A GIS-Based Design and Implementation Approach for Modeling Driver's Behavior in Route Selection Using Fuzzy-Neural Networks

Parham Pahlavani, Mahmoud Reza Delavar

Center of Exellence in Geomatics Eng. and Disaster Management, Dept. of Surveying and Geomatics Eng., Eng. Faculty, University of Tehran, Tehran, Iran

Summary

For modeling a driver's behavior in route selection in outdoor situations we have two problems: 1-real situation is very often not crisp and deterministic and cannot be described precisely, 2-the complete description of driver's behavior in route selection often require much more detailed data than driver could ever recognize process, and understand simultaneously. In this paper we have designed and implemented a GIS-based fuzzy-neural approach for modeling driver's behavior which represents the correlation of the attributes with the driver's route selection. A recommendation or route fitness is provided to the driver based on a training of the fuzzy adaptive neural network on the main criteria of route selection such as length, time and the degree of difficulty. Tests of route selection for a part of North-West of Tehran traffic network are conducted and the results show the efficiency of the algorithm and support our analyses.

Key words:

Route selection, deriver's behavior, fuzzy adaptive neural network(FALCON)

1. Introduction

Decisions are often evaluated on the basis of quality of the processes behind. It is in this context that geospatial information systems (GIS) and spatial decision support systems (SDSS) increasingly are being used to generate alternatives to aid decision-makers in their deliberations. Decision making itself, however, is broadly defined to include any choice or selection of alternative course of actions, and is therefore of importance in many fields in both the social and natural sciences including geospatial information sciences.

Among so many implementations GIS, a GIS application for Transportation (GIS-T) has become an outstanding one. It is possible to state unequivocally that GIS-T has arrived and now represents as one of the most important application areas of GIS.

Advanced Traveler Information Systems (ATIS) assist travelers with planning, perception, analysis and decision making to improve the convenience, safety and efficiency of travel. ATIS is one component of the Intelligent Transport Systems (ITS) that currently being developed to improve the safety and efficiency of automobile travel. Route planning is therefore an essential component of ATIS, aiding travelers in choosing the optimal path to their destinations in terms of travel distance, travel time and many other criteria. It is this multi-criteria aspect of route planning that we wish to tackle.

For the first time, we outlined a GIS-based novel approach for using a genetic algorithm for urban multi-objective optimized route selection in static environment [1] and an innovative method that extends the previous novel approach in order to include driver's unspecified sites [2]. Although the above approaches proposed the quasi-optimal route by the driver's consideration for the importance of each route criterion, it is essential to say that none of them concerned the driver's behavior.

It is believed that each driver has a set of route choice preferences. Very often, drivers would try to select the route which is optimum with reference to their preferences. In the other words, each quasi-optimal route may have an especial meaning for each driver, so it would be necessary to have a ranking route engine for proposed genetic algorithm in previous approaches in order to calculate the fitness of each route based on every driver's behavior and preferences.

The objectives of this paper are to design a ranking route engine as follows:

- It is a decision support system for route selection.
- It can model the behavior of the drivers by storing their preference and previous choices.
- It can adapt and learn from the recent decisions of the drivers.

Each route candidate has a set of attributes. A GIS-based fuzzy-neural 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 drivers based on a training of the fuzzy-neural network on the main criteria of route selection. This convenience is needed and may happen when planning a special trip on a particular day. It is used as a quick and convenient means

Manuscript received February 5, 2007
Manuscript revised February 25, 2007

for drivers to specify their requirements to the routing algorithm.
In section 2, related works are exhibited. The strategy for modeling deriver's behavior is outlined in section 3. In section 4, a method is proposed in modeling deriver's behavior. Experiments are presented in section 5, and finally section 6 is devoted to the results and their interpretations.

2. Related work

The use of fuzzy logic methodology in route selection was first proposed by Teodorovic and Kikuchi [3]. They have looked at the problem of route choices between two alternative 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 [4] have also proposed a modeling framework for route choice based on the driver's perception of attributes of the network, attractiveness of alternate routes as well as models for reaction to information. Such an approach works for a particular origin-destination(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 [5] have considered route choice problem in air transportation using fuzzy logic. In addition to 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 form the origin to the destination. 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. The researches in route choice selection using fuzzy logic are an ongoing process. As a conclusion, the main problem of approaches due to fuzzy reasoning for driver's behavior in route selection is to design the set of fuzzy rules according to numbers and the essence of any involved criteria in routing. As it will be discussed later, this modeling is not totally possible by the driver because of high complexity in decision making process.

3. Strategy

For modeling a driver's behavior in route selection in outdoor situations we have two problems: 1-real situations are very often not crisp and deterministic and cannot be described precisely, 2-the complete description of driver's behavior in route selection often requires much more detailed data than driver could ever recognize process, and understand simultaneously. In these situations, driver’s decisions are based on vague or imprecise concepts, which can often be expressed linguistically.
In one side a driver's choice of a route is normally based on a complex evaluation process in which the attribute of the entire feasible route are also measured subjectively. For example, despite the fact that estimated travel time is a measurable parameter, when drivers make the route choice, their notion of travel time is often fuzzy, also, they would tradeoff the different route criteria involved and make their judgment. The modeling of such a decision-making process of drivers is complex and it is believed that fuzzy logic and approximate reasoning model can help to understand the process.
At the other side neural networks can be developed to model the driver’s 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 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. For our application, the inputs will be the various attributes of a route and the output will be the fitness or an acceptance measure of the route. The neural network can be trained off line. The real-time execution of the neural network will be extremely fast. It also has the ability to adapt to different users. Any new user can train the network to learn his/her preferences.
A fuzzy adaptive neural network approach can combine the advantages of both fuzzy and neural network approaches.
We face the following problems in order to model the driver's behavior due to fuzzy adaptive neural network utilization:

1. The problem with fuzzy-neural networks is that they require good training data sets that should be a good representation of the complete data set. Moreover, such data could lead to an unpredictable convergence rate of the network learning, which may potentially threaten the successful application.
2. Because of high complexity in decision making process, it is impossible for driver to choose all the rules for the best alternative in route selection. So in modeling the driver's behavior, a method should be proposed with the capability of introducing some limited rules by driver in

preprocessing step along with the other rules achieved by GIS training data in processing step.

4. Proposed method

The proposed method for modeling the driver's behavior in this study is includes (1) the generating of training data, and (2) designing and implementing of a fuzzy adaptive neural network for answering the two latter problems discussed in previous section.
Figure 1 shows the general steps of the algorithm for modeling deriver's behavior and the details are presented next.

/*A. Generating of training data*/
1. Determining the importance or weight of the criterion of length(W_1), time (W_2) and degree of difficulty (W_3) by driver for each route ($R_{O,D}$) in which the sum of weighs equals to 1.
2. Running the proposed GA [1] for an origin-destination (O/D) in order to reach to a set of evolutionary route population.
3. Classifying the obtained evolutionary route collection from the proposed GA to 5 categories as very bad, bad, medium, good and very good in the range of [0,0.2), [0.2,0.4), [0.4,0.6), [0.6,0.8), and [0.8,1], respectively.
4. Choosing routes form previous 5 classes mentioned in step 3 and inserting them in the $X_{training}$ data set.

/*B. FALCON-H*/
5. Introducing some rules by the driver which are expressing his preferences
6. Self organizing learning phase in order to achieve other rules form training data generated in part A.
7. Supervised learning phase in order to adjust the parameters of the (input and output) membership functions optimally.

Fig. 1 The algorithm of the proposed method

4.1 Generating of training data

A route attribute is a characteristic of a route used by a driver as an assessment criterion in route selection. In this study the length, time and the degree of difficulty (DoD) of a route are considered as the primary criteria for route selection.
These three attributes are called primary attributes because they are primary in the sense that they are the important attributes and are widely-used by most drivers in the assessment of a route [6].
Given a set of origin-destination (O/D) pair, there could be many possible routes for a driver. Each of these candidate routes has different values in their primary attributes. One route may have a high value in one attribute (e.g. shortest distance) but a low value in another attribute (e.g. the route with the highest DoD).
As stated earlier, for the first time, we outlined a GIS-based novel approach for using a genetic algorithm for urban multi-objective optimized route selection in static environment [1].
As a result of running the above proposed GA, shown in step 2 of Figure 1, a population of candidate routes starts evolution in order to reach a population with the highest average fitness. The fitness of each route (**y^d**) in proposed GA, using sum of weighted global ratios (SWGR) [7] is designed in such a way that its value would be in a range of [0,1] i.e., 1 is the highest fitness value and 0 is the lowest.
Proposed GA utilization produces a vast range of routes which could be used as the preferred routes by the driver. For this purpose the routes were classified to the 5 categories according to their fitness shown in step 3 of Figure 1. This classification is done due to fuzzy adaptive neural network design, described in section 4.2.
Figure 2 illustrates the step 4 of the Figure 1.

/***R_1:** first route; **R_2:** second route; **y_1^d and y_2^d:** fitness of first and second /*routes respectively computed by proposed GA; **C_1,C_2,C_3,C_4 and C_5:** /*route fitness classes including very good, good, medium, bad and very /*bad respectively
1. For i=1 to 5 /*Due to 5 classes*/
2. Select two random routes (R_1 and R_2)from a class C_i
3. If the driver chooses non of the two routes, then Go to step 2
4. If the driver chooses one route (e.g., R_1) and i=1 or i=2 or i=3 then
5. If $y_1^d < y_2^d$ then $y_1^d = y_2^d$ end if
6. Else if the driver chooses one route (e.g., R_1) and i=4 or i=5 then
7. If $y_1^d > y_2^d$ then $y_1^d = y_2^d$ end if
8. End if
9.Compute x_1,x_2 and x_3 of selected route e.g.,R_1 (from Eqs 1,2 and 5)
10.Form the path descriptive vector (pdv)of selected route (e.g., pdv(R_1)= (x_1,x_2,x_3,y_1^d))and insert it in $X_{training}$ data set
11. next i

Fig. 2 The algorithm of route selection in driver's view point

After accomplishing the steps 4 to 7 of Figure 2 by the driver, the selected weights in step 1 of Figure 1 will be changed.
The step 9 shown in Figure 2, will be described in section 4.1.1.

4.1.1 The specific approach used in measuring each route criteria

In this part, every rate of length (x_1), time (x_2) and the degree of difficulty (x_3) of each training data will be calculated. A detailed discussion of each of the primary attributes is given bellow. Each attribute is designed to have a score between zero and one.

4.1.1.1 Travel distance and travel time

Let x_1^k be used to describe the length rate attribute of route k. For this attribute, a score of 1 denotes that this route has the shortest travel distance among the particular set of route candidates in $X_{training}$ and a score of 0 designates that this route has the longest travel distance among the particular set of route candidates in $X_{training}$. For other candidate routes, the score of this attributes is defined in Eq.(1) [1],[2]:

$$x_1^k = \text{length\_ratio}_k = \frac{\max(\text{length\_value}) - \text{length\_value}_k}{\max(\text{length\_value}) - \min(\text{length\_value})} \quad (1)$$

min(length_value) = best length values in all GA populations
max(length_value) = worst length values in all GA populations

Moreover, min(length_value) is equal to shortest length values obtained by independent run of modified Dijkstra algorithm based on d-Heap's structure with d=2 for length criterion.

"Travel time" attribute can be developed in a way similar to travel distance shown is Eq.(2) [1], [2]:

$$x_2^k = time\_ratio_k = \frac{\max(time\_value) - time\_value_k}{\max(time\_value) - \min(time\_value)} \quad (2)$$

min(time_value) = best time values in all GA populations
max(time_value) = worst time values in all GA populations

For this attribute, a score of 1 denotes that this route has the shortest travel time among the particular set of route candidates in $X_{training}$ and a score of 0 designates that this route has the longest travel time among the particular set of route candidates in $X_{training}$.

Moreover, min(time_value) is equal to shortest length values obtained by independent run of modified Dijkstra algorithm based on d-Heap's structure with d=2 for time criterion.

4.1.1.2 Degree of difficulty

This attribute can be computed as a function of the type and nature of the road such as the narrowness, winding, slope, number of traffic lights and number of stop signs. A simpler way would be to assign different values for different types of road. For example, the following table can be used as a guideline for determining the degree of difficulty (DoD) of the route [1], [2].

Table 1: Guideline for determining degree of difficulty (DoD)

Road type	Penalty for DoD
Exp. Way with a negative slope	0
Exp. Way without a slope	0.1
Exp. Way with a positive slope	0.2
Major Arterial (outside the central district)	0.4
Major Arterial(inside the central district)	0.5
Minor Arterial	0.6
Collector-Feeder	0.8
Local street	1

4.1.1.3 Overall score of degree of difficulty for each candidate route

For each candidate route, the travel distance and travel time are supposed to be known and a score in the range [0,1] can be calculated for each of these attributes. However, different road sections for each candidate route would have different attribute scores for degree of difficulty. In order to calculate the complete set of primary attributes of the route candidate, a method is needed to combine the attribute score of different road sections into an overall score for degree of difficulty. A method is developed for this calculation which is described below:

Let n be the number of road sections of the route candidate.

Let d_p and c_p be the distance and degree of difficulty attribute scores of each road section, respectively (p = 1 to n).

Let w_p be a weight of each road section which is defined in Eq.(3) [1], [2]:

$$w_p = \frac{d_p}{\sum_{p=1}^{n} d_p} \quad (3)$$

Then, the overall score of the degree of difficulty for candidate route is calculated using Eq.(4) [1], [2]:

$$DoD = DoD\_value_k = \sum_{p=1}^{n} w_p \times c_p \quad (4)$$

Note that $\sum_{p=1}^{n} w_p = 1$ and $0 \le c_p \le 1$. The final overall score of the degree of difficulty attribute (DoD_value_k) is also in the range [0,1].

Let x_3^k be used to describe this attribute of route k, then the score of this attributes is defined in Eq.(5) [1], [2]:

$$x_3^k = DoD\_ratio_k = \frac{\max(DoD\_value) - DoD\_value_k}{\max(DoD\_value) - \min(DoD\_value)} \quad (5)$$

min(DoD_value) = best DoD values in all **GA** populations
max(DoD_value) = worst DoD values in all **GA** populations

For this attribute, a score of 1 denotes that this route has the lower travel degree of difficulty among the particular set of route candidates in $X_{training}$ and a score of 0 designates that this route has the highest travel degree of difficulty among the particular set of route candidates in $X_{training}$.

Moreover, min(DoD_value) is equal to lower degree of difficulty values obtained by independent run of modified Dijkstra algorithm based on d-Heap's structure with d=2 for degree of difficulty criterion.

4.2 Proposed fuzzy-neural control

We use a general connectionist model, the fuzzy adaptive learning control network (FALCON) proposed by Lin and Lee to study hybrid structure parameter learning strategies [8,9]. The FALCON is a feedforward multilayer network which integrates the basic elements and functions of a traditional fuzzy logic controller into a connectionist structure that has distributes learning abilities. In this connectionist structure, the input and output nodes represent the input states and output control or decision signals, respectively, and in the hidden layers, there are nodes functioning as membership functions and fuzzy logic rules. The FALCON can be contrasted with a traditional fuzzy logic control and decision system in terms of its network structure and learning abilities. Such fuzzy control and decision networks can be constructed from training examples by neural learning techniques, and the connectionist structure can be trained to develop fuzzy logic rules and determine proper input-output membership functions. This connectionist model also provides human-understandable meaning to the normal feedforward multilayer neural network in which the internal units are always opaque to users. So, if necessary, expert knowledge can be easily incorporated into the FALCON. The connectionist structure also avoids the rule-matching time of the inference engine in the traditional fuzzy control system. The structure and function of the proposed FALCON and its learning scheme are described bellow.

Figure 3 shows the structure of the FALCON. The system has a total of five layers. The nodes in layer 1 are input nodes (*linguistic nodes*) that represent input linguistic variables including length rate (x_1), time rate (x_2), and the degree of difficulty rate (x_3) of each route in the $X_{training}$ data set, and layer 5 is the output layer. There are two linguistic nodes for each output variable. One is for training data (desired output or y^d) to feed into the network, and the other is for decision signals (actual output or y') to be pumped out of the network for each route in the $X_{training}$ data set. Nodes in layers 2 and 4 are *term nodes* which act as membership functions representing the terms of the respective linguistic variables.

The linguistic variables and the terms related to layers 1,2,4 and 5 are presented in Table 2.

Table 2: Proposed FALCON linguistic variables and terms

	Linguistic Variables	*Linguistic terms*
Input	x_1	too_long, long, medium, short, too_short
	x_2	too_long, long, medium, short, too_short
	x_3	very_high,high,medium,low,very_low
Output	y'	very_bad,bad,medium,good,very good
	y^d	very_bad,bad,medium,good,very good

Each node in layer 3 is a rule node that represents one fuzzy logic rule. Links in layers 3 and 4 function as a *connectionist inference engine,* which avoids the rule-matching process. Layer 3 links define the preconditions of the rule nodes and layer 4 links define the consequents of the rule nodes. Therefore, for each rule node, there is at most one link (maybe none) from some term nodes of a linguistic node. This is true for both precondition links (links in layer 3) and consequent links (links in layer 4). The links in layers 2 and 5 are fully connected between linguistic nodes and their corresponding terms nodes. The arrow on the link indicates the normal signal flow direction when the network is in use after it has been built and trained.

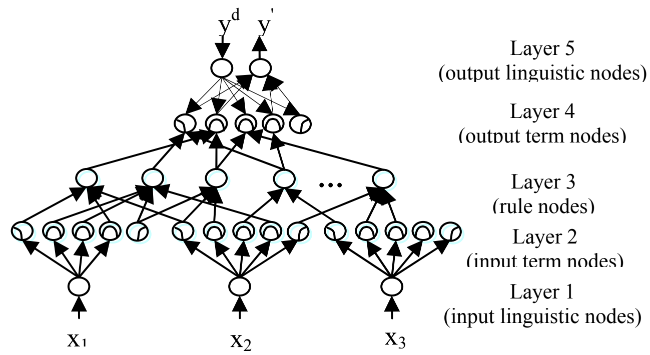

Fig. 3 The structure of the proposed FALCON

With this five layered structure of the FALCON, we shall define the basic functions of a node. The FALCON consists of nodes that have some finite fan-in of connections represented by weight values from other nodes and a fan-out of connections to other nodes. Associated with the fan-in of a node is an integration function *f* which serves to combine information, activation, or evidence from other nodes. This function provides the net input to this node

$$net_i = f(u_1^{(k)}, u_2^{(k)}, \ldots, u_p^{(k)}; w_1^{(k)}, w_2^{(k)}, \ldots, w_p^{(k)}) \tag{6}$$

where $u_1^{(k)}, u_2^{(k)}, \ldots, u_p^{(k)}$ are inputs to this node and $w_1^{(k)}, w_2^{(k)}, \ldots, w_p^{(k)}$ are the associated link weights. The superscript k in the Eq.(6) indicates the layer number. A second action of each node is to output an activation value as a function of its net input:

$$\text{output} = o_i^{(k)} = a(net_i) = a(f) \tag{7}$$

where a(.) denotes the activation function. The functions of the nodes in each of the five layers of the FALCON are described below.

Layer 1: The nodes in this layer only transmit input values to the next layer directly. That is,

$$f = u_i^{(1)} \quad \text{and} \quad a = f. \tag{8}$$

From Eq.(8), the link weight at layer 1 ($w_i^{(1)}$) is unity.

Layer 2: We use a single node to perform a bell-shaped membership function:

$$f = M_{x_i}^{j}(m_{ij}, \sigma_{ij}) = -\frac{(u_i^{(2)} - m_{ij})^2}{\sigma_{ij}^2} \quad \text{and} \quad a = e^f \tag{9}$$

where m_{ij} and σ_{ij} are, respectively, the center (or mean) and the width (or variance) of the bell-shaped function of the *j*th term of the *i*th input linguistic variable x_i. Hence, the link weight at layer 2 ($w_{ij}^{(2)}$) can be interpreted as m_{ij}.

Layer 3: The links in this layer are used to perform precondition matching of fuzzy logic rules. Hence, the rule nodes perform the fuzzy AND operation,

$$f = \min(u_1^{(3)}, u_2^{(3)}, \ldots, u_p^{(3)}) \quad \text{and} \quad a = f. \tag{10}$$

The link weight in layer 3 ($w_i^{(3)}$) is then unity.

Layer 4: The nodes in this layer have two operation modes: *down-up* transmission and *up-down* transmission modes. In the down-up transmission mode, the links in layer 4 perform the fuzzy OR operation to integrate the fired rules which have the same consequent:

$$f = \sum_i u_i^{(4)} \quad \text{and} \quad a = \min(1, f). \quad (11)$$

Hence, the link weight $w_i^{(4)}$ =1. In the up-down transmission mode, the nodes in this layer and the links in layer 5 function exactly the same as those in layer 2 except that only a single node is used to perform a membership function for output linguistic variables.

Layer 5: There are two kinds of nodes in this layer also. The first kind of node performs up-down transmission for training data being fed into the network. For this kind of node,

$$f = y_i \quad \text{and} \quad a = f. \quad (12)$$

The second kind of node performs down-up transmission for the decision signal output. These nodes and the layer 5 links attached to them act as the defuzzifier. If m_{ij} and σ_{ij} are, respectively, the center and the width of the membership function of the *j*th term of the *i*th output linguistic variable, then the Eq.(13) can be used to simulate the *center of area* defuzzification method:

$$f = \sum_j w_{ij}^{(5)} u_{ij}^{(5)} = \sum_j (m_{ij}\sigma_{ij}) u_{ij}^{(5)} \quad \text{and} \quad a = \frac{f}{\sum_j \delta_{ij} u_{ij}^{(5)}}. \quad (13)$$

Here the link weight in layer 5 ($w_i^{(5)}$) is $m_{ij}\sigma_{ij}$.

Based on this connectionist structure, a supervised gradient-descent learning procedure is developed to determine the proper centers (m_{ij}) and widths (σ_{ij}) of the term nodes in layers 2 and 4.

We shall now present a hybrid learning algorithm to set up the FALCON (FALCON-H) from a set of supervised training data. The hybrid learning algorithm consists of two separate stages of a learning strategy which combines unsupervised learning and supervised gradient-descent learning procedures to build the rule nodes and train the membership functions. In phase 1 of the hybrid learning algorithm, a self-organized learning scheme (i.e., unsupervised learning) is used to locate initial membership functions and to detect the presence of fuzzy logic rules. In phase 2, a supervised leaning scheme is used to optimally adjust the parameters of the membership functions for desired outputs.

4.2.1 Self-organized learning phase

It is possible to introduce some limited rules by the driver in the proposed FALCON which are his ideal condition indicators. It is for being assured that all the needed knowledge from the deriver is transferred in order to model his behavior. Other rules are derived from the self-organized learning phase.

The problem for self-organized learning can be stated as follow: Given the training input data $x_i(t)$, i=1,2,3, the corresponding desired output value $y^d(t)$, the fuzzy partitions $|T(x_i)|$ and $|T(y)|$ and the desired shapes of membership functions, we want to locate the membership functions and find the fuzzy logic rules. Here $|T(x_i)|$ denotes the number of x_i terms (i.e., the number of fuzzy partitions of the input state linguistic variable).

In this phase, the network works in a two-sided manner; that is, the nodes and links in layer 4 are in the up-down transmission mode so that the training input and output data can be fed into the FALCON from both sides.

First, the centers (or means) and the widths (or variances) of the membership functions are determined by self-organized learning techniques analogous to statistical clustering technique. This serves to allocate network resources efficiently by placing the domains of membership functions covering only those regions of the input-output space where data is present. Kohonen's learning rule algorithm [9,10] is adopted here to fine the center m_{ij} of the *i*th membership function of 'x', where 'x' represents any one of the input or output linguistic variables $x_1, \ldots, x_n, y_1, \ldots, y_n$;

$$\|x(t) - m_{closest}(t)\| = \min_{1 \le i \le k}\{\|x(t) - m_i(t)\|\}, \quad (14)$$

$$m_{closest}(t+1) = m_{closest}(t) + \alpha(t)[x(t) - m_{closest}(t)] \quad (15)$$

$$m_i(t+1) = m_i(t), \text{ for } m_i \neq m_{closest} \quad (16)$$

where $\alpha(t)$ is a monotonically decreasing scalar learning rate and k= $|T(x)|$. This adaptive formulation runs independently for each input and output linguistic variable. The determination of which of the m_i is $m_{closest}$ can be accomplished in constant time via a winner-take-all circuit. Once the centers of the membership functions are found, their widths can be determined using the *N-nearest-neighbor* heuristic by minimizing the Eq.(17) with respect to the widths σ_{ij} [9]:

$$E = \frac{1}{2}\sum_{i=1}^{N}[\sum_{j \in N_{nearest}} (\frac{m_i - m_j}{\sigma_i})^2 - r]^2 \quad (17)$$

where "r" is an overlap parameter. Since the second learning phase will optimally adjust the centers and the widths of the membership functions, the widths can be simply determined by the first-nearest-neighbor heuristic at this stage as represented in Eq.(18) [9]:

$$\sigma_i = \frac{|m_i - m_{closest}|}{r} \quad (18)$$

After parameters of the membership functions have been found, the signals from both external sides can reach the output points of term nodes in layers 2 and 4. Furthermore, the outputs of terms nodes in layer 2 can be transmitted to rule nodes through the initial connection of layer 3 links. So we can obtain the firing strength of each rule node. Based on these rule firing strengths [denoted as $o_i^{(3)}(t)$] and the outputs of term nodes in layer 4 [denoted as $o_j^{(4)}(t)$], we want to determine the correct consequent links (layer 4 links) of each rule node to fine the existing fuzzy logic rule by *competitive learning* algorithms. As stated before, the links in layer 4 are initially fully connected. We denote the weight of the link between the *i*th rule node and the *j*th output term node as w_{ji}. The following competitive learning law is used to update these weights for each training data set [9,11],

$$\dot{w}_{ji}(t) = o_j^{(4)}(-w_{ji} + o_i^{(3)}) \tag{19}$$

where $o_j^{(4)}$ serves as a win-loss index of the *j*th term node in layer 4. The essence of this law is *learn if win.* In the extreme case, if $o_j^{(4)}$ is a 0/1 threshold function, then this law indicates *learn only if win.*

After competitive learning involving the whole training data set, the link weights in layer 4 represent the strength of the existence of the corresponding rule consequent. From the links connecting a rule node and the term nodes of an output linguistic node, at most one link with maximum weight is chosen and the others are deleted. Hence, only one term in an output linguistic variable's terms set can become one of the consequents of a fuzzy logic rule. If all the link weights between a rule node and the term nodes of an output linguistic node are very small, then all the corresponding links are deleted, meaning that this rule node has little or no relation to this output linguistic variable. If all the links between a rule node and the layer 4 nodes are deleted, then this rule node can be eliminated since it does not affect the outputs.

After the consequents of rule nodes are determined, a rule combination is used to reduce the number of rules. The criteria for combining a set of rule nodes into a single rule node are: (1) they have exactly the same consequents, (2) some preconditions are common to all the rule nodes in the set, and (3) the union of other preconditions of these rule nodes comprised the whole term set of some input linguistic variables. If a set of nodes meet these criteria, a new rule node with only the common preconditions can replace this set of rule nodes.

4.2.2 Supervised learning phase

The problem for supervised learning can be stated as: Given the training input data $x_i(t)$, i=1,2,3 the corresponding desired output value $y^d(t)$, the fuzzy partitions $|T(x_i)|$ and $|T(y)|$, and the fuzzy logic rules, adjust the parameters of the input and output membership functions optimally. Here $|T(x_i)|$ denotes the number of x_i terms (i.e., the number of fuzzy partitions of the input state linguistic variable). In supervised learning phase, the network works in the feed-forward manner; that is, the nodes and the links in layers 4 and 5 are in the down-up transmission mode. The back-propagation algorithm is used for this supervised learning. Considering a single-output case, the goal is to minimize the error function presented in Eq.(20).

$$E = \frac{1}{2}(y^d(t) - y(t))^2 \tag{20}$$

where $y^d(t)$ is the desired output and y(t) is the current output. For each training data set, starting at the input nodes, a forward pass is used to compute the activity levels of all the nodes in the network to obtain the current output y(t). Then, starting at the output nodes, a backward pass is used to compute $\partial E / \partial w$ for all the hidden nodes. Assuming that w is the adjustable parameter in a node (e.g., m_{ij} and σ_{ij} in our case), the general learning rule used is shown in Eq.(21).

$$w(t+1) = w(t) + \eta(-\frac{\partial E}{\partial w}) \tag{21}$$

where η is the learning rate and

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial(\text{activation function})}\frac{\partial(\text{activation function})}{\partial w} = \frac{\partial E}{\partial a}\frac{\partial a}{\partial w} \tag{22}$$

To illustrate the learning rule for each parameter, we shall show the computations of $\frac{\partial E}{\partial w}$, layer by layer, starting at the output nodes. We will use bell-shaped membership functions with centers m_i and widths σ_i (single-output case) as the adjustable parameters for these computations.

Layer 5: The adaptive rule of the center m_i is derived as presented in Eq.(23) [9].

$$m_i(t+1) = m_i(t) + \eta[y^d(t) - y(t)]\frac{\sigma_i u_i^{(5)}}{\sum_i \sigma_i u_i^{(5)}} \tag{23}$$

The adaptive rule of the width σ_i is derived as shown in Eq.(24) [9].

$$\sigma_i(t+1) = \sigma_i(t) + \eta[y^d(t) - y(t)] * \frac{m_i u_i^{(5)}(\sum_i \sigma_i u_i^{(5)}) - (\sum_i m_i \sigma_i u_i^{(5)}) u_i^{(5)}}{(\sum_i \sigma_i u_i^{(5)})^2} \tag{24}$$

The error propagated to the preceding layer is calculated using Eq. (25) [9].

$$\delta^{(5)} = -\frac{\partial E}{\partial a^{(5)}} = -\frac{\partial E}{\partial y} = y^d(t) - y(t) \tag{25}$$

Layer 4: In the down-up transmission mode, there is no parameter to be adjusted in this layer. Only the error signals ($\delta_i^{(4)}$) need to be computed and propagated. The error signal $\delta_i^{(4)}$ is derived as shown in Eq. (26) [9].

$$\delta_i^{(4)}(t) = [y^d(t) - y(t)] * \frac{m_i\sigma_i(\sum_i \sigma_i u_i^{(5)}) - (\sum_i m_i \sigma_i u_i^{(5)})\sigma_i}{(\sum_i \sigma_i u_i^{(5)})^2} \tag{26}$$

Layer 3: As in layer 4, only the computation of error signals is required. This error signal can be derived as shown in Eq.(27) [9].

$$\delta_i^{(3)} = \delta_i^{(4)} \tag{27}$$

Layer 2: The adaptive rule of m_{ij} (multi-input case) is derived as represented in Eq.(28) [9].

$$m_{ij}(t+1) = m_{ij}(t) + \eta\delta_i^{(2)} e^{fi} \frac{2(u_i^{(2)} - m_{ij})}{\sigma_{ij}^2} \tag{28}$$

The update rule of σ_{ij} becomes is shown in Eq. (29) [9].

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \eta\delta_i^{(2)} e^{fi} \frac{2(u_i^{(2)} - m_{ij})^2}{\sigma_{ij}^3} \tag{29}$$

5. Experiments

The proposed method due to fuzzy adaptive learning control network in GIS was implemented by ArcGIS utilization and customization. ArcGIS has a unique feature in architectural design which enables it to be developed by COM programming in any visual environment.

To evaluate the performance of the outlined method, we performed some experiments, using actual road maps of a part of North-West of Tehran network with 5121 edges and 4389 nodes at a scale of 1:2000. In addition, each result given bellow was performed on AMD Athlon(tm) XP 1600+ (1.40 GHz).

To study the training process of the proposed neuro-fuzzy inference system, 20 training data sets were selected from the proposed method in Figure 2 having min(length_value)= 6568.12, max(length_value) = 8564.34, min(time_value) = 14.08, max(time_value)= 26.14, min(DoD_value) = 0.28, max(DoD_value)= 0.53, with the adjusted criterion weight: w_1=0.25, w_2=0.65 and w_3=0.1 called control reference data. The initial weights of each route criterion (step 1; Figure 1) were w_1=0.2, w_2=0.6 and w_3=0.2 respectively which were changed after choosing preferred routes by the driver (steps 4 to 7; Figure 2). To do so, the adjusted weights are calculated as following:

If $X=[\underline{x}_1,\underline{x}_2,\underline{x}_3]$; $W=[w_1,w_2,w_3]$ and $Y^d=[\underline{y}^d]$ then $XW=Y^d$

$$\Rightarrow \quad W=(X^TX)^{-1}X^TY^d \tag{30}$$

With this amount of training data, it can be expected that the variety of the appearances of driver's behavior in route selection is high which strengthens the significance of the training results, in particular, the differences between the initial set membership parameters and the adapted ones after learning. 10 training samples have been selected additionally with the adjusted criteria weight: w_1=0.25, w_2=0.65 and w_3=0.1 to check the learning process called check reference data.

The control reference data and the check reference data are presented in Tables 3 and 4 respectively.

Table 3: The control reference data

	Length (m)	Time (min)	DoD	x_1	x_2	x_3	y^d	y'
C1	7039.9	14.08	0.364	0.764	1.000	0.653	0.906	0.899
	7343.4	14.15	0.327	0.612	0.994	0.804	0.879	0.870
	7444.9	14.81	0.342	0.561	0.939	0.742	0.825	0.817
	7603.4	14.65	0.357	0.481	0.953	0.682	0.808	0.799
C2	7743.9	14.55	0.359	0.411	0.961	0.676	0.795	0.788
	7607.8	15.35	0.339	0.479	0.895	0.754	0.777	0.769
	7513.2	15.87	0.387	0.527	0.852	0.564	0.742	0.733
	7453.5	18.5	0.350	0.556	0.633	0.710	0.622	0.613
C3	8002.5	18.05	0.356	0.281	0.671	0.686	0.575	0.566
	7823.5	18.05	0.469	0.371	0.671	0.236	0.552	0.544
	8054.5	18.46	0.477	0.255	0.637	0.204	0.498	0.508
	8286.2	18.94	0.523	0.139	0.597	0.016	0.424	0.435
C4	8250.5	19.98	0.510	0.157	0.511	0.070	0.378	0.384
	8100.1	22.02	0.400	0.233	0.342	0.510	0.331	0.337
	8425.4	21.01	0.493	0.070	0.425	0.140	0.308	0.296
	8308.5	22.5	0.480	0.128	0.302	0.190	0.247	0.238
C5	8425.1	23.55	0.490	0.070	0.215	0.150	0.172	0.161
	8350.7	24	0.490	0.107	0.177	0.150	0.157	0.145
	8221.6	25.02	0.490	0.172	0.093	0.150	0.118	0.108
	8400.5	25.6	0.500	0.082	0.045	0.110	0.061	0.074

Table 4: The check reference data

	Length (m)	Time (min)	DoD	x_1	x_2	x_3	y^d	y'
C1	7232.4	14.3	0.315	0.667	0.980	0.848	0.889	0.876
	7445.1	14.9	0.351	0.561	0.931	0.704	0.816	0.803
C2	7258.1	16.2	0.333	0.654	0.828	0.778	0.780	0.794
	6568.1	17.9	0.375	1.000	0.680	0.610	0.753	0.743
C3	7946.5	18.6	0.482	0.310	0.629	0.180	0.504	0.516
	8206.9	18.8	0.512	0.179	0.611	0.060	0.448	0.434
C4	8354.6	19.6	0.484	0.105	0.546	0.172	0.399	0.386
	8277.9	22.7	0.490	0.144	0.285	0.150	0.236	0.232
C5	8221.6	25	0.460	0.172	0.093	0.270	0.130	0.116
	8285.4	25.1	0.510	0.140	0.090	0.070	0.100	0.097

As stated earlier in section 4.2.1, each driver would have his own perspective of a desirable route. The FALCON is designed in order to let a driver specify his preferences using fuzzy rules with some predefined linguistic terms.

Other rules are derived from the self-organized learning phase. In this part of the experiment, a set of fuzzy rules are defined by the driver:

- **IF** *Length* is *too_long* **AND** *time* is *too_long* **AND** *DoD* is *very_high* **THEN** *route fitness* is *very_bad*
- **IF** *Length* is *very_short* **AND** *time* is *long* **AND** *DoD* Is *very_low* **THEN** *route fitness* is *bad*
- **IF** *Length* is *medium* **AND** *time* is *medium* **AND** *DoD* is *medium* **THEN** *route fitness* is *bad*

- **IF** *Length* is *short* **AND** *time* is *medium* **AND** *DoD* is *medium* **THEN** *route fitness* is *medium*
- **IF** *Length* is *short* **AND** *time* is *medium* **AND** *DoD* is *low* **THEN** *route fitness* is *medium*
- **IF** *Length* is *very_short* **AND** *time* is *medium* **AND** *DoD* is *very_ low* **THEN** *route fitness* is *good*
- **IF** *Length* is *medium* **AND** *time* is *short* **AND** *DoD* is *medium* **THEN** *route fitness* is *good*
- **IF** *Length* is *short* **AND** *time* is *short* **AND** *DoD* is *low* **THEN** *route fitness* is *very_ good*
- **IF** *Length* is *short* **AND** *time* is *very_short* **AND** *DoD* is *medium* **THEN** *route fitness* is *very_ good*

To implement the proposed FALCON in modeling deriver's behavior, the membership functions of the length rate(x_1), time rate (x_2) and the degree of difficulty rate (x_3) as well as the route fitness (y) descriptors have to be specified. The initial membership functions are depicted in Figures 4,5,6 and 7.

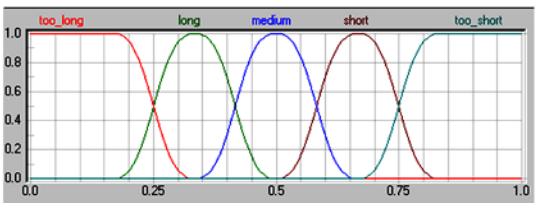

Fig. 4 Initial membership function of Length ratio(x_1)

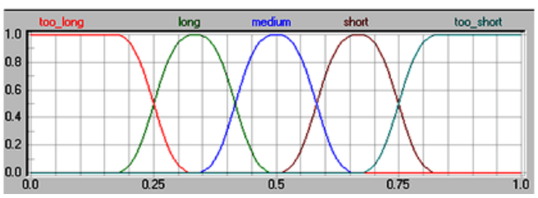

Fig. 5 Initial membership function of Time ratio(x_2)

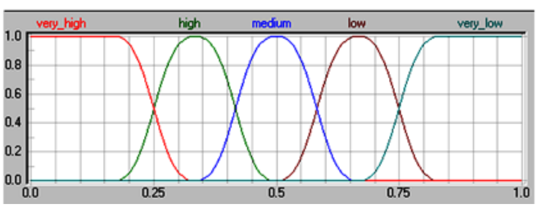

Fig. 6 Initial membership function of DoD ratio(x_3)

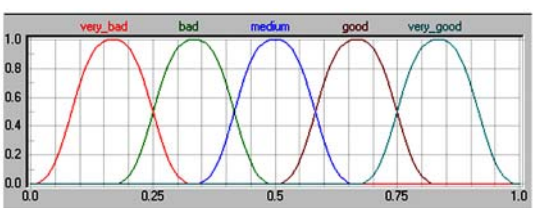

Fig. 7 Initial membership function of route fitness (y)

All the control and check reference data are participated in the self-organized learning phase in order to locate the membership functions and find the rest of fuzzy logic rules. Check reference data have not been included in the objective function (Equation 20) of the optimization process i.e., in supervised learning phase.

The overlap parameter 'r' is set to 2.0, the learning rate (α *and* η) is 0.15, and the error tolerance is 0.01. After Self-organized learning phase, 51 fuzzy logic rules are obtained.

In order to perform rule verification, the 3D plot of changeability of the route fitness plane is shown in Figures 8 and 9, respectively.

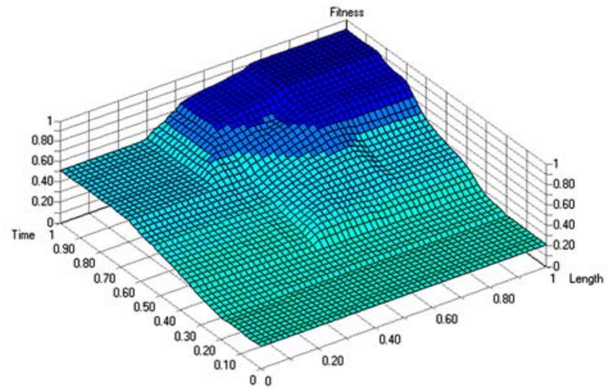

Fig. 8 The changeability of the route fitness plane with respect to time rate(x_2) and length rate(x_1)

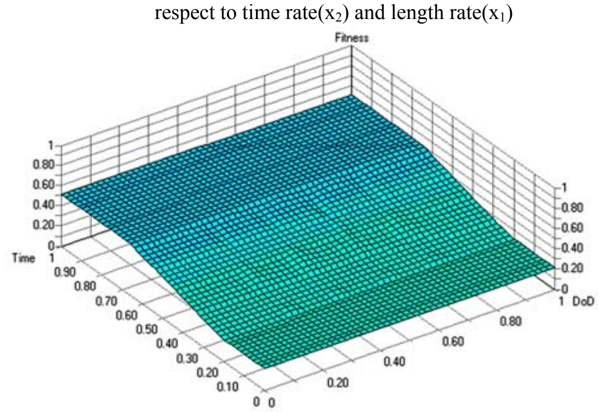

Fig. 9 The changeability of the route fitness plane with respect to time rate(x_2) and the degree of difficulty rate(x_3)

As shown in Figures 8 and 9, the more the length rate, time rate and DoD rate become closer to 1, the higher fitness value is achieved.

Moreover, the degree of importance of each route criterion (w_1=0.25, w_2=0.65 and w_3=0.1) are clear in the Figures. For instance, if time rate would be in the range of 0.7 to 1, the fitness plane is around 0.5 and it is because of higher importance degree of time with respect to the degree of difficulty (Figure 8).

For the whole areas of the 3D Plot, all values of the transfer characteristic are plotted. This means, we have not areas in which no inference result was derived by the system due to either non-overlapping membership functions or undefined rules. It means that the proposed method was capable to generate the training data, and the generated data could be sufficient for extracting the rest of fuzzy rules.

The learned membership functions of x_1,x_2,x_3 and y after unsupervised self organized learning phase are shown in Figures 10,11,12 and 13, respectively.

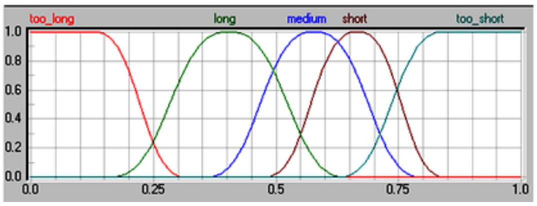

Fig. 10 Membership function of Length ratio(x_1) after unsupervised learning

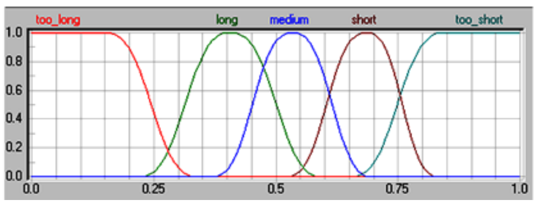

Fig. 11 Membership function of Time ratio(x_2) after unsupervised learning

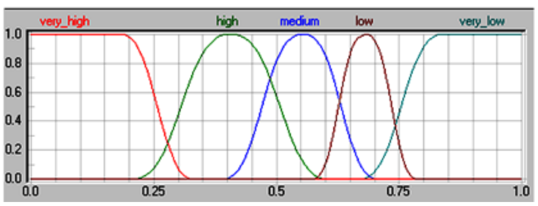

Fig. 12 Membership function of DoD ratio(x_3) after unsupervised learning

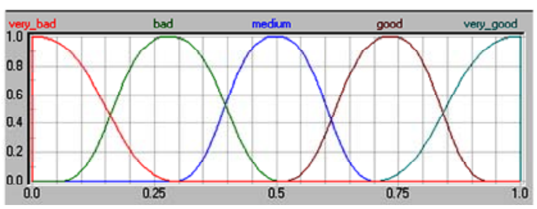

Fig. 13 Membership function of route fitness (y) after unsupervised learning

By taking the output of the trained proposed FALCON for the reference data a corresponding mean square error measure related to the 'fitness truth' (y^d), output can be calculated. Figure 14 shows the error measure as a function of the number of iterations. The curve for the error measure calculated with the check reference data is, as expected, slightly above the error curve obtained for the control reference training data. The very small difference for the final iterations indicates that the achieved route fitness with the trained proposed FALCON agrees well with the check reference data.
The column y' which is the indicator of the proposed FALCON output after supervised training is shown in Tables 3 and 4.

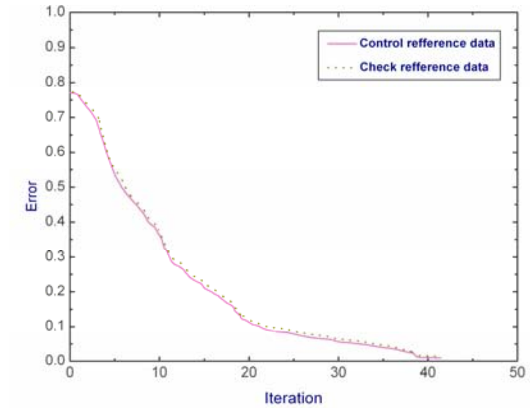

Fig. 14 Training data graph

Figures 15,16,17 and 18 show the adapted membership functions after training. The comparison with the initial membership functions indicates that most of the membership functions have changed more significantly indicating that for these ones the training was quite useful.

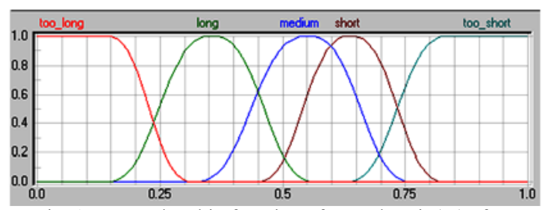

Fig. 15 Membership function of Length ratio(x_1) after supervised learning

Fig. 16 Membership function of Time ratio(x_2) after supervised learning

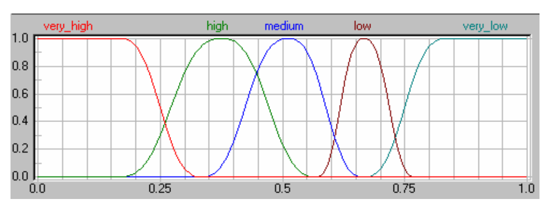

Fig. 17 Membership function of DoD ratio(x_3) after supervised learning

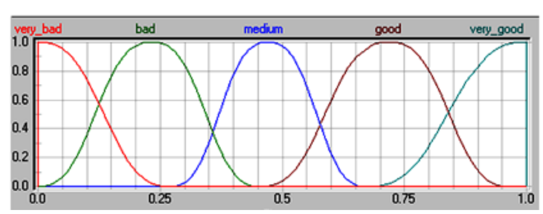

Fig. 18 Membership function of route fitness (y) after supervised learning

6. Conclusions

In this paper we have designed and implemented a GIS-based fuzzy-neural approach for modeling driver's behavior which represents the correlation of the attributes with the driver's route selection.
A recommendation or route fitness is provided to the driver based on a training of the adaptive fuzzy-neural network on the main criteria of route selection such as length, time and the degree of difficulty.
In this context, we have applied an innovative methodology based on modified genetic algorithm to generate training data set as well as a fuzzy adaptive learning control network (FALCON) with hybrid learning to extract fuzzy rules from the generated training data and to train the membership functions.
Major characteristics of this innovative approach are as follows:
1. Generating proposed FALCON training data by GA utilization in a GIS-based approach with the capability of taking and adjusting the "importance" of each route criterion chosen by the driver.
2. Utilizing "range (scale)-independent ranking" in measuring each route criteria as FALCON inputs.
3. Letting driver to specify his preferences using fuzzy rules with some predefined linguistic terms and deriving other rules from the self-organized learning phase of proposed FALCON from generated training data. This fuzzy rules extraction provides a simple representation of complex procedures of driver's decision making and reflects a kind of knowledge which is applied in modeling driver's behavior.
4. Performing better than the purely supervised learning algorithm (e.g., the back-propagation algorithm) because of a priori classification of training data through an overlapping receptive field according to unsupervised learning before the supervised learning.
Further efforts will be made on expanding the algorithm in combination with GA and proposed FALCON in dynamic route selection.

References

[1] P. Pahlavani, M. R. Delavar, and F. Samadzadegan, " The use of a GIS –based genetic algorithm in multi-objective quasi-optimization in urban route selection", Accepted to be published in Journal of University College of Engineering, Engineering Faculty, University of Tehran, ISSN 0803-1026, Issue.5, Vol.40, No.100, 2006
[2] P. Pahlavani, F. Samadzadegan and M. R. Delavar, "A GIS-based approach for urban multi-criteria quasi optimized route guidance by considering unspecified site satisfaction", Lecture Notes in Computer Science, ISSN 0302-9743, Vol. 4197 , pp. 287-303, 2006
[3] D. Teodorovic, S. Kikuchi, "Transportation route choice model using fuzzy inference technique", Paper presented at the 1st Internat. Symp. on: Uncertainty Modeling and Analysis: Fuzzy Reasoning, Probabilistic Models, and Risk Management, University of College Park, MD, pp. 140–145,1990.
[4] T. Lotan, and H. N. Koutsopoulos, "Modeling default behavior in the presence of information and its application to the route choice problem." International Journal of Intelligent Systems, Vol. 14, pp.501-533, 1999.
[5] D. Teodorovic and M. Kalic, "A fuzzy route choice model for air transportation networks." Transport. Plann. Technol, pp. 109–119, 1995.
[6] B. Ran, and D. Boyce, "Modeling Dynamic Transportation Networks." Second Revised Edition, Springer Verlag, Berlin, Germany 1996.
[7] P. J. Bently, "Generic Evolutionary Design of Solid Objects Using a Genetic Algorithm." Ph.D. Thesis, University of Huddersfield, Heddersfield, UK, 1996.
[8] C. T. Lin, and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system", IEEE Trans. Comput. vol. 4, no.12, pp. 1320-1336, 1991
[9] C. T. Lin, and C. S. G. Lee, "Neural Fuzzy Systems: A Neuro-fuzzy synergism to intelligent systems" Prentic-Hall, Inc. New Jersey, USA, 1996.
[10] T. K. Kohonen, "Self-Organization and Associative Memory", 3rd ed. New York: Springer-Velag, 1989.
[11] D. E. Rumelhart, and D. Zipser, "Feature Discovery by Competitive Learning.", In D. E. Rumelhart , J. L. McClelland, et al., eds., Parallel Distributed Processing, Vol. 1, Chap.5. Cambridge, MA: MIT Press, 1986.

Parham Pahlavani, receiving the B.S. degree as a top student in Civil Eng.- Surveying engineering from industrial K.N.Toosi Univ, Tehran, Iran in 2001, and the M.S. degree as a top student in geospatial information system (GIS) engineering from the Tehran University, Tehran, Iran in 2005, being as a GIS researcher in MAS R&D Co. from 2002 to 2006.
His current research interests are using genetic algorithms, neural networks, learning systems, fuzzy control and approximate reasoning in GIS.

Mahmoud Reza Delavar received the BSc. degree in Civil Eng.- Surveying from K.N.Toosi Univ. Iran, in 1989, M.E. in Civil Eng. – Photogrammetry and Remote Sensing from Univ. of Roorkee, India, and a PhD in Geomatic Eng.-GIS from Univ. of New South Wales, Australia in 1997. Since 1998 he has been working in Dept. of Surveying Eng., Eng. Faculty, University of Tehran as an assistant professor. His research interest includes GIS-based artificial intelligence, agent-based spatial modeling, spatial data quality, spatio-temporal GIS, land administration, spatial data infrastructure, disaster management and cadastre. He is president of Iranian Society of Surveying and Geomatic Eng. He is also national representative of Iran in UDMS.