optimal guidance for the driver. And in the aspect of traffic management, it would rather be desirable to present several alternatives than a single shortest route to drivers whose travel routes(O-D pairs) are similar, so that it may elevate the transport efficiency of overall road network by decentralizing the entire traffic volume. Therefore, we develop a driver-optimal route search algorithm to be effectively used to an intelligent route guidance system, which considers diverse demands of a driver such as a difficulty degree of driving, surrounding scenery of a road, etc. In order to implement the proposed algorithm, Genetic Algorithm(GA) and fuzzy logic are used.

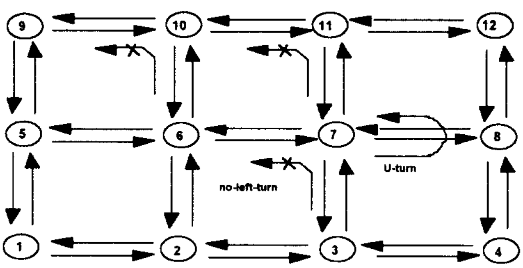
Nowadays, GA is effectively applied to the search of a global optimal solution in diverse areas that have multi-variables, because it is difficult to find an optimal solution in problems with multi-variable. We developed a new path chromosome and genetic operations based the GA-path crossover and path mutation. The suggested genetic method can provide multiple alternatives at one trial, because it is based on the concept of evolution of path individuals and presents multiple superior ones in the final generation. Fuzzy logic theory provides a mathematical framework that can handle fuzziness in the real world. Using fuzzy logic, the proposed route search algorithm considers driver's preferences and finds driver-preference optimal route among the searched route candidates by GA.

2. Representation of Road Network DB and Path

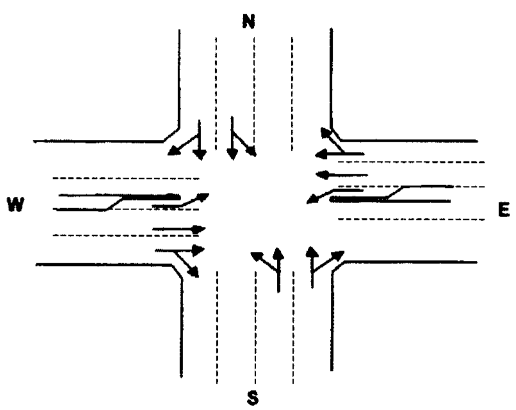
2.1 Road Network and Data Structure

A road network is generally represented by a weighted digraph $G = (N, L)$, consisting of a set of nodes $N = n_1, n_2, ..., n_m$ that represent crosses and a set of links $L = l_1, l_2, ..., l_n$ that represent road segment between two adjacent crosses. A link l_k in L represents an ordered pair of nodes (i, j). Let o(origin) and d(destination) be two given nodes of (N, L). The chain from o to d in (N, L) is a sequence of nodes and links, of the form $o = n_p, l_p, n_{p-1}, l_{p-1}, ..., l_{q-1}, n_q = d$, and is called a path. Moreover, when a path has the smallest link cost among all the paths, it is said to be the shortest path[2].

A typical city-road network is grid shaped as shown in (Figure 1). There are some constraints in real road networks such as U-turn prohibition, left-turn prohibition, which do not exist general networks. For example, paths including loops can also be selected as shortest paths. As shown in (Figure 1), with origin 3 and destination 9, a path 3-7-8-7-6-5-9 including a U-turn can be selected as the shortest path.



(Figure 1) Representation of typical road network



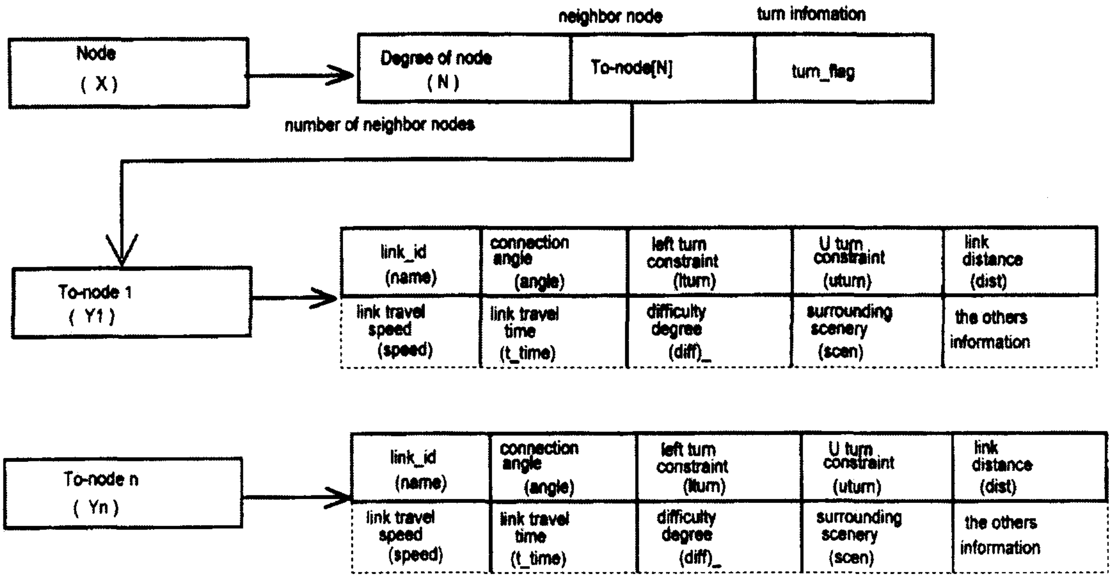
(Figure 2) Structure of general 4-cross in a road network

(Figure 2) represents the physical structure of a cross, considering actual road traffic. There are many methods to represent a cross according to the object oriented, which have each other computational loads. Generally speaking, the representation of cross in a road network can be implemented by a single node or four nodes for modeling turn status.

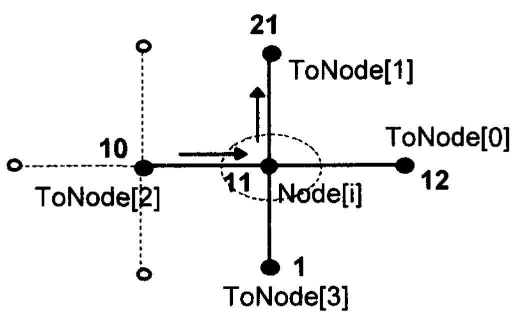
The method by four nodes is easy to descript turn status, but on the other hand it follows enormous computational loads according to increasing the number of nodes. Therefore we represented a cross with a single node for alleviating computational loads, and added turn information for processing turn status. We designed the data structure of a road network as node-oriented structure depicted in (Figure 3) for simulating the proposed algorithm in a tentative road network, and so we can reduce computational loads to search proper O-D paths and process effectively several road attributes, such as node and link information, turn constraints, drivers' preferences, etc.

2.2 Turn Constrains and Path

Turn status in traveling a path is grasped by using connection angle between two adjacent nodes in the developed algorithm. Connection angle is defined as an angle which the current node is connected with its neighboring node by a link in traveling a path and also it is defined as a method that the angle is increased counter-clockwise. Turn status in traveling a path is effectively seized by the difference of connection angle between two adjacent nodes (see (Figure 4)). If the difference of connection angle is 90 degree, turn status is right-turn . If the



(Figure 3) The data structure of road network DB (topology)



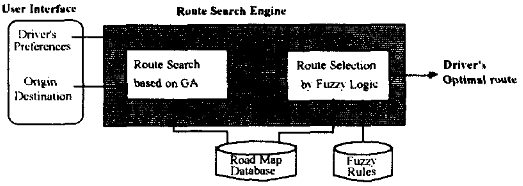
(Figure 4) Representation example of connection angle between adjacent nodes

difference of connection angle is 180 degree, turn status is U-turn. And if the difference of connection angle is 270 degree(or - 90 degree), turn status is right-turn. In doing so, appropriate margin value($\pm$20 degree) is set for preparing the case that the difference of connection angle is exactly agreed (ex. 80 degree).

Turn constrains to neighboring node is classified with several cases, according to the status of constraints (no U-turn and no left-turn to all neighboring nodes, no U-turn to all neighboring nodes and no left-turn to some neighboring nodes, etc.). It is effectively represented and processed by setting the value of the appropriate data field, using turn-flag as depicted in (Figure 3). The path individual as a result of the search is represented as a node set which are composed with the path, and it is saved in path array. Here, we represent the searched path as the path individual, assuming that it is an evolutionary living creature in the aspect of GA(Genetic Algorithm). When the path individual is saved in path array, -1 is added next to terminal node in order to describe the end of the path, because the length of each path is different.

3. Genetic-Fuzzy Approach

Drivers have many kinds of their criteria for route selection. There are travel time, travel distance, difficulty of driving, scenery of road, avoiding or favoring the specific region and road types. The first two of these has been the two major attributes used for route selection. For the shortest time seems to be a more important criterion to avoid heavy traffic nowadays. The remaining criteria are less important than the above two, but have to be considered to make drivers decide the best choice. Therefore, it is a good approach that the route guidance system finds several routes in terms of travel time and provides drivers with the best one of them, with reference to their preferences. This method can satisfy various user preferences efficiently. We suggest genetic-fuzzy route search system. The architecture of the proposed system is shown in (Figure 5). In this, the genetic algorithm provides multiple superior routes in order of travel time in the first stage. In the next stage, the fuzzy logic is used to make a decision for selecting an optimum route among them. In this research, difficulty of driving and scenery of road are used as assessment criteria to find the best route for a driver.



(Figure 5) The Architecture of Genetic-Fuzzy Route Search system

3.1 Genetic Algorithms

Genetic algorithms consider many points from the search space simultaneously and therefore have a reduced chance of converging to local optimum. The process by which populations are generated and tested is similar to a natural population of biological creatures in which successive generations of organisms are produced and raised until they themselves are ready to reproduce. In most conventional search techniques, a single point is considered based on some decision rules. These methods can be dangerous in multi-modal search spaces because they can converge to local optimum. However, GAs generate entire populations of points, test each point independently, and combine qualities from existing points to a new population containing improved points. Aside from producing a more global search, the GA's simultaneous consideration of many points makes it highly adaptable to parallel processors, since the evaluation of each point is an independent process. A genetic algorithm in its simplest form uses three operators : Reproduction, Crossover, and Mutation[7, 8].

1) Reproduction : Reproduction is a process in which individual strings are copied according to their objective function values, f(fitness function). Intuitively, we can think of the function f as some measure of profit, utility, or goodness that we want to maximize. The strings with a higher value have a higher probability of contributing one or more offspring and those with a lower value have a lower probability of contributing one or less offspring in the next generation. To achieve this, the strings are selected ac-

cording to what has become known as the stochastic remainder selection without replacement.

2) Crossover : Simple crossover may proceed in three steps. First, two newly reproduced strings are selected from the mating pool formed through reproduction. Second, a position along the two strings is selected uniformly at random. For example, the following binary coded string A and B of length 8 are shown aligned for cross over

A : 01011 | 101
B : 10010 | 010

Notice how crossing site five has been selected in this particular example through random choice, although any of the other three positions were just as likely to have been selected. The third step is to exchange all characters following the crossing site.

A′ and B′ are two new strings following this crossing :

A′ : 01011010
B′ : 10010101

String A′ is made up of the first part of strong A and the tail of string B. Likewise, string B′ is made up of the first part of string B and the tail of string A. Although crossover has a random element, it should not be thought of as a random walk through the search space.

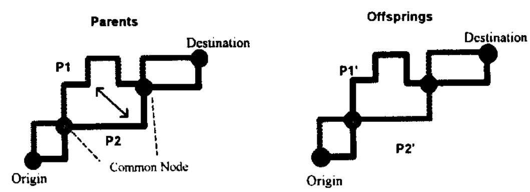
3) Mutation : Reproduction and crossover give GA's the majority of their search power. The third operator, mutation, enhances the ability of the GA to find near optimal solutions. Mutation is the occasional alteration of a value at a particular string position. It is an insurance policy against the permanent loss of any simple bit. A generation may be created that is void of a particular character at a given string position.

3.2 The Design of Genetic Operator

For successful application of GA, the design of the efficient genetic operator(crossover and mutation) appropriated to the problem is essential[7, 8]. We implemented the genetic operator through the mutual exchange of node information between the two path individuals and the change of sub-path contained in the path individual, similar to the evolution process of creatures.

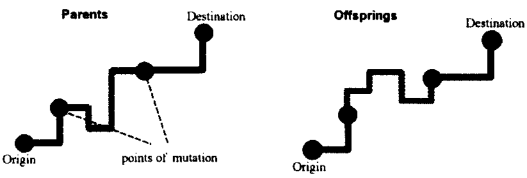
In the crossover operation, combining two individuals must not break the route. Crossover points in path individuals(P1 and P2) are determined at a common node. If two individuals have a common node, the latter segments are exchanged and two new children(P1′ and P2′) are created. If more than one node exist, segments between the common nodes are randomly selected. This process is shown in (Figure 6).

(Figure 7) represents mutation operation. A



(Figure 6) Path crossover operation

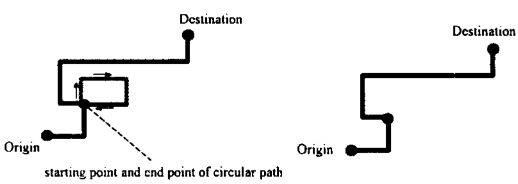
mutation operator randomly selects two points on a path. With these points as the origin and the destination, a new partial path is created and replaces the original segment.



(Figure 7) Path mutation operation

3.3 Creation Process of Path Individual

The creation process of path individual between origin and destination proposed in this paper is the process that the current node and the next node adjacent to it are acquired repeatedly, starting from originating node until terminal node is reached through probalistic random selection. In this process, next node is selected in the consideration of the relative distance to destination and turn constraints, in order to elevate search efficiency. Also circular sub-paths(composed of nodes more than three nodes) which can be contained in the process of creation are removed for improving search efficiency (see (Figure 8)). In doing so, only the circular sub-paths which are composed of nodes more than three nodes are removed so that U-turns are allowed.



(Figure 8) Execution example of circular sub-path deletion procedure

[Overall procedure of path creation]

- Stage 1 : Originating node(Origin) and terminal node(Destination) is designated.
- Stage 2 : Current node is established as a starting node.
- Stage 3 : Path extension process is executed. Neighboring nodes are searched and selected as next node, and then are saved in path array.
  If current node = starting node, one node among neighboring nodes of current node is selected as next node at random.
  Otherwise, one node among neighboring nodes of current node is selected as next node at random, in the consideration of turn constraints under relation current node to previous node. (nodes related to turn constraints are excluded in the course of selection)
- Stage 4 : If length of path array excesses the designated maximum length, go to stage 1.
- Stage 5 : Current node is established as previous node, and next node is established as current node (node information is updated).
- Stage 6 : if current node = terminal node (destination), the creation of path individual is terminated.
  Otherwise, go to stage 3.
- Stage 7 : The created path(individual) is investigated whether circular sub-

paths are existed or not. If circular sub-paths are existed, those sub-paths are removed.

3.4 Shortest Path Search Procedure

For elevating search efficiency, we changed the population size of path individuals in each generation, applying the concept of age and lifetime to path individuals. The overall procedure for searching the shortest path by the suggested genetic method is as follows :

- Stage 1 : Initializing parameters

The values of parameters such as initial population size, maximum population size, maximum number of generations, probabilities of path crossover and path mutation, etc. are established.

- Stage 2 : Creation of initial population

Using the proposed method of path creation, initial population of path individuals is created.

- Stage 3 : Creation of auxiliary population

A new auxiliary population of path individuals is created as many as the difference between current population size and maximum population size, using the proposed method of path creation.

- Stage 4 : Reproduction of path individuals

Using the proposed genetic operators - path crossover and path mutation, a new population of path individuals is reproduced with ease.

- Stage 5 : Process of elite

A few inferior individuals are exchanged with superior individuals, which are saved in a special memory location.

- Stage 6 : Evaluation of individuals and increment of age of each path individuals

Evaluating individuals, the lifetime of each path individual is assigned in accordance to the fitness value of each individual and the age of each path individual is increased by one. Therefore, the lifetime of superior individuals is longer than inferior ones.

- Stage 7 : Removal of old individuals

Old individuals which expire their lifetime are removed.

- Stage 8 : Investigating termination-condition

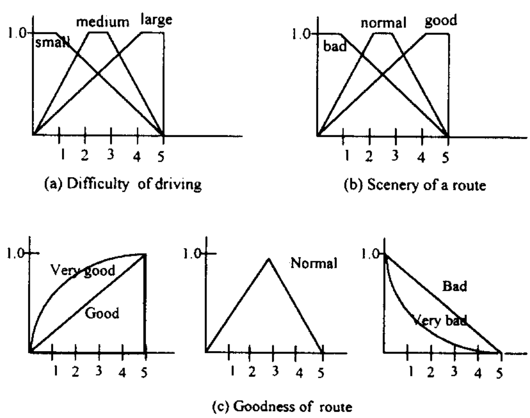
If the number of current generation is smaller than maximum number of generation, go to Step 3. Otherwise, the search process is terminated.

- Stage 9 : Output of the shortest route or multiple superior ones

4. Route Selection Based on Fuzzy Logic

Fuzzy logic was introduced by Dr. Lotfi Zadeh in the 1960s. It is a superset of Boolean (conventional) logic that has been extended to handle values between "true" and "false". As we know, Boolean logic has only two variables or truth-values, true and false. In contrast, fuzzy logic can deal with partial truth, that is, fuzziness, in concept. Fuzzy logic is very effective in dealing with tasks that involve uncertainties such as qualitative terms, linguistic vagueness, and human intervention. Fuzzy logic provides a simple way to draw definite conclusions from vague, imprecise information[5, 6].

There are assessment criteria in driver-preference route selection such as travel time, travel distance, difficulty of driving, and scenery of route, etc. The travel time is used to search the shortest routes by GA and the last two attributes to find the best route for a driver by fuzzy logic among the searched routes. Membership functions for input and output variables of the fuzzy system are shown in (Figure 9). For difficulty of driving, a value of 0 indicates ideal road environments, which is easy to drive, and 5 denotes the worst case. We assigned various values between 0 and 5 to all links of the virtual network. For scenery of route, 5 indicate the best scenery. The bigger the value is, the better it is. For outside road of virtual network, we assigned smaller values to difficulty of driving and bigger values to the scenery of a road than the others. For example, a value of 5 in difficulty of driving is assigned to a complex road section beside market area and a value of 0 in scenery of road to tunnel road or underground road.



(Figure 9) Membership functions for fuzzy system

A driver specifies his(her) preferences through user interface, selecting one of 5 levels : <very important>, <important>, <normal>, <not important>, <don't care>. By this, he(she) specifies the relative importance of the attributes. The value of each route attribute is multiplied by the predefined weight according to the preferences that he(she) selects and is passed to the fuzzy system.

For fuzzy inference, a type of fuzzy rules used is as follows :

IF difficulty of driving is <small> and scenery of route is <good>
THEN route is <very good>
IF difficulty of driving is <medium> and scenery of route is <good>
THEN route is <good>
. . . .

The entire fuzzy rule set for the proposed algorithm is fabricated in <Table 1>.

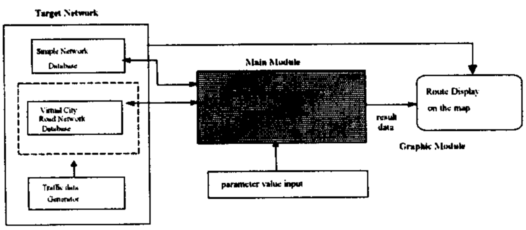
〈Table 1〉 Fuzzy rule set for the proposed algorithm

D \ S	Bad	Normal	Good
Big	Very Bad	Bad	Normal
Medium	Bad	Normal	Good
Small	Normal	Good	Very Good

S = Surrounding Scenery of Route
D = Difficulty Degree of Driving

5. Simulation and Results

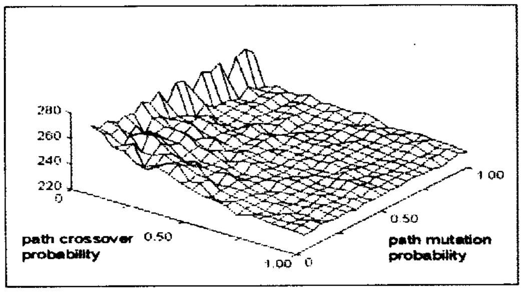
Computer simulation was carried to evaluate the performance of the proposed search algorithm(see (Figure 10)). We implemented a simple network and a virtual road network for simulation. Simple network is composed of the weighted digraph with 100 nodes, which don't have turn constraints. Link cost in the simple network is given to each link through a simple equation. It is simply used to analyze the effect according to the change of value of genetic parameters such as the probability of path crossover, probability of path crossover, initial and maximum population size, etc.



(Figure 10) The system structure of simulating the optimal route search

We observed the change of the travel cost of the same route when path crossover probability and path mutation probability were changed in the simple network. From this (as depicted in (Figure 11)), we ascertained that the performance of the proposed algorithm was most highly improved when pc is 0.9 and pm is 0.2 (pc : probability of path crossover, pm : probability of path mutation).

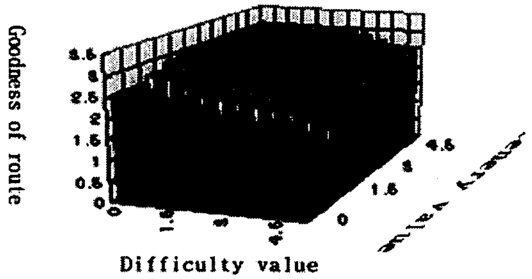
Virtual road network is composed of a weighted digraph with 100 nodes that has various turn constraints and 328 links, for assuming a typical city-road network. The data that were randomly generated between 800m and 1000m was assigned to each link in the virtual network as link distance.



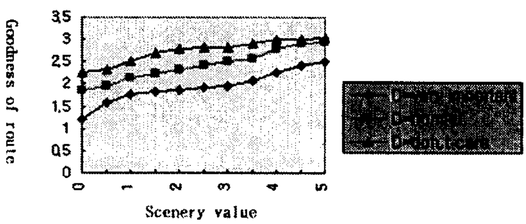
(Figure 11) Change of search performance by the change of path crossover probability and path mutation probability

Also, the speed data which are randomly generated between 10km/h and 90km/h through traffic generator are assigned to each link in the virtual network as link travel speed. Link cost becomes the value (link travel time)that divides link distance with link speed. The links indicated by thick line in the network represent the links which have heavy traffic volumes(link speed is below 30 km/h), assuming such as congestion, traffic accidents, road construction, etc. We assigned various values between 0 and 5 to all links of the virtual network. For scenery of route, 5 indicates the best scenery. The bigger the value is, the better it is. For outside road of virtual network, we assigned smaller values to difficulty of driving and bigger values to the scenery of a road than the others, assuming river-side road or city outer road.

(Figure 12) shows results of fuzzy inference where goodness of route is calculated with respect to the change of the values of the scenery of route and the change of the values of difficulty of driving respectively. It is shown in this figure that the smaller value of the difficulty or the bigger value of the scenery, the bigger goodness of route.



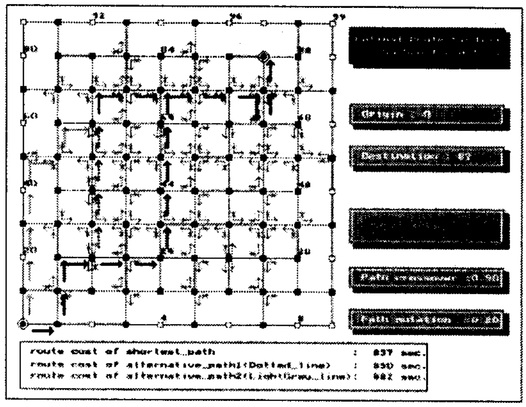
(Figure 12) The result of fuzzy inference by the change of fuzzy input



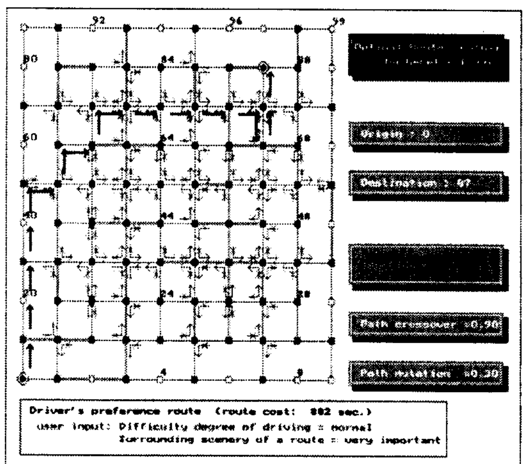
(Figure 13) The result of fuzzy inference by the change of the driver's preference in difficulty of driving.

(Figure 13) represents the results of fuzzy inference when driver's preference to difficulty and the value of scenery are varied. In here, the value of difficulty is fixed to 5 and driver's preference to scenery to <very important>. It is shown in this figure that the output values of fuzzy inference(goodness of route) are bigger, when driver's preference to difficulty is <don't care>. Because output values are almost not affected by the value of difficulty, even though difficulty degree of driving is the highest, when driver's preference to difficulty is <don't care>. From this result, we find that fuzzy inference is successfully worked.

(Figure 14(a)) shows high three routes in the rank, as the interim result of the proposed algorithm applying in the virtual road network. The route represented with thick line is the shortest one in the sense of travel time. (Figure 14(b)) shows best route for the driver, as the final result of the proposed algorithm. In here, origin is 0 and destination 87. And user preference for difficulty degree is normal and that for scenery very important. In this figure, it can be noticed that the best route for the driver



(a) Candidate routes searched by the proposed algorithm



(b) The best route for a driver searched by the proposed algorithm

(Figure 14) Result of driver's preference optimal route search by the proposed algorithm.

contains the outside road section(links) which has good scenery and is easy to drive.

5. Conclusion

As traffic pattern is complex and congestion in a metropolitan area is severe, dynamic route guidance systems become popular for increasing the efficiency of a road network. Most present route guidance systems are designed to let drivers follow the guidance provided by the system. But usually, each driver has his/her own driving preference. For efficient and intelligent route guidance, various attributes of a road network such as driving difficulty degree of driving, surrounding scenery of a road, etc. as well as traveling time should be considered. Future systems may be developed to allow users for having guidance choices or just simply to make their own decisions. Thus, we developed optimal route search algorithm, which reflects driver's preferences effectively, based on genetic-fuzzy approaches. The suggested algorithm can be effectively employed to complex road networks with diverse turn constrains, stochastic traffic volume, and various road attributes. From the results of simulation in the virtual road network, the suggested algorithm promptly produces the best route for a driver in consideration of his or her preferences. Therefore, the method can provide solutions to such traffic problems as road congestion, while satisfying diverse driver's needs. It is expected that this research will allow drivers to have guidance choices for their favorite routes in the future system.

References

[1] D.J.Jeffery, "Route Guidance and Vehicle Location Systems," Mobile Information Systems, pp.177-202, ARTECH HOUSE, 1990.

[2] G.S.Choi, S.Park, K.B.Woo, "Using the Paging Network to Develop the Advanced Traveler Information Systems," Proc. of The Second World Congress on Intelligent Transport Systems '95 YOKOHAMA, pp. 640-645, 1995.

[3] K. Takahashi, G. Pang, H. Takenaga, T. Yokota, "Drivers Route Seclection Based on Fuzzy-Neural Approaches," Proc. of The Second World Congress on Intelligent Transport Systems '95 YOKOHAMA, pp. 1854-1859, 1995.

[4] T.Kitamura, M.Kobayashi, K. Takeuchi, "The Dynamic Route Guidance Systems of UTMS," Proc. of The Second World Congress on Intelligent Transport Systems '95 YOKOHAMA, 1995.

[5] L.A. Zadeh, "Fuzzy Sets," Information and Control, Vol.8, No.3, 1965, pp.338-353.

[6] Y.Zhao, A.M.Kirson, L.G.Seymour, "Fuzzy-Logic Application in Vehicle Control and Navigation," Proc. of The Third World Congress on Intelligent Transport Systems '96 AMERICA, 1996.

[7] Z.Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Berlin :

Springer-Verlag, 1992.

[8] David E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Co. Inc., 1989.

[9] Yoon-Chul, Choi and Kwang-Sik, Kim, "Development of a Route Guidance System and its Application," Journal of Korea Transportation Research Society Vol.11, No.2, 1993.

[10] Timothy L. Nyerges, "Understanding the Scope of GIS : Its Relationship to Environmental Modeling," GEOGRAPHICAL INFORMATION SYSTEMS Vol.1, pp.76-92, Longman Scientific & Technical, 1991.

[11] James A. McHugh, *Algorithmic Graph Theory*, prentice-Hall International., Inc., 1990.

■ 저자소개

최 규 석

연세대학교 전기공학과를 졸업하고, 동 대학원에서 공학석사 및 공학박사 학위를 취득하였으며 (주) SK 텔레콤에서 이동통신시스템 및 신규서비스 관련 연구업무를 담당하였다. 현재 청운대학교 컴퓨터과학과 조교수로 재직하고 있으며 주요관심분야는 AI 및 Mobile Computing 분야이다.

박 종 진

1989년 연세대학교 전기공학과를 졸업하고, 동 대학원에서 공학석사 및 공학박사 학위를 1991년과 1997년에 각각 취득. 현재 청운대학교 인터넷컴퓨터학과에 교수로 재직하고 있으며 주요관심분야는 퍼지시스템, 지능제어 및 인터넷 컴퓨팅이다.