Traffic Signal Control for Isolated Intersections Based on Fuzzy Neural Network and Genetic Algorithm

TAHERE .ROYANI1, JAVAD. HADDADNIA2, MOHAMMAD. ALIPOOR3
(1,2,3)Department of electrical engineering
Tarbiat Moallem University of Sabzevar
Sabzevar
Iran
tahere.royani@gmail.com; Haddania@sttu.ac.ir; m.biomedical@yahoo.com

Abstract: In this paper a fuzzy neural network is applied for real time traffic signal control at an isolated intersection. The FNN has advantages of both fuzzy expert system (fuzzy reasoning) and artificial neural network (self-study). A traffic light controller based on fuzzy neural network can be used for optimum control of fluctuating traffic volumes such as oversaturated or unusual load condition. The objective is to improve the vehicular throughput and minimize delays. The rules of fuzzy logic controller are formulated by following the same protocols that a human operator would use to control the time intervals of the traffic light. For adjusting the parameters of FNN, genetic algorithm was used. Compared with traditional control methods for traffic signal, the proposed FNN algorithm shows better performances and adaptability.

Key-Words: Fuzzy Neural Network, Traffic Control, Delay, Genetic Algorithm, Performance.

1 Introduction
Traffic congestion is a crucial problem in large cities. Signal control methods include traditional control methods and intelligent control methods. Since intelligent control methods are superior to traditional control methods, lots of intelligent signal control models were put forward in recent years. It is partially caused by improper control of traffic lights, which is not corresponding to the current traffic conditions. To alleviate traffic congestion in urban areas, the concept of Intelligent Transportation Systems (ITS) has been widely accepted in developed countries. ITS is a highly promising system for providing key solutions to current road congestion problems [1]. The problem of intelligent traffic control has been studied in the area of ITS for many years. There are many conventional methods for traffic signal control but most of them sometimes fail to deal efficiently with the complex, time-varying traffic conditions and controller can’t satisfy real-time character for traffic signal [2]. They are modeled based on the preset cycle time to change the signal without any analysis of traffic situation. Due to fixed cycle time, such systems do not consider that which intersection has more load of traffic, so should kept green more or should terminate earlier then complete cycle time. Recently, major research on urban traffic focuses on artificial intelligence techniques, such as fuzzy control, genetic algorithm and neural network. Using timed Petri Nets [3], SPSA [4], ant algorithm [5], knowledge based multi-agent system [6, 7], and a mobile agent [8] have also been suggested.

Trabia [9] designed a multi-phase fuzzy logic controller for an isolated intersection with through and left-turning movements. In [10], a new fuzzy controller based on fuzzy logic and weighting coefficients is designed. Bingham [11] obtained intersection fuzzy control parameters from neural networks, and improved fuzzy control result. Chen Xiangjun [12] put forward a self-learning traffic signal control approach, which controls intersection signal with fuzzy algorithm, and updates fuzzy control rules with genetic algorithm. These studies have their own characteristics and theoretical foundations; however, an intersection signal control model should consider three factors: (1) Simplified computing model, control schemes should output in a specified period; (2) consider both under control intersection and its adjacent intersection, for realizing linear or group control; (3) self-learning ability. This paper tries to consider these factors in intersection control model. Through fuzzy classifying traffic flow in under control intersection, the model save signal control schemes in different traffic flow into knowledge-database as rule set. In control process, the model use neural network to update rule set according to different control effect of control schemes in different traffic flow, thus the model has self-learning ability [13]. The rules of fuzzy logic controller are formulated by following the same protocols that a human operator would use to control the time intervals of the traffic light. The length of current green phase is extended or terminated depending upon the "arrival", number of vehicle approaching the green phase.
and “queue” that corresponds to the number of queuing vehicles in red or green phases. By using MATLAB tool for simulation and experiments, it proves that model control effect is obviously superior to traditional control methods. The paper is organized as follows: Section 2 deals with basic of fuzzy logic systems, rule base and membership functions. The fuzzy neural network model for traffic signal control at the isolated intersection and proposed method are explained in section 3. experimented and analyses are shown in section 4. Section 5 describes conclusions

2 Fuzzy logic systems for traffic signal control

Fuzzy logic is a distinct idea for developing models of physical processes. Fuzzy models are less externally complex; they can be understood easily and very much suitable for non-linear processes. Models with fewer rules are more advantageous. Fuzzy controllers have the ability to take decision even with incomplete information. More and more sophisticated fuzzy logic controllers are being developed for traffic control [14-21]. These algorithms are continually improving the safety and efficiency by reducing the waiting delay of vehicles on signals [22]. Fuzzy logic allows linguistic and inexact data to be manipulated as a useful tool in designing signal timings. Also the linguistic control strategy that is decided by “if-then-else” statement can be converted in to a control algorithm using fuzzy logic. The design of a fuzzy signal controller needs an expert knowledge and experience of traffic control in formulating the linguistic protocol, which generates the control input to the traffic signal control system.

The input variables of the fuzzy controller in this paper are designed as follow; one is the number of approaching vehicles in the current green phase (denoted by AVi); the another is the number of queuing vehicles in the current green phase (denoted by qgi) and the third one is the number of queuing vehicles in the current red phase (denoted by qri). Here, i refers to the sequence number of the signal current phase. The output variable is the extended time in the current green phase (expressed by \(\Delta t\)). Suppose for this work that AVi, take on following linguistic values: L(low), M(medium), H(high); and qgi , qri take on the following linguistic values: S(short), M(medium), L(long), VL(very long). \(\Delta t\) take on Z(zero), S(short), M(medium), L(long). The fuzzy sets of input and output variables are shown in Fig. 1. For this inputs we have 48 fuzzy rules that some of them are shown in Table 1.

We know that the maximum of phase extension is 30s and maximum of queue is 90 and Initialize the phases minimal time \(t_{\text{min}}=15\)s.

![Fig.1. membership function of inputs and output](image)

<table>
<thead>
<tr>
<th>Table 1: Sum of Fuzzy Rules of Traffic Control Sys.</th>
</tr>
</thead>
<tbody>
<tr>
<td>If Qg is short and Qr is short and AV is low Then (\Delta t) is short.</td>
</tr>
<tr>
<td>If Qg is long and Qr is medium and AV is medium Then (\Delta t) is short.</td>
</tr>
<tr>
<td>If Qg is medium and Qr is long and AV is low Then (\Delta t) is zero.</td>
</tr>
<tr>
<td>If Qg is medium and Qr is very long and AV is high Then (\Delta t) is zero.</td>
</tr>
<tr>
<td>If Qg is very long and Qr is short and AV is low Then (\Delta t) is long.</td>
</tr>
<tr>
<td>If Qg is short and Qr is very long and AV is high Then (\Delta t) is zero.</td>
</tr>
</tbody>
</table>

3 proposed method

3.1 Fuzzy neural network modeling

Artificial neural network (ANNs) (learning systems) and expert system (knowledge-based systems) have been extensively explored as approaches for decision making. While the ANNs compute decisions by learning from successfully solved examples, the expert systems rely on a knowledge base developed by human reasoning for decision making. An important aspect in intelligent system design is decision explanation, which involves supplying a coherent explanation of its decisions [2]. This is required for 1) acceptability of the solution and 2) correctness of the reasoning process by evaluating the trace generated by the inference engine or by analyzing the rule base (which typically use “IF THEN” rules). Also, in learning system such as ANNs, knowledge is represented in the form of weighted connections, making
decision tracing or extraction difficult. Therefore by using an ANN or an expert system approach to intelligent decision making leads to different levels of performance depending on the model as well as the application. By integrating the two approaches, it is possible to overcome the deficiencies associated with using a single approach. The key properties of neuro-fuzzy systems are the accurate learning and adaptive capabilities of the neural networks, together with the generalization and fast-learning capabilities of fuzzy logic systems [2]. The fuzzy neural network that is used in our work is a five-layer dedicated neural network, as shown in Fig. 2, designed according to the working process of fuzzy controller systems; it was presented in [2]. The relation and functions of the nodes in the network are as follows.

The first layer is input layer; the nodes represent input linguistic variables of queri, quegi, and AVi. So the outputs of first layer can be written as

$$O_1^{(1)} = q_{ri}, O_2^{(1)} = q_{gi}, O_3^{(1)} = AV_i$$

The second layer is membership-function layer. Each node in this layer represents the membership function of a linguistic value associated with an input linguistic variable. The output of each node is in the range [0, 1] and represents the membership grade of the input with respect to the membership function. Here, Gauss function is used to carve up the input signal. The output of each node is:

$$O_{jk}^{(2)} = \exp \left( - \frac{(O_j^{(1)} - a_{jk})^2}{b_{jk}} \right)$$

Where j=1, 2; k=1, 2, 3; 4; and for j=3; k=1, 2, 3; a_{jk} and b_{jk} are parameters of centroid and width respectively. These parameters will be adjusted in back propagation step.

The third layer of the fuzzy controller network corresponds to the rule base. A fuzzy rule is characterized by the relationship between the antecedent and the consequent, and an action that is inferred from a set of fuzzy rules and a fuzzy relation composed from the rules. If A, B and C are fuzzy subsets representing linguistic variables, then a decision from a rule “if A&B then C” is defined by the membership functions on A and B and a node p in the third layer computes the rule firing strength. In our network, the “and” combiner in the rules is interpreted as the minimum operator suggested by Zadeh [23].

Refer to “(3),” p is the number of rules; we have 48 rules in our work.

The forth layer of the fuzzy controller network corresponds to the rule consequents. Initially, the links between the third and forth layer are fully connected so that all the possible fuzzy rules are embedded in the structure of the network. The weight $\alpha_p$ (1≤p≤48) of an input link in the layer represents the certainty factor of a fuzzy rules. Hence, these weights are adjustable while learning the knowledge of fuzzy rules. We choose the max-operator suggested by Zadeh [23] as the function of a node in the layer. With this explanation the outputs of this layer are

$$O_j^{(4)} = \text{Max} (\alpha_p O_p^{(3)}), j=1,...,4$$

The fifth layer is defuzzification layer. The node in this layer represents the output linguistic variable and performs defuzzification, taking into consideration the effects of all the membership functions of the linguistic values of the output. We choose the correlation-product inference and the fuzzy centroid defuzzification scheme, and then the function of the output node is defined as follows:

$$O_j^{(5)} = \frac{\sum_{p=1}^{48} (O_j^{(4)} \alpha_p)}{\sum_{p=1}^{48} (\alpha_p)}$$

Where $a_j$ and $b_j$ are the area and centroid of the membership function of the output linguistic value respectively. Since it is assumed that the membership functions of the output linguistic values are known, the areas and centroids can be calculated before learning.
3.2 Training the FNN network with Genetic Algorithm

The error gradient descent is chosen as the training algorithm. In the training phase, the concept of genetic algorithm is used to minimize the least mean square (LMS) error function:

\[ E(\hat{w}) = \frac{1}{2} \sum_{d \in D} \sum_{k \in \text{outputs}} (t_{kd} - o_{kd})^2 \]

(6)

\(E\) defined as a sum of the squared errors over all the output \(k\) units for all the training examples \(d\).

\[ E_i = \frac{(t_i - O^{(5)})^2}{2} \]

(7)

Refer to “(7),” \(t_i\) is the our target for \(i\)th input, and \(O^{(5)}\) is output of our network. For adjusting parameters in learning process, we use genetic algorithm in stead of backpropagation method, because BP for this network is very complicated and suffer from computational complexity. A genetic algorithm maintains a population of candidate solutions for the problem at hand, and makes it evolve by iteratively applying a set of stochastic operators. Genetic algorithm attempts to minimize the squared error between the network output values and the target values for these outputs. The weight update loop may be iterated thousands of times in a typical application.

Construct the training algorithm:

Step1: Input the parameters of the control algorithm and the sample value to train.
Step2: define squared error as fitness function of GA and FNN weights as its variables.
Step 3: Produce an initial population of individuals
Step 4: Evaluate the fitness of all individuals
Select fitter individuals for reproduction

Recombine between individuals
Mutate individuals
Evaluate the fitness of the modified individuals
Generate a new population

Step5: If termination condition met and the trained error is less than the demanded trained error, then the train of the network is end and the weights are outputted, else return to step4 and continue to train the network.

4 Simulation and results

3.1 Model of an intersection

An isolated signalized intersection with four-legs and three lanes in each leg has is studied in this paper. Fig.3 shows a typical intersection with lanes and vehicle detectors configuration. Each approach has straight movement without left-turning and right-turning. Four cameras for vehicle detection are installed on stop-lines. The cameras provide vehicles’ information and number of vehicles waiting in the lanes. Traffic signal is controlled by two phases. Minimum of cycle is about 40s and maximum of cycle is about 120s.

3.2 Simulation

The simulation is carried out using MATLAB 7.8 and the Genetic Algorithm Toolbox. The Genetic Algorithm Toolbox is useful to train quickly the network and changes are easily made. This significantly reduces the development time of the simulation model. The new fuzzy neural network traffic controller can optimally control traffic flows under both normal and abnormal traffic conditions. The Criterion of optimization is the decrement length of queues and the average of waiting time vehicles in intersection. Our samples are real and extracted from traffic video sequences. Sampling time is 10s. The minimum green time (\(g_{min}\)) is preset as 10 seconds in order to let the vehicles cross the intersection safely; the maximum of the extension of green time (\(e_{max}\)) is 30 seconds. Pedestrians are not considered in this study.

The parameters of this GA are set as: population size=220, crossover probability=0.5, mutation probability=0.01.

The stop condition is \((f_{i+1} - f_i) \leq \varepsilon\) where \(f_i\) is the maximum fitness function among the population for the \(i\)th iterative evolution and \(\varepsilon\) is 0.01.

3.3 Results

For testing our traffic control system, we use it at a simulated intersection in MALAB with real situation. We compare its results with fixed time control system. The results of simulation for Fixed time control and Fuzzy Neural Network control were demonstrated in Fig.4 and Fig.5.
Fig. 5. Table 2 shows the results of waiting time for fixed time control and fuzzy intelligent control for 1000 vehicles in four lanes of intersection.

<table>
<thead>
<tr>
<th>Control Method</th>
<th>Average Waiting Time (second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Time Control</td>
<td>330s</td>
</tr>
<tr>
<td>FNN control</td>
<td>178s</td>
</tr>
</tbody>
</table>

**5 Conclusion**

In this paper we attempted to apply the Fuzzy Neural Network model to traffic signal controller with expert knowledge. The fuzzy neural network is tuned by the proposed learning method. We use genetic algorithm for learning process. By the learning of the neural network, we can tune the fuzzy model and optimize system’s parameters. The research results have proved feasibility and validity of the proposed FNN algorithm.

References:


