

# Adaptive Route Selection for Dynamic Route Guidance System Based on Fuzzy-Neural Approaches

Grantham K. H. Pang, *Member, IEEE*, Kazunori Takahashi, Takayoshi Yokota, *Member, IEEE*, and Hiroshi Takenaga

**Abstract**—One functionality of an in-vehicle navigation system is route planning. Given a set of origin–destination (O/D) pairs, there could be many possible routes for a driver. A useful routing system should have the capability to support the driver effectively in deciding on an optimum route to his preference. The objective of this work is to model the driver behavior in the area of route selection. In particular, the research focuses on an optimum route search function in a typical in-car navigation system or dynamic route guidance (DRG) system. In this work, we want to emphasize the need to orientate the route selection method on the driver's preference. Each feasible route has a set of attributes. A fuzzy-neural (FN) approach is used to represent the correlation of the attributes with the driver's route selection. A recommendation or route ranking can be provided to the driver. Based on a training of the FN net on the driver's choice, the route selection function can be made adaptive to the decision making of the driver.

## I. INTRODUCTION

WITH THE recent developments of advanced technologies, which includes communications, microelectronics, sensors, and information technology, the provision of real-time information on traffic conditions to drivers has become technically possible. For example, measurement of queue length and traffic flow in a traffic network can be obtained from inductive loop detectors or TV camera for surveillance. Information on the degree of congestion and estimated travel time can be collected by a traffic control center like Vehicle Information and Communication System Center (VICS), which started operation in Japan in 1996. VICS is an advanced information system designed to send real-time data on traffic jams, travel time, accidents, and road restrictions to in-vehicle units. The real-time road traffic data and information are gathered by the VICS Center and then distributed by radio wave and infrared beacons installed at every major road junction in the Tokyo and Osaka areas. The service area will gradually extend to the highways and other areas. Thus, the new navigation systems will be able to utilize the real-time traffic information by simply adding a receiver.

One functionality of an in-vehicle navigation system is route planning. Here, we represent a road network in the form of nodes (representing junctions) and a set of links (representing

roads). Given an origin–destination (O/D) pair, there could be many possible routes through the network. Generally, the cost functions are related to the links, which could be reflected by the travel time, distance, cost of travel, etc. The problem is understanding the complex evaluation process involved in the route choice and implementing a route selection function for the in-vehicle guidance system.

A route guidance system is a routing system that provides instructions to drivers based upon “optimum” route solutions. A driver can make the destination known to the system. The origin can be input or obtained directly from the use of a differential global positioning system (DGPS). A dynamic route guidance (DRG) system would route drivers using the current traffic conditions such as congestion and roadworks. The system can then provide actual routing advice based on real-time information regarding conditions and incidents of the traffic network.

One objective of such a dynamic route guidance system is to balance the level of service on all major network links so as to increase the efficiency, speed, safety and quality of travel (e.g., to minimize travel time). Such a system would be particularly useful when accidents or roadworks occurred in the traffic network. Also, the system is highly beneficial to the motorist when driving in unfamiliar areas. A DRG system would act as the driver's assistant and try to reduce his tension.

The issue of driver behavior in terms of route choice and response to guidance is complex. One focus in this paper is on the modeling of driver behavior in route choice on the DRG system. Here, we assume that the DRG system has a voluntary choice scheme and the driver has the option of not following the advice from the DRG system. The driver's preference is modeled as a fuzzy expert system, and his reaction to the advice and information provided by a DRG system is stored. The previous choices of the driver, in particular the deviation from the recommendation of the system, are then used for training so that the route selection function is made adaptive to the driver's preference.

### A. Literature Review

The study on route choice has been under the topic of traffic assignment. To solve the traffic assignment problem, the rule by which drivers choose routes between their origin and destination of travel must be defined. Van Vuren and Van Vliet [1] assumed that every driver wishes to minimize

Manuscript received May 3, 1995; revised April 18, 1997, October 21, 1997, and August 17, 1998.

G. K. H. Pang is with the Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong.

K. Takahashi, T. Yokota, and H. Takenaga are with the Hitachi Research Laboratory, Hitachi, Ltd., Ibaraki-ken 319-12, Japan.

Publisher Item Identifier S 0018-9545(99)07903-7.

his personal travel cost. They have also assumed that time minimization is the only criterion for the driver's route choice. A more comprehensive study on this topic is by Bovy and Stern [2]. They have identified a number of factors related to route selection and they fall into four categories: the characteristics/attributes of the feasible routes, the character of the traveler, the nature of that particular trip (e.g., purpose, budget), and other circumstances (e.g., weather, day/night). One study of route choice factors among truck drivers on motorways in Austria has come up with the following order of importance: travel time, width of the road, travel distance, route angularity, probability of delays, dangerous segments, slope of the road, multilanes, road safety, expected weather, and traffic density on the road.

As mentioned in [2], studies have revealed that people's preferences for various route characteristics do vary, and variables can be related to the characteristics of the travelers, their trips, and the routes to which they have been exposed. The difference seemed to be related to the great importance of direct/quick access to the destination in work trips, and importance of amenities (comfort/pleasant scenery) in leisurely trips. For example, scenery was found the most important noneconomic factor in shopping trips.

So far, the idea of "**optimum**" has been taken in a very limited sense. Most route guidance systems nowadays compute the "best" route for the driver based on either the shortest time [3], [4] or the shortest distance [5]. Some systems would provide information on congestion of the road as well [6]. Hence, the route selection function makes use of the distance data (static) and information on average travel speed (dynamic). In the VICS project announced by the National Police Agency in 1991, **optimum route selection** was a focus, but the focus was based only on travel distance and travel time reduction based on empirical and real-time information. Also, it was not clear how the two criteria can be resolved or compromised to give the optimum route. For example, in the driver's route selection logic in [7], it is not clear how to tradeoff the relative importance of minimum distance route and minimum time route.

From a survey carried out in 1989 [8], respondents were asked to choose between guidance systems that chose routes on the basis of shortest time, shortest distance or some combination of the two. In London, 42% of the respondents would prefer a system that can perform a tradeoff between travel time and travel distance. 56% said they would choose route based on the shortest travel time. A similar result is obtained in Paris. In Munich, 71% would prefer a tradeoff between travel time and travel distance, and only 27% would choose route based just on the shortest travel time. Therefore, it can be observed that a routing algorithm that can accommodate the various route selection criteria and their tradeoff would be highly desirable.

The use of fuzzy logic methodology in route selection was first proposed by Teodorovic and Kikuchi [9]. They have looked at the problem of route choice between two alternate routes. The driver's perceived travel time on each route is treated as a fuzzy number, and his choice of route is based on an approximate reasoning model and fuzzy inference. The model consists of rules which indicate the degree of preference for each route given the approximate travel time of the two routes. The approach considers only the travel time criterion and cannot be easily generalized to multiple routes.

Lotan and Koutsopoulos [10] have also proposed a modeling framework for route choice based on fuzzy set theory and approximate reasoning. The approach is based on the driver's perception of attributes of the network, attractiveness of alternate routes as well as models for reaction to information. An example of a fuzzy rule is given at the bottom of the page.

Such an approach works for a particular O/D set and does not seem general enough for different O/D pairs. Also, for an O/D pair, the inclusion of an additional feasible route means an entirely new set of fuzzy rules.

Teodorovic and Kalic [11] have considered route choice problem in air transportation using fuzzy logic. Other than travel time, the approach can handle additional route selection criteria such as travel costs, flight frequency, and the number of stopovers. However, the method works well when there are two possible routes (A and B) from the origin to the destination. An example of a fuzzy rule is given at the bottom of the next page.

The approach aims to explain the phenomenon of route choice when there are alternatives. Any extension such as having a third route would mean the development of an entirely different and carefully designed rule base.

### B. Decision Support in Route Selection

Advances in information technology allow the introduction of real-time data exchange and real-time control as new functions in road traffic. The information system must also support the driver effectively in decisions (i.e., information in the form of advice for proper action). The general information should be screened for decision making while driving, so that the system does not overload the driver with information.

In this paper, the focus will be on route selection, i.e., the capability to support the driver effectively in deciding on an optimum route to his preference. Fig. 1 describes such a navigation system. The core of such a system is an adaptive route selection algorithm based on a hybrid fuzzy-neural (FN) approach. Each feasible route has a set of attributes associated with it. The attributes are correlated and the final decision (choice) by the driver is perceived as a nonlinear function of the attributes.

This work is different from [9] as many more road attributes are treated in the route selection. Hence, route choice is not

---

IF	the perceived travel time on route 1 IS medium AND the perceived travel time on route 2 IS very high,
THEN	attractiveness of route 1 IS I will probably take route 1 AND I will definitely not take route 2

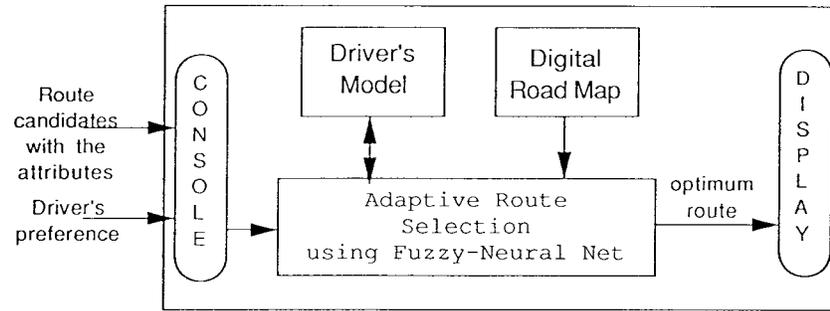


Fig. 1. A navigation system.

simply minimization of travel time. Also, this work does not need to assume that the driver is familiar with the network, streets and the traffic conditions. In addition, the approach can handle more than two feasible routes. When comparing with [10] and [11], the approach is more general and can apply to any O/D pair with any number of feasible routes.

This paper is outlined as follows. Section II gives a description on how a route selection system is setup based on some important attributes of a route. Section III presents a hybrid FN approach and the basic ideas behind such an approach. The details on the construction of an FN network is given in Section IV. Section V describes a modeling of the driver's route selection behavior using fuzzy rules, which is then used for determining the weights of the FN network. Section VI gives the details on the training of the FN network. An adaptive route selection algorithm is then given in Section VII, which is followed by an example in Section VIII. The paper is concluded in Section IX.

## II. SYSTEM DESCRIPTION

### A. System Setup

1) *Route Characteristics*: It is perceived that a driver may select a route based on many different factors which include:

- travel distance;
- travel time;
- degree of congestion (number of cars on the road, queue length);
- toll (of expressway or highway);
- degree of difficulty of travel (width of the road, number of lanes, and number of pedestrians and bicycles on the road, etc.);
- scenery (especially for long-distance trip).

2) *Route Attributes*: It is perceived that a feasible route has many different attributes. These attributes coincide with the factors which are used by the driver in route selection. Below is a set of some of the most important attributes of a feasible route. Note that each attribute has a range from zero (0) to one (1).

- *Travel Distance*: **1** denotes the route with the **shortest travel distance**, relative to the set of feasible routes. **0** can be used to denote routes which are  $x$  km longer than the shortest route, where  $x$  is a system parameter. The attribute value for other routes can be decided based on a linear scale.
- *Travel Time*: **1** denotes **shortest travel time**, relative to the set of feasible routes. **0** can be used to denote routes which are  $y$  minutes longer than the quickest route, where  $y$  is a system parameter. Again, the attribute value for other routes can be decided based on a linear scale.
- *Degree of Congestion*: **0** denotes **no congestion** at all. **1** denotes the worst situation.
- *Toll (of Expressway or Highway)*: **0** denotes **no toll and no highway** at all. **1** denotes the worst situation.
- *Difficulty of Travel (Narrowness and Winding of the Road, Number of Traffic Lights, Road Work, and Number of Pedestrians and Bicycles on the Road, etc.)*: **0** denotes the **ideal road situation**, very easy to drive. **1** denotes the worst situation.
- *Scenery (Especially on Long-Distance Trip)*: **1** denotes the **best scenery**. The higher, the better.

3) *Types of Attribute*: It can be noticed that for the attributes travel distance, travel time and scenery, a driver would like those attribute scores to be as close to one (i.e., as large) as possible for an ideal route. These three attributes are therefore called "positive attributes." As for attribute scores for congestion, toll and degree of difficulty, a driver would prefer them to be as close to zero (i.e., as small) as possible and they are called "negative attributes."

It can also be perceived that some attributes of a feasible route are dynamic while some can be considered as static. The dynamic ones are travel time, degree of congestion and degree of difficulty of travel. The static ones are travel distance, toll, and scenery.

4) *Driver's Dynamic Settings*: It is perceived that in some situations, a driver may have a particular preference. This may happen when planning for a special trip on a particular day. The following panel can be used as an interface for the driver

---

```

IF      there is a small negative difference in travel time AND
      there is a bigger frequency than on the alternate route B
THEN   there is a medium percentage of passengers using route A

```

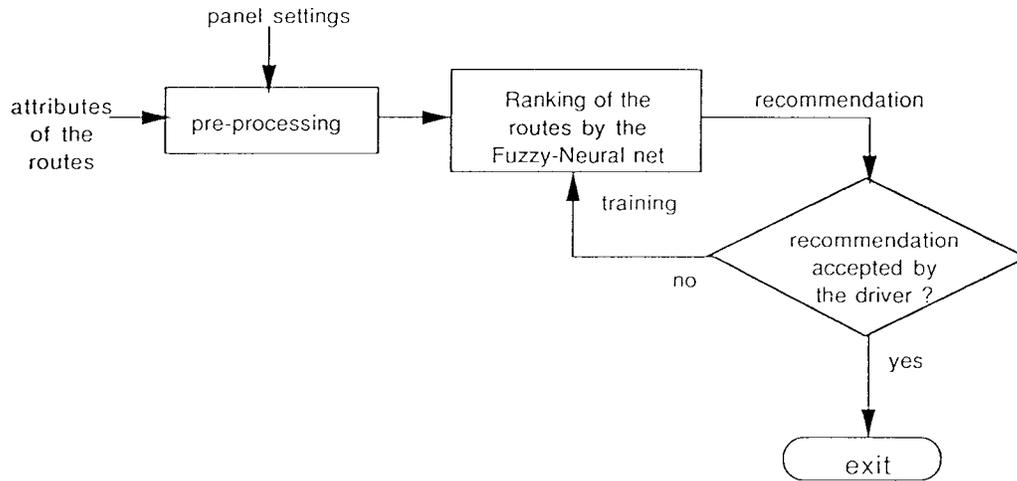


Fig. 2. Decision support to the driver.

to specify the relative importance of the route attributes given at the bottom of the page.

The introduction of these panel weights gives a quick and convenient means for a driver to specify his requirements to the routing algorithm. Effectively, the value of each route attribute is multiplied by its associated panel weight before passing to the route selection algorithm. For example, if a driver is very much concerned with avoiding congestion, and has the usual concern of arriving the destination by a quick route, the settings can be arranged as the panel shown above. In this way, the road attributes “toll,” “difficulty of travel,” and “scenery” will not be taken into consideration by the routing algorithm. On the other hand, the “degree of congestion” attribute should be given more weight than “travel time” and “travel distance.” The suggested values for  $p_i$  are as follows.

“Don’t care”	0.0.
“Not important”	0.4.
“Normal”	0.7.
“Important”	1.0.

5) *Decision Support (Fig. 2)*: It is perceived that at a particular instance of time, a number of different feasible routes which have different set of attributes should be considered by the driver. The driver has to make a decision based on the relative importance of the different factors for route selection. Each decision is based on a combination of different factors. There could be some heuristics in route selection, but some preferences could be difficult to express in words. The objective here is to design an **optimum route search function**

in an in-car navigation system so that it will have the following characteristics.

- It is a decision-making assistant to the driver in route selection. In other words, it embodies a route selection algorithm.
- It can model the behavior of the driver by storing his preference and previous decisions/choices.
- It can adapt and learn from the recent decisions of the driver.

### III. FUZZY-NEURAL APPROACH

#### A. Artificial Neural Net (ANN)

Neural networks [12] can be developed to model the driver behavior. It is chosen for this study for their ability to learn from examples, to generalize, to predict and to cope with incomplete input data. A neural network is a parallel distributed information processing system. It consists of a large number of highly interconnected, but very simple processing elements known as neurons. Each neuron has a number of inputs and one output which branches out to inputs of other neurons. The output of a neuron is a nonlinear function of the sum of all inputs through the weighted links. Hence, the knowledge of a network is distributed throughout the weighted links.

For our application, the inputs will be the various attributes of a route and the output will be an acceptance measure of the route. The ANN can be trained off line. The real-time execution of the ANN will be extremely fast. It also has the

---

Don't care	Not important	Normal	Important	
I-----I-----I---X-----I-----I				min. travel distance
I-----I-----I---X-----I-----I				min. travel time
I-----I-----I-----I-----X-----I				avoid congestion
I-----X-----I-----I-----I-----I				avoid highway/toll
I-----X-----I-----I-----I-----I				avoid difficult road
I-----X-----I-----I-----I-----I				good scenery

ability to adapt to different users of the car. Any new user can train the network to learn his preference.

### B. Fuzzy Systems

A driver's choice of a route is normally based on a complex evaluation process in which the attributes of all the feasible routes are weighted subjectively. The utilities of individual route attributes are also measured subjectively. For example, despite the fact that estimated travel time is a measurable parameter, when a driver makes the route choice, his notion of travel time is often fuzzy. Also, he would tradeoff the different route attributes involved and makes his judgement. The modeling of such a decision-making process of a driver is complex and it is believed that fuzzy logic and approximate reasoning model can help to understand the process. The aim of this paper is to develop a fuzzy expert system [13] which can be used to rank the feasible routes. At the beginning of the system, heuristic rules on how the driver evaluates the route attributes can be specified.

### C. Advantage of Fuzzy Systems

Rule-based fuzzy systems are based on fuzzy theory, with expert knowledge represented explicitly using a set of fuzzy if-then rules. They offer a high degree of transparency into the system being modeled.

### D. Advantage of Neural Systems

Neural networks have very strong learning capability. Given some numerical data on how a system should behave, there are many neural network algorithms which try to learn this system behavior. Knowledge is stored implicitly in the weights of a neural network. However, the neural network offers no insight to internal dynamics and relationships.

### E. A Hybrid Approach

A hybrid FN approach can combine the advantages of both approaches. This will further enhance the intelligence of the DRG system, especially in the modeling of the driver behavior. The ideas are as follows.

- 1) A rule-based fuzzy system is developed which represents a preliminary model of the driver.
- 2) The rule-based fuzzy system is then implemented using a neural network. A method of constructing a neural network which is equivalent to the fuzzy system is developed. It is constructed so that the procedures and membership functions of the fuzzy system can be retrieved from the implementation of the neural network.
- 3) A special learning algorithm is then used to learn and adapt itself to the recent choices of the driver. The weights of the network will be adjusted. The derivation of the learning algorithm is based on a gradient descent algorithm.
- 4) After the training procedure, the modified membership functions of the fuzzy systems can be retrieved. This fuzzy system with modified membership functions represent the latest model of the driver. The fuzzy system

has now been tailored to the particular driver. The model of the driver can also be represented by a set of weights of the equivalent neural network.

## IV. CONSTRUCTION OF FN NETWORK

In this section, the architecture of a kind of FN network is described. The network is essentially a parallel implementation of a fuzzy system using a particularly structured neural network. The structure involves the construction of a fuzzification subnetwork and a defuzzification subnetwork. The two subnetworks will be integrated in such a way that the structure and decision-making process of the original fuzzy system can be fully retrieved from its network implementation. The corresponding neural network should have similar performance as the original fuzzy system.

Fig. 3 shows the architecture of the entire FN network. Each subnetwork performs a different function explained as follows.

- *Fuzzification Subnetwork*: To represent the membership functions of the linguistic terms of the input attributes.
- *Defuzzification Subnetwork*: To generate a defuzzified value which represents the acceptance level of a feasible route.

Here, a summary of the notations used in the discussion is given.

$x_i$	Inputs to the fuzzy system or the fuzzification subnetwork, in the range $[0, 1]$ .
$\alpha_i$	Outputs from the fuzzification subnetwork or inputs to the defuzzification subnetwork.
$w_{ij}$	Weights in the defuzzification subnetwork.
$u$	Defuzzified output from the fuzzy system or output from the defuzzification subnetwork.
$\beta_i$	Weights in the second layer of the defuzzification subnetwork.
$p_i$	Panel weights in the range $[0, 1]$ .
$\mu_i$	Intermediate quantity in the defuzzification subnetwork.
$\gamma$	Sum of all the $\mu_i$ .
$r$	Number of fuzzy rules.
$m$	Number of route attributes.
$n$	Number of discrete values in the output range $u$ .
$q$	Number of feasible routes when given an O/D pair.
$d$	Datum value used in the defuzzification procedure.

### A. Fuzzification Subnetwork

The inputs of the system go to the first layer which is consisted of a number of fuzzification subnetworks. The input  $x_i$  to each fuzzification subnetwork is the same as the input to the fuzzy logic system. The outputs from the fuzzification subnetwork correspond to the degree of membership  $\mu_{ik}$  where  $k$  is the number of linguistic terms corresponding to the  $x_i$ . Fig. 3 shows some two-layer fuzzification subnetworks for this purpose. It should be noted that the number of hidden layers in a fuzzification subnetwork is not important as long as it can represent the membership function of the linguistic term of the input attribute. Initially, each subnetwork should be trained with the specified membership functions of  $x_i$ . If

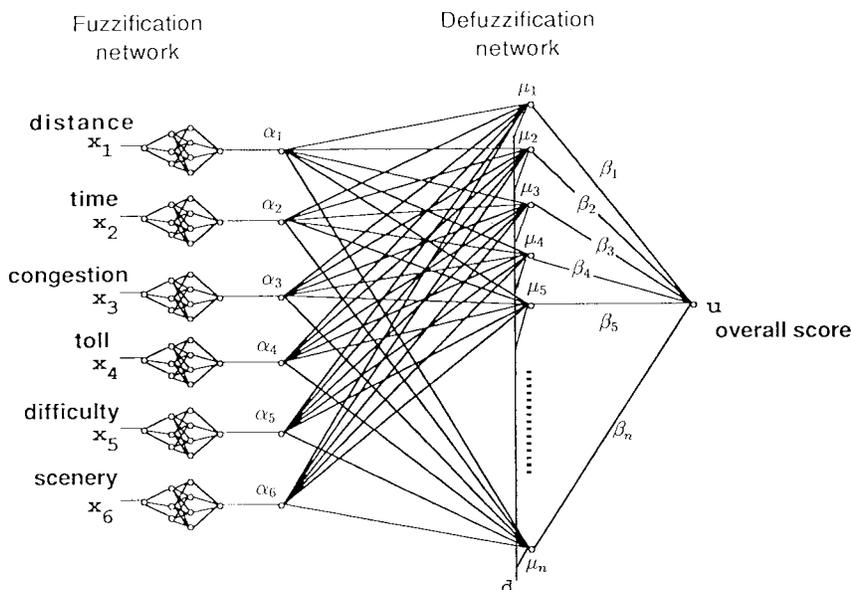


Fig. 3. Architecture of the FN network.

two inputs have the identical set of linguistic terms, then the same fuzzification subnetwork can be used at the beginning. During the subnetwork learning stage described later on, the membership functions of the linguistic terms will be changed subsequently using the training data.

**B. Defuzzification Subnetwork**

This is the most important stage in the construction of the FN network. The aim is to compute the acceptance value of a feasible route. The main issue concerns with ways to represent membership functions of output linguistic terms using a network implementation and to perform defuzzification. The defuzzification subnetwork consists of two layers. The first layer is introduced to represent the membership functions of output linguistic terms. The number of inputs  $\alpha_i$  to this layer is the same as the number of fuzzy rules  $r$ . Let  $n$  be the number of discrete values in the output range  $u$ . Let  $U = \{u_1, u_2, \dots, u_n\}$  be a set of discrete output values. The initial weights of this layer are calculated from membership functions of the output as

$$w_{ij} = \mu_{C_j}(u_i) \tag{1}$$

where  $i = 1, \dots, n$  and  $j = 1, \dots, r$ . Note that  $C_j$  is the linguistic label of rule  $j$ . Essentially, it means that the weight  $w_{ij}$  equals the degree of membership of  $C_j$  at  $u = u_i$ . The second layer in this subnetwork performs the task of defuzzification. Here, a center-of-area method is used. In this case, the weights of the second layer is given by

$$\beta_i = \frac{u_i}{\gamma} \tag{2}$$

where

$$\gamma = \sum_{i=1}^n \mu_i. \tag{3}$$

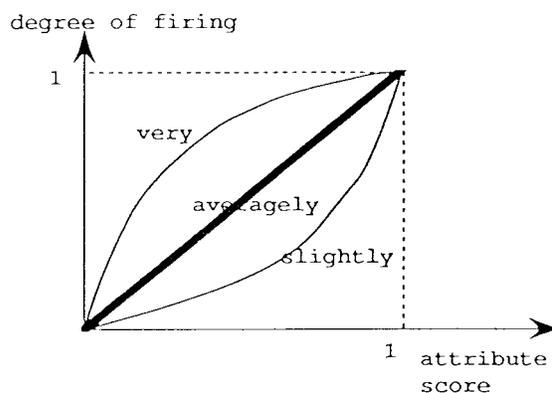


Fig. 4. The definition for the terms very, averagely, and slightly.

The defuzzified value  $u$  is given by

$$u = \sum_{i=1}^n \beta_i \mu_i \tag{4}$$

whereas

$$\mu_i = \sum_{j=1}^m w_{ij} \cdot \alpha_j + d. \tag{5}$$

Again, the initial weights in the first layer of the defuzzification subnetwork can be changed later by the learning algorithm and the training data. However, the learning process will only change the membership functions of the output linguistic terms in the first layer and not the defuzzification operations in the second layer.

**V. MODELING OF DRIVER BEHAVIOR**

**A. Fuzzy Ranking Rules**

Each driver would have his own perspective of a desirable route. The system is designed so that a driver can specify his

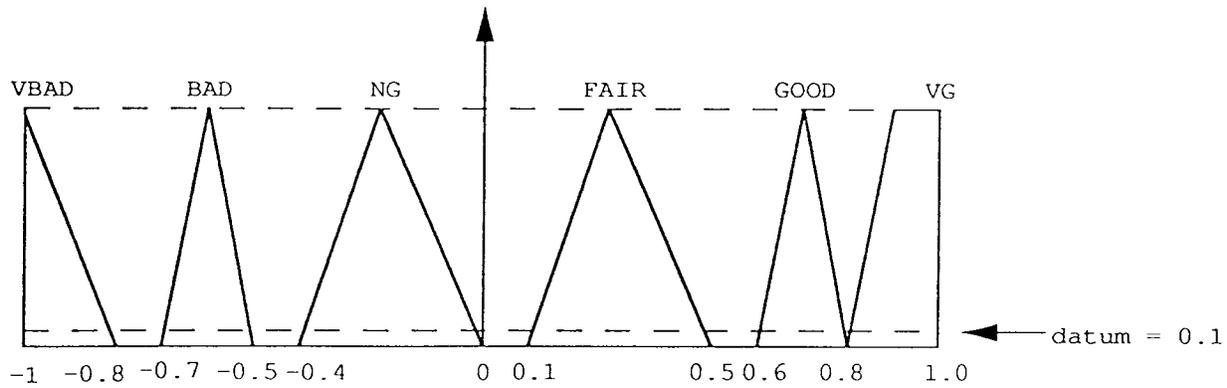


Fig. 5. Definition for the membership functions: VBAD—very bad; BAD—bad; NG—not good; FAIR—fair; GOOD—good; and VG—very good.

Rule 1 : If <travel distance> is <averagely> short,  
 then <route> is <good>.

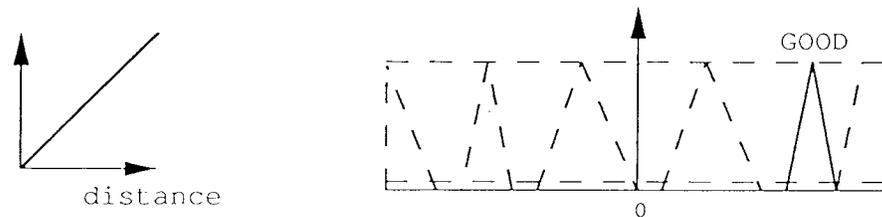


Fig. 6. First fuzzy rule used for modeling of the driver’s route selection.

preference using fuzzy rules with some predefined linguistic terms. Below is an example of a set of fuzzy rules that may be used by a typical driver for route ranking.

- 1) If **travel distance** is *<averagely>* short, then route is *<good>*.
- 2) If **travel time** is *<averagely>* short, then route is *<very good>*.
- 3) If **congestion** is *<averagely>* heavy, then route is *<very bad>*.
- 4) If **toll** is *<averagely>* high, then route is *<not good>*.
- 5) If **degree of difficulty** is *<averagely>* high, then route is *<bad>*.
- 6) If **scenery** is *<averagely>* good, then route is *<fair>*.

The linguistic terms are in brackets. In the rule antecedent part, the driver can choose from the term set (very, averagely, and slightly). The definition for them can be seen in Fig. 4. The definition of the membership function of *good*, *very good*, *very bad*, *not good*, *bad*, and *fair* is shown in Fig. 5. The first fuzzy rule defined above is represented as in Fig. 6.

*B. Linguistic Terms and Rule Firing*

An example of definition of the membership function for the output linguistic terms is shown in Fig. 5. The numeric value of a route attribute is the crisp input to the associated fuzzy rule. The fuzzification subnetwork will compute the degree of match between the crisp input and the fuzzy set describing the meaning of the rule antecedent. The fuzzy set describing the meaning of the rule antecedent is scaled to the same degree to which the rule antecedent has been matched by the crisp input.

Finally, the scaled fuzzy sets of each rule are aggregated to obtain the defuzzified value, which is a performance measure of the feasible route.

In graphical representation, the firing of each fuzzy rule will contribute an area to the defuzzification procedure. The algorithm adopted in this paper is the center-of-area (COA) method, which is the best well-known defuzzification method.

A datum value *d* is introduced in the defuzzification procedure and the effect is analogous to having a uniform rectangular area over the output range. This is an area of fixed size and it is always taken into account in the defuzzification procedure. This area is important because if the driver is interested in ranking routes based on just a single road attribute, the defuzzified value from the route algorithm will give the desired ranking.

An example is given below. Suppose the driver would like to rank the routes based only on the road attribute “travel distance.” Therefore, the panel settings of all the other attributes should be placed to the far left-hand side (i.e., Don’t Care). Suppose the “travel distance” attribute values of the two feasible routes are 0.3 and 0.7. Also, let the datum value be 0.1. The graphical representation of the defuzzification method in these two cases are given by the two figures in Fig. 7. In each case, the defuzzification procedure has to handle two areas and the defuzzified value gives the correct ranking. If the datum value is not used, the defuzzified value would be the same as the defuzzification procedure would not be able to distinguish the two cases as the center-of-area would be the same. This datum value in the defuzzification algorithm can also be viewed as a scaling factor.

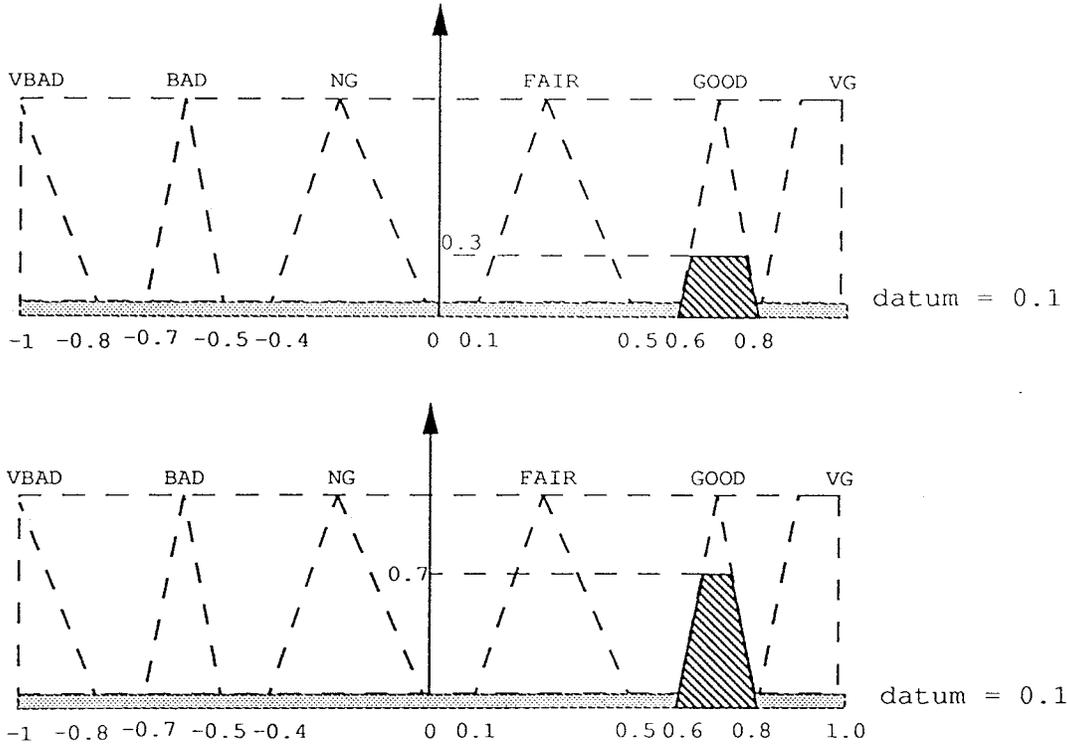


Fig. 7. Defuzzification with the datum (area) would give the correct ranking.

### VI. TRAINING OF THE FN NETWORK

In this section, an algorithm to train the FN network is described. The architecture of the FN network for adaptive route selection is shown in Fig. 3. The derivation of the equations is in the Appendix. The objective is to train the neural network so that it will refine the membership functions of the fuzzy systems. A set of learning rules similar to the backpropagation learning algorithm can be derived from the gradient descent method.

The back-propagation learning algorithm developed for standard multilayer feedforward neural networks can be extended for the training of the FN network. Here, we assume that a set of desired or optimal input/output pairs is available. Let  $u$  be the output of the network and  $u^d$  be the desired value. The error function is defined as

$$E = \frac{1}{2}(u^d - u)^2. \quad (6)$$

During the training process, the defuzzification algorithm needs to be maintained, and, hence, the  $\beta_i$  are calculated from (2) using  $\mu_i$ . The rule for updating  $w_{ij}$  in the first layer of the defuzzification subnetwork can be shown as

$$w_{ij}(t+1) = w_{ij}(t) + \epsilon \alpha_j (u^d - u) \left( \beta_i - \sum_{l=1}^n u_l \mu_l / \gamma^2 \right) \quad (7)$$

where  $\epsilon$  is a learning rate between 0 and 1.  $\beta_i$  and  $\gamma$  are defined as in (2) and (3) respectively.  $\alpha_j$  represents the degree of firing of fuzzy rule  $j$  in the fuzzification subnetwork. Note that  $w_{ij}(t+1)$  is updated only if  $\mu_i$  is within the range [0, 1]. The error term  $e_j$  that is back-propagated to the output of

the fuzzification subnetwork is

$$e_j = (u^d - u) \sum_{i=1}^n w_{ij} \left( \beta_i - \sum_{l=1}^n u_l \mu_l / \gamma^2 \right). \quad (8)$$

This error term is used to adjust the weights of the fuzzification subnetwork using the back-propagation algorithm.

When the training process has been completed, the new membership functions can be recovered from individual subnetwork. From the fuzzification subnetwork, we can obtain the modified membership functions for each linguistic label of input  $x_i$ . To obtain the modified membership functions for the output linguistic labels, we have to set  $\alpha_i$  to one (where  $i = 1, \dots, r$ ), all others to zero and obtain  $\mu_i$  (where  $i = 1, \dots, n$ ).

### VII. ADAPTIVE ROUTE SELECTION ALGORITHM OF THE DRG SYSTEM

In this section, a route ranking/selection algorithm is presented. An important feature of this algorithm is on the training of the FN network. With this, the system is made adaptive to the preference of the driver. This preference is assumed to be contained in the recent selection cases made by the driver.

The ARS algorithm is given as follows.

- 1) *Route Attribute*: Let  $X = \{x_i\}$  be a set of attributes of a route. Also, let  $m$  be the number of route attributes. Note that each attribute is normalized to between 0 and 1. For example
  - $x_1$  travel distance (positive attribute; the higher, the better);

- $x_2$  travel time (positive attribute; the higher, the better);
- $x_3$  congestion (negative attribute; the smaller, the better);
- $x_4$  toll charge (negative attribute; the smaller, the better);
- $x_5$  difficulty of travel (negative attribute; the smaller, the better);
- $x_6$  scenery (positive attribute; the higher, the better).

It should be noted that the values of this set of attribute have already been modified by the panel settings specified by the driver. If the original values of the route attributes are  $\{\tilde{x}_i\}$  and the panel weights are  $\{p_i\}$ , then

$$x_i = p_i * \tilde{x}_i.$$

- 2) *Fuzzy Rules*: For each of the relevant route attributes  $x_i$ , determine the associated fuzzy rule using the given set of linguistic terms. Let  $R$  be this set of fuzzy rules.
- 3) *Fuzzy-Neural Network*: Construct an FN network based on  $R$ . Note that all weights in the FN network are determined by the linguistic terms of the fuzzy rules. Let  $W$  be the set of weights in the network. Set the learning rates of the FN network. A heuristic value to use is 0.6 for the defuzzification subnetwork and 0.1 for the fuzzification subnetwork.
- 4) *Datum*: Set the datum  $d$  of the FN network. This value should be fixed for the rest of the system operation. A heuristic value to use is 0.1.
- 5) *Feasible Routes*: For any given O/D pair specified by the driver, let  $S$  be the set of feasible routes. Let  $q$  be the number of feasible routes. Hence,  $S = \{s_1, s_2, \dots, s_q\}$ .
- 6) *Route Score*: For each feasible route, let  $X$  be the inputs to the FN network and  $u$  be the output of the network, which is a score of the feasible route. The FN network can be viewed as a nonlinear mapping  $f$  from  $X$  to  $u$ . That is

$$u_j = f(x_1^j, x_2^j, x_3^j, x_4^j, x_5^j, x_6^j) = f(\bar{x}^j)$$

where  $j = 1, \dots, q$ .

- 7) *Route Ranking*: Given  $S$ , we obtain  $U = \{u_j\}$  ( $j = 1, \dots, q$ ). A feasible route in  $S$  can be ranked according to its score. This route ranking function is an important decision-support assistant for the driver. The ranking is carried out in accordance with the preference of the driver specified by  $R$ .
- 8) *Adaptive Route Selection*: Let  $s_c$  be the chosen route by the driver (i.e., the  $c$ th route is selected). Also, let  $a$  be the score of the chosen route, and  $b$  be the highest score among  $S$ . That is

$$\begin{aligned} a &= f(\bar{x}^c) \\ b &= \max\{u_i\}. \end{aligned}$$

Let  $s_k$  be the route with the highest score. Hence

$$b = f(\bar{x}^k).$$

If  $a = b$ , the driver has accepted the recommendation from the route selection function. The procedure can be repeated with another O/D pair by going back to step 5). If  $a \neq b$ , the selection of the driver is different from the recommendation and the driver has selected a different route for travel. In this case, two training pairs are formed

$$\begin{aligned} &(\bar{x}^c, b) \\ &(\bar{x}^k, a). \end{aligned}$$

The reasoning is as follows. The route that is recommended by the system is the one which has the highest score among all the feasible routes. If this route is not accepted by the driver, the score of this route should have been lowered. On the other hand, the score of the chosen route should have been higher to reflect the preference of the driver. Of course, it is not clear how much the scores should have been changed. Thus one can argue that the chosen route should have been given the highest score  $b$ , whereas the declined route can be given a score equal to  $a$ , which is lowered than before. This is reasonable as the interchange will also keep the numeric values of the FN network to scale.

- 9) *Training the FN Network*: The two training pairs that are formed are used as input-output training pairs of the FN network. Note that each case of deviation from the recommendation will form two training pairs. On the frequency of training the FN network, it can be carried out after every  $z$  case(s), where  $z$  can be set at the beginning of the system operation.
- 10) *New Weights*: Store  $W$  in a data file associated with the driver. Go back to step 5).

To summarize, the route selection procedures are as follows.

- 1) The system is setup by a driver inputting his fuzzy rules on route selection using a set of predefined linguistic terms. This is the initial setup of the system and an FN network is formed. The weights of the FN network are determined from the shape of the linguistic terms.
- 2) When planning a trip or during a trip, the driver can modify the relative importance of the various route attributes using some settings on a panel. This is a convenient way for specifying driver's preference, which could be useful for planning a special-purpose trip.
- 3) The driver inputs his origin and destination to the system, and a set of feasible routes is obtained.
- 4) For each feasible route, the attribute scores are inputs to the FN network, and the output is an overall score of that feasible route. With the computation of this overall score, a ranking of the set of feasible routes is performed.
- 5) The driver can accept the recommendation from the system. Alternatively, he can choose an alternate route. Any derivation from the recommendation will be stored, and this information is used for forming the training

pairs of the FN network. Hence, the system can be made adaptive to the decision-making of the driver.

### VIII. APPLICATION EXAMPLE OF THE FN NETWORK

#### A. Example on the Adaptive Feature of the FN Network

An example on the use of the FN network is given in this section. Suppose the number of feasible routes is five. The FN network would give a score to each feasible route, and a ranking is resulted. In order to show the adaptive ability of the network, suppose we have four cases of deviation from the recommendation of the FN network. In each case, the routes are ranked according to the fuzzy rules previously setup by the driver. However, in this example, the driver has deliberately changed his route choice and selected the route with the best scenery in all the four cases.

For example, in the first case, five feasible routes are presented to the driver. The route attributes are given at the bottom of the page.

Using the same notation as in the ARS algorithm

- $m = 6$  (i.e., the number of road attributes)
- $d = 0.1$  (i.e., the datum of the FN network)
- $q = 5$  (i.e., the number of feasible routes)
- $\bar{x}^1 = [0.6, 1.0, 0.3, 0.6, 0.3, 0.3]$
- $\bar{x}^2 = [1.0, 0.8, 0.5, 0.4, 0.2, 0.5]$
- $\bar{x}^3 = [0.85, 0.5, 0.2, 0.2, 0.3, 0.5]$
- $\bar{x}^4 = [0.45, 0.3, 0.4, 0.3, 0.1, 0.9]$
- $\bar{x}^5 = [0.4, 0.25, 0.45, 0.35, 0.15, 0.85]$ .

Note that Route 1 is the route with the shortest travel time. Route 2 is the route with the shortest travel distance. Route 4 is the route with the best scenery. Route 5 is the route with the second best scenery. From the set of fuzzy rules specified before, the score for each feasible route is obtained from the FN network. The scores are as follows:

- $y_1 = f(\bar{x}^1) = 0.207638$
- $y_2 = f(\bar{x}^2) = 0.201110$
- $y_3 = f(\bar{x}^3) = 0.215649$
- $y_4 = f(\bar{x}^4) = 0.115851$
- $y_5 = f(\bar{x}^5) = 0.072935$ .

Note that Route 3 has the highest score. Here, suppose the driver would like the route with the best scenery, i.e., Route 4.

Hence

- $k = 3$   
(i.e., route  $k$  has the highest score from the FN network)
- $c = 4$  (i.e., route  $c$  is selected by the driver)
- $a = 0.115851$
- $b = 0.215649$ .

Two training pairs are formed

$$\begin{aligned} &(\bar{x}^c, b) \\ &(\bar{x}^k, a) \end{aligned}$$

which are

$$\begin{aligned} &(\bar{x}^4, 0.215649) \\ &(\bar{x}^3, 0.115851) \end{aligned}$$

or

$$\begin{aligned} &([0.45, 0.3, 0.4, 0.3, 0.1, 0.9], 0.215649) \\ &([0.85, 0.5, 0.2, 0.2, 0.3, 0.5], 0.115851). \end{aligned}$$

#### B. Training of the FN Network

The following input–output pairs are used for training the FN Network. They are collected from the four deviation cases made by the driver

$$\begin{aligned} &([0.45 \ 0.3 \ 0.4 \ 0.3 \ 0.1 \ 0.9], \ 0.215649) \\ &([0.85 \ 0.5 \ 0.2 \ 0.2 \ 0.3 \ 0.5], \ 0.115851) \\ &([0.5 \ 0.8 \ 0.6 \ 0.2 \ 0.3 \ 0.7], \ 0.214050) \\ &([0.7 \ 1.0 \ 0.2 \ 0.4 \ 0.7 \ 0.3], \ 0.163516) \\ &([0.7 \ 0.8 \ 0.5 \ 0.2 \ 0.3 \ 0.8], \ 0.258547) \\ &([1.0 \ 0.8 \ 0.3 \ 0.3 \ 0.2 \ 0.6], \ 0.201303) \\ &([0.7 \ 1.0 \ 0.4 \ 0.7 \ 0.7 \ 0.7], \ 0.213644) \\ &([1.0 \ 0.85 \ 0.2 \ 0.6 \ 0.5 \ 0.4], \ 0.157301). \end{aligned}$$

#### C. Testing of the FN Network

For the case 1 example described in Section VIII-A1, the attributes of the five feasible routes are provided for the newly adapted FN network. New scores are obtained and given at the bottom of the next page.

It is noticed that the route with the best scenery is now given the highest score, followed by the route with the second best scenery. This is what would be expected from an adapted system. That is, if the same route choice situation arises in the future, the route selection function would offer the desired ranking of the feasible routes.

---

	distance	time	congestion	toll	difficulty	scenery
Route 1:	0.6	1.0	0.3	0.6	0.3	0.3
Route 2:	1.0	0.8	0.5	0.4	0.2	0.5
Route 3:	0.85	0.5	0.2	0.2	0.3	0.5
Route 4:	0.45	0.3	0.4	0.3	0.1	0.9
Route 5:	0.4	0.25	0.45	0.35	0.15	0.85

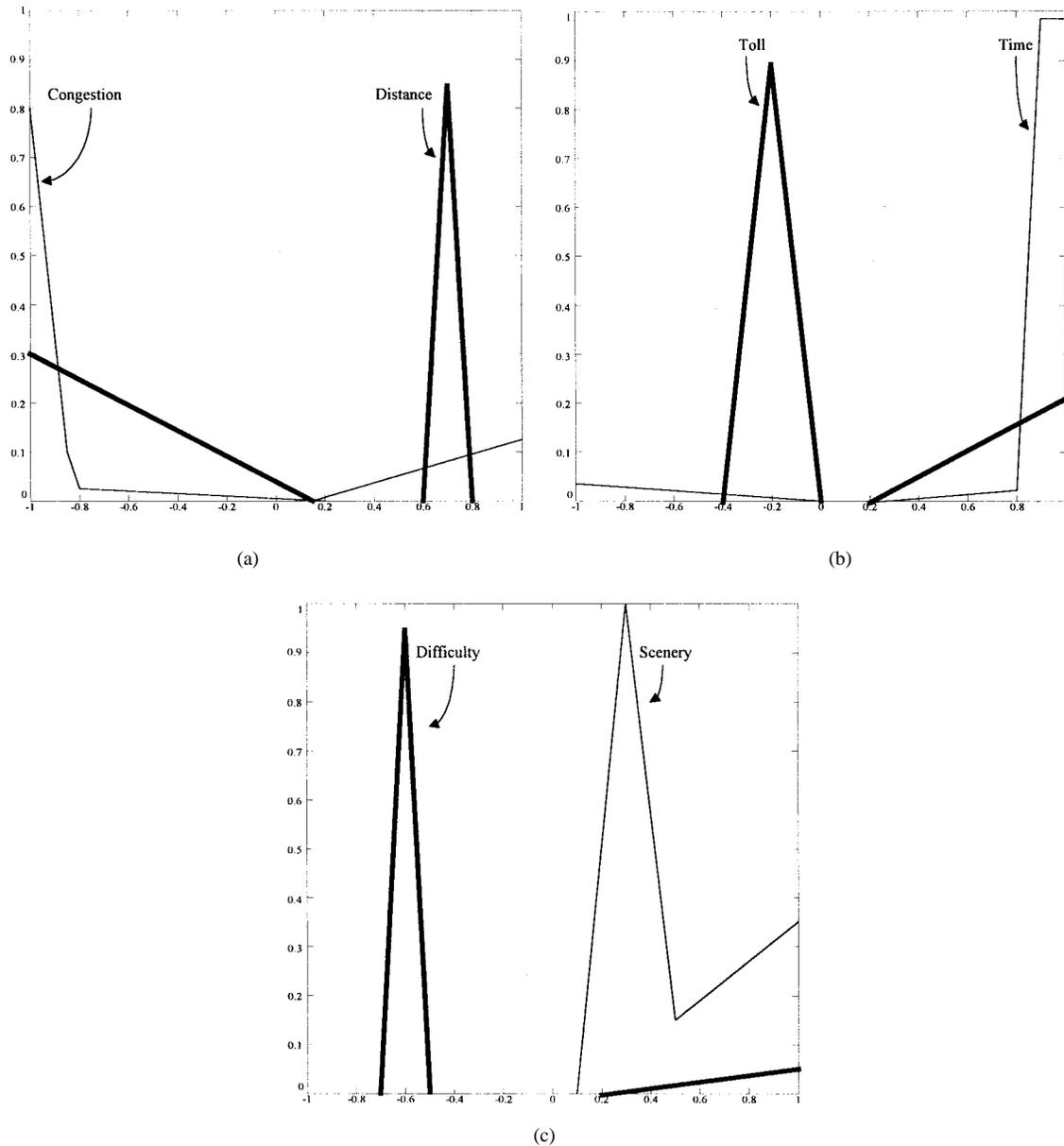


Fig. 8. Modified membership function of the route attributes.

*D. Modified Membership Function*

The modified membership function of the route attributes is shown in Fig. 8. It can be noticed that the peaks of the membership functions such as distance, congestion, time, difficulty and toll have all been lowered. In addition, some “counterweights” on those membership functions can be observed. Those two observations suggest that the FN network

is trying to reduce the importance of those five route attributes.

On the other hand, the membership function of the “scenery” attribute has been strengthened as we can observe additional weights being added on the positive scale.

From the modification to the membership functions of the FN network, we can observe that the route selection system

		new score
Route 1: old score = 0.207 638		--- 0.187 520
Route 2: old score = 0.201 110		--- 0.169 031
Route 3: old score = 0.215 649 (best score)		--- 0.127 172
Route 4: old score = 0.115 851 (best scenery)		— 0.206 356
Route 5: old score = 0.072 935 (second best scenery)		--- 0.197 862

is trying to be adaptive to the recent changes in the driver's selection of routes. It has correctly identified that the driver is giving more preference to routes with the best scenery. After the training of FN network with the data from four cases of those choices, the FN network has adapted itself and is giving more preference to routes with good scenery.

### E. Implementation

The FN network described above and the adaptive route selection algorithm has been implemented using the C programming language. This work will form an important component of an intelligent route navigation system.

## IX. CONCLUSION

A useful routing system should have the capability to support the driver effectively in deciding on an optimum route to his preference. In this paper, an **optimum route search function** in a typical in-car navigation system is developed. The main contributions of this paper are as follows.

- A FN approach is used for the modeling of the decision-making process in route choice of a driver.
- The approach is general in the sense that the fuzzy rules can handle any number of feasible routes when given an O/D pair.
- The approach can also handle multiple route selection criteria (travel time, travel distance, congestion, scenery, etc.).
- Six primary criteria for route selection are defined in this paper. They serve as the fundamental factors in an ARS algorithm.
- The most important feature of the ARS algorithm is its adaptive nature. The developed FN network can be trained (made adaptive) with the information on the driver's deviation from some previous recommendations of the FN network. An example is included in this paper to show this feature.

To summarize, the route selection algorithm is orientated on the driver's preference. An **FN approach** is used to represent the correlation of the attributes with the driver's route selection. A recommendation or route ranking can be provided to the driver. Based on a training of the FN net on the driver's choice, the route selection function can be made adaptive to the decision-making of the driver. This methodology paves the way for more intelligent navigation systems.

### APPENDIX I DEFUZZIFICATION ALGORITHM

In this section, the equation of the defuzzification algorithm is derived.

Let  $U = \{u_1, u_2, \dots, u_i, \dots, u_n\}$  be a set of discrete values within the range of output in a fuzzy system. Also, let  $\mu_{C_j}$  be the membership function of the output linguistic term  $C_j$  defined for rule  $R_j$ . Now, suppose the degree of firing rule  $R_j$  be  $\alpha_j$ .

With the notation defined above,  $\mu_{C_j}(u_i)$  is the degree of membership of  $C_j$  at  $u = u_i$ . In the defuzzification

subnetwork, this is denoted by  $w_{ij}$  which are the initial weights when implementing the network. If the degree of firing rule  $R_j$  is  $\alpha_j$ , we denote the degree of membership by  $U_{C_j}(u_i)$  where

$$U_{C_j}(u_i) \stackrel{\text{def}}{=} \alpha_j \cdot \mu_{C_j}(u_i).$$

At output  $u = u_i$ , let us define

$$\mu_i = \sum_{j=1}^m U_{C_j}(u_i) \quad (9)$$

where  $m$  is the number of rules.

The equation for output defuzzification using the center-of-area (COA) method is given by

$$u = \frac{\sum_{i=1}^n u_i \mu_i}{\sum_{i=1}^n \mu_i} \quad (10)$$

$$= \sum_{i=1}^n \mu_i \left( \frac{u_i}{\sum_{i=1}^n \mu_i} \right) \quad (11)$$

$$= \sum_{i=1}^n \mu_i \beta_i \quad (12)$$

where

$$\beta_i = \frac{u_i}{\gamma} \quad (13)$$

and

$$\gamma = \sum_{i=1}^n \mu_i. \quad (14)$$

## APPENDIX II

### DERIVATION OF THE LEARNING ALGORITHM

With reference to (1)–(6) in the paper, first we differentiate  $\beta_i$  w.r.t.  $\gamma$  in (2)

$$\frac{\partial \beta_i}{\partial \gamma} = \frac{-u_i}{\gamma^2}. \quad (15)$$

From (3), differentiate  $\gamma$  w.r.t.  $\mu_k$  where  $k = 1, \dots, n$

$$\frac{\partial \gamma}{\partial \mu_k} = 1. \quad (16)$$

From (15) and (16)

$$\frac{\partial \beta_i}{\partial \mu_k} = \frac{-u_i}{\gamma^2}. \quad (17)$$

From (4)

$$\frac{\partial u}{\partial \beta_i} = \mu_i \quad (18)$$

$$\frac{\partial u}{\partial \mu_i} = \beta_i. \quad (19)$$

From (5)

$$\frac{\partial \mu_i}{\partial w_{ij}} = \alpha_j. \quad (20)$$

Now

$$u = u(\mu_1, \mu_2, \dots, \beta_1, \beta_2, \dots).$$

Therefore

$$\begin{aligned}\frac{\partial u}{\partial w_{ij}} &= \sum_{l=1}^n \frac{\partial u}{\partial \mu_l} \frac{\partial \mu_l}{\partial w_{ij}} + \sum_{l=1}^n \frac{\partial u}{\partial \beta_l} \frac{\partial \beta_l}{\partial w_{ij}} \\ \frac{\partial u}{\partial w_{ij}} &= \sum_{l=1}^n \frac{\partial u}{\partial \mu_l} \frac{\partial \mu_l}{\partial w_{ij}} + \sum_{l=1}^n \sum_{k=1}^n \frac{\partial u}{\partial \beta_l} \frac{\partial \beta_l}{\partial \mu_k} \frac{\partial \mu_k}{\partial w_{ij}}.\end{aligned}$$

However

$$\frac{\partial \mu_k}{\partial w_{ij}} = 0$$

for  $k \neq i$ . This means that  $\mu_i$  is connected only to weights  $w_{ij}$  ( $j = 1, \dots, m$ ) only.

Therefore

$$\frac{\partial u}{\partial w_{ij}} = \frac{\partial u}{\partial \mu_i} \frac{\partial \mu_i}{\partial w_{ij}} + \sum_{l=1}^n \frac{\partial u}{\partial \beta_l} \frac{\partial \beta_l}{\partial \mu_i} \frac{\partial \mu_i}{\partial w_{ij}}.$$

Hence

$$\frac{\partial u}{\partial w_{ij}} = \beta_i \cdot \alpha_j + \sum_{l=1}^n \left[ \mu_l \cdot \frac{-u_l}{\gamma^2} \cdot \alpha_j \right] \quad (21)$$

from (17), (4), (18), and (19)

$$\Delta w_{ij} = \frac{-\partial E}{\partial w_{ij}}. \quad (22)$$

Hence, from (6) and (20)

$$\Delta w_{ij} = \alpha_j (u^d - u) \left( \beta_i - \sum_{l=1}^n u_l \mu_l / \gamma^2 \right) \quad (23)$$

e.g.,

$$\begin{aligned}\frac{\partial u}{\partial w_{23}} &= \sum_{l=1}^n \frac{\partial u}{\partial \mu_l} \frac{\partial \mu_l}{\partial w_{23}} + \sum_{l=1}^n \frac{\partial u}{\partial \beta_l} \frac{\partial \beta_l}{\partial w_{23}} \\ &= \frac{\partial u}{\partial \mu_1} \frac{\partial \mu_1}{\partial w_{23}} + \frac{\partial u}{\partial \mu_2} \frac{\partial \mu_2}{\partial w_{23}} + \frac{\partial u}{\partial \mu_3} \frac{\partial \mu_3}{\partial w_{23}} \\ &\quad + \frac{\partial u}{\partial \beta_1} \frac{\partial \beta_1}{\partial w_{23}} + \frac{\partial u}{\partial \beta_2} \frac{\partial \beta_2}{\partial w_{23}} + \frac{\partial u}{\partial \beta_3} \frac{\partial \beta_3}{\partial w_{23}} \\ &= \frac{\partial u}{\partial \mu_2} \frac{\partial \mu_2}{\partial w_{23}} + \frac{\partial u}{\partial \beta_1} \frac{\partial \beta_1}{\partial \mu_2} \frac{\partial \mu_2}{\partial w_{23}} \\ &\quad + \frac{\partial u}{\partial \beta_2} \frac{\partial \beta_2}{\partial \mu_2} \frac{\partial \mu_2}{\partial w_{23}} + \frac{\partial u}{\partial \beta_3} \frac{\partial \beta_3}{\partial \mu_2} \frac{\partial \mu_2}{\partial w_{23}} \\ &= \left[ \beta_2 - \left( \frac{\mu_1 u_1}{\gamma^2} + \frac{\mu_2 u_2}{\gamma^2} + \frac{\mu_3 u_3}{\gamma^2} \right) \right] \alpha_3.\end{aligned}$$

### APPENDIX III

#### DERIVATION OF THE ERROR TERM

$$e_j = -\frac{\partial E}{\partial \alpha_j} = -\sum_{i=1}^n \left[ \frac{\partial E}{\partial \mu_i} \frac{\partial \mu_i}{\partial \alpha_j} + \frac{\partial E}{\partial \beta_i} \frac{\partial \beta_i}{\partial \alpha_j} \right]. \quad (24)$$

Then

$$e_j = -\sum_{i=1}^n \left[ \frac{\partial E}{\partial u} \frac{\partial u}{\partial \mu_i} \frac{\partial \mu_i}{\partial \alpha_j} + \frac{\partial E}{\partial u} \frac{\partial u}{\partial \beta_i} \frac{\partial \beta_i}{\partial \alpha_j} \right]. \quad (25)$$

That is,

$$e_j = -\sum_{i=1}^n \left[ \frac{\partial E}{\partial u} \frac{\partial u}{\partial \mu_i} \frac{\partial \mu_i}{\partial \alpha_j} + \sum_{l=1}^n \frac{\partial E}{\partial u} \frac{\partial u}{\partial \beta_l} \frac{\partial \beta_l}{\partial \mu_i} \frac{\partial \mu_i}{\partial \alpha_j} \right].$$

But

$$\begin{aligned}\frac{\partial E}{\partial u} &= -(u^d - u) \\ \frac{\partial u}{\partial \mu_i} &= \beta_i \\ \frac{\partial \mu_i}{\partial \alpha_j} &= w_{ij} \\ \frac{\partial u}{\partial \beta_l} &= \mu_l \\ \frac{\partial \beta_l}{\partial \mu_k} &= -u_l / \gamma^2\end{aligned}$$

where  $k = 1, \dots, n$ .

Therefore

$$e_j = (u^d - u) \sum_{i=1}^n \left( \beta_i - \sum_{l=1}^n u_l \mu_l / \gamma^2 \right) w_{ij}. \quad (26)$$

### REFERENCES

- [1] T. van Vuren and D. van Vliet, *Route Choice and Signal Control*. Avebury ITS Institute for Transport Studies: Ashgate, 1992.
- [2] P. H. Bovy and E. Stein, *Route Choice: Wayfinding in Transport Networks*. Norwell, MA: Kluwer, 1990.
- [3] N. B. Hounsell, M. McDonald, and R. A. Lambert, "The integration of SCOOT and dynamic route guidance," in *Proc. 6th Int. Conf. Road Traffic Monitoring and Control*, Apr. 1992, pp. 168–172.
- [4] G. Hoffmann and J. Janko, "Travel times as a basic part of the LISB guidance strategy," in *Proc. 3rd Int. Conf. Road Traffic Control*, May 1990, pp. 6–10.
- [5] T. Saito, J. Shima, H. Kanemitsu, and Y. Tanaka, "Automobile navigation system using beacon information," in *Proc. 1st Int. Conf. Vehicle Navigation & Information Systems (VNIS)*, Toronto, Canada, 1989, pp. 139–144.
- [6] K. Sakai, H. Fukushima, and H. Fujii, "Characteristics of drivers' route selection and survey system," in *Meeting on Road Traffic and Automobile*, 1992, RTA-92-23.
- [7] M. Shibata, T. Matsumura, Y. Tanaka, and N. Imacho, "The economic effect of in-car navigation system," in *3rd Int. Conf. Vehicle Navigation & Information Systems (VNIS)*, Norway, Sept. 2–4, 1992, pp. 632–637.
- [8] P. W. Bonsall and T. Parry, "Drivers' requirements for route guidance," in *Proc. 3rd Int. Conf. Road Traffic Control*, May 1990, pp. 1–5.
- [9] D. Teodorovic and S. Kikuchi, "Transportation route choice model using fuzzy inference technique," in *Proc. 1st Int. Symp. Uncertainty Modeling and Analysis: Fuzzy Reasoning, Probabilistic Models, and Risk Management*, Univ. Maryland, College Park, 1990, pp. 140–145.
- [10] T. Lotan and H. N. Koutsopoulos, "Route choice in the presence of information using concepts from fuzzy control and approximate reasoning," *Transp. Planning Technol.*, vol. 17, pp. 113–126, 1993.
- [11] D. Teodorovic and M. Kalic, "A fuzzy route choice model for air transportation networks," *Transp. Planning Technol.*, vol. 19, pp. 109–119, 1995.
- [12] P. K. Simpson, *Artificial Neural Systems*. New York: Pergamon, 1990.
- [13] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1992.



**Grantham K. H. Pang** (M'86) received the Ph.D. degree from the University of Cambridge, Cambridge, U.K., in 1986 for research in multivariable control system design and expert systems.

He was with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, from 1986 to 1996. He joined the Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong, in 1996. Since 1988, he published more than 100 technical papers and has authored or coauthored three books.

In 1994, he was a Senior Visiting Researcher at Hitachi Research Laboratory, Japan. He has been a consultant to many companies including Mitsubishi Electric Corporation, Japan, Northern Telecom, Canada, and Imperial Oil Ltd., Canada. He is an Editor of the *International Journal of Intelligent Control and Systems*. He is also an Editor of *Control and Computers* published by the International Association of Science and Technology for Development (IASTED). His research interests include expert systems for control system design, intelligent control, intelligent transportation system, neural networks, control theory, and computer-aided design.

Dr. Pang received the ICI Prize for the best paper on application of the theory of control published in the *Transaction of Institute of Measurement and Control* in 1989. He was the Organizing Chair of the 1996 IEEE Symposium on Computer-Aided Control System Design. He was appointed by the President of the IEEE Control Systems Society as the Chair of the Technical Committee on Computer-Aided Control System Design (1993–1995). He is a Chartered Electrical Engineer and a member of the IEE and HKIE.



**Kazunori Takahashi** received the B.S. and M.S. degrees in electrical engineering from Hokkaido University, Sapporo, Japan, in 1985 and 1987, respectively.

He is currently a Senior Researcher at the Hitachi Research Laboratory, Hitachi, Ltd., Ibaraki-ken, Japan. His research interests include road traffic control and simulation technology in intelligent transport systems.



**Takayoshi Yokota** (M'83) was born in October 1956 in Tsuchira city, Ibaraki, Japan. He received the Ph.D. degree from the Tokyo Institute of Technology, Tokyo, Japan, in 1984.

Since 1984, he has been with the Hitachi Research Laboratory, Hitachi, Ltd., Ibaraki-ken, Japan. During 1988, he was a Visiting Researcher at Carnegie–Mellon University, Pittsburgh, PA. His background covers digital signal processing, traffic engineering, and artificial intelligence. He is now the Group Leader and a Senior Researcher of the

road traffic systems research group at Hitachi Research Laboratory.

Dr. Yokota is a member of the IEEE Computer Society.



**Hiroshi Takenaga** received the M.S. degree in electrical engineering from Kanazawa University, Japan, in 1978.

He is a Senior Researcher at the Hitachi Research Laboratory, Hitachi, Ltd., Ibaraki-ken, Japan. His current research focuses on developing the image processor and applying the pattern recognition algorithm to vehicle control.