Fuzzy Group-Based Intersection Control via Vehicular Networks for Smart Transportations

Jialang Cheng, Weigang Wu, *Member, IEEE*, Jiannong Cao, *Fellow, IEEE*, and Keqin Li, *Fellow, IEEE*

***Abstract*—Vehicular network has been recently used to achieve high efficient and flexible traffic scheduling at intersection roads for smart transportation systems. Different from existing works, where traffic signal is used to schedule waiting vehicles at each lane, we propose to divide vehicles in the same lane into small groups and schedule vehicle groups via wireless communication rather than traffic lights. Such direct scheduling of vehicles can reduce waiting time and improve fairness, especially when the traffic volume in different lanes is imbalanced. The key challenge in such a design lies in determining appropriate size of groups with respect to real-time traffic conditions. To cope with this issue, we propose a neuro-fuzzy network-based grouping mechanism, where the network is trained using reinforcement learning technique. Also, vehicle groups are scheduled via a neuro-fuzzy network. Simulations using ns3 are conducted to evaluate the performance of our algorithm and compare it with similar works. The results show that our algorithm can reduce waiting time and at the same time improve fairness in various cases, and the advantage against traffic light algorithms can be up to 40%.**

***Index Terms*—Fuzzy neural networks, intelligent transportation system (ITS), intersection control, machine learning, Vehicular Ad hoc NETworks (VANETs).**

I. Introduction

VEHICULAR networks, especially Vehicular Ad hoc NETwork (VANET) [8], [16], have been recently considered in intelligent transportation systems (ITS) [15] to achieve high accuracy, efficiency, and flexibility. Among others, intersection control has been always a key issue in ITS for the construction of smart cities. The key point is how to schedule traffic signal efficiently according to traffic volume information so as to reduce waiting time and improve fairness. Most existing works on intersection control adopt traffic light-based approach, including fixed rotation of green light [17], and traffic detector-based adaptive scheduling [3], [18].

VANET-based intersection control is a new approach developed recently [5], [7]. With VANETs, one vehicle can communicate with other vehicles (V2V) or infrastructures (V2I) via wireless links. Then, mobility state information of individual vehicles, e.g., id, speed, and position, can be collected and integrated into traffic signal scheduling. Therefore, VANET-based intersection control algorithms are more flexible and efficient than detector-based ones.

Our algorithm is also VANET based, where V2I communications are used to collect vehicles' mobility information. However, different from existing works, which schedule vehicles indirectly via traffic lights, we schedule vehicles directly via wireless communications between the controller and vehicles. More importantly, we propose to group waiting vehicles in the same lane and scheduling them in the granularity of group.

Our design has two major advantages against existing algorithms. First, group-based scheduling reduces average waiting time (AWT), especially when traffic flows in concurrent lanes are imbalanced or the flow varies largely from time to time. By grouping vehicles dynamically in a real-time way, vehicles in concurrent lanes are divided into groups with similar size, and more concurrent passing is enabled so as to improve system efficiency. Second, group-based scheduling improves fairness. With groups, vehicles arriving much later than previous vehicles in the same lane will be divided into a new group and not scheduled together with previous vehicles.

However, grouping vehicles is not a trivial task due to the dynamics of traffic conditions. To achieve high efficiency, groups at concurrent lanes should have similar size. On the other hand, vehicles with much different arrival time should be grouped into different groups so as to improve fairness. Therefore, grouping must be done with various factors considered in real-time way. To cope with such challenges, we adopt the approach of neuro-fuzzy control [1], which combines the advantage of fuzzy logic and neural network. Since the desired output is usually unavailable in traffic control, network training before deployment is impossible. Therefore, we use reinforcement learning [13] to adjust the parameters in the neural network, which does not require a training procedure.

The rest of this paper is organized as follows. Section II reviews existing intersection control algorithms, especially those based on VANETs. The system model assumed is presented in

Manuscript received December 26, 2015; revised June 16, 2016; accepted July 2, 2016. Date of publication July 11, 2016; date of current version April 18, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61379157, in part by Guangdong Frontier and Key Technology Innovation Fund under Grant 2015B010111001, in part by Program of Science and Technology of Guangdong under Grant 2015A010103007, and in part by Program of Science and Technology of Guangzhou under Grant 201510010068. Paper no. TII-15-1858. *(Corresponding author: Weigang Wu.)*

J. Cheng and W. Wu are with the School of Data and Computer Science, Sun Yat-sen University and Guangdong Province Key Laboratory of Big Data Analysis and Processing, Guangzhou 510006, China (e-mail: chjlang@mail2.sysu.edu.cn; wuweig@mail.sysu.edu.cn).

J. Cao is with the Department of Computing, Hong Kong Polytechnic University, Hung Hom, Hong Kong (e-mail: csjcao@comp.polyu.edu.hk).

K. Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2016.2590302

Section III. Section IV describes our intersection control algorithm, including system architecture, fuzzy rules, and detailed operations in grouping and scheduling. Performance evaluation is reported in Section V. Finally, Section VI concludes this paper with future directions.

II. Related Work

Most existing works on intersection control are traffic signal based [3], [17] and the key issue is to determine a good signal scheduling plan. Signal scheduling can be modeled as a combinatorial optimization problem and the optimal scheduling plan can be calculated via various methods such as branch-and-bound [11] and linear programming [12]. Unfortunately, due to the dynamics of traffic load, traffic control systems are large complex nonlinear stochastic systems, so determining the optimal time of green light is very hard even if not impossible.

To cope with the complexity of traffic dynamics, mathematic models and computational intelligence [17] have been widely used in traffic signal scheduling and many algorithms have been proposed, including genetic-based algorithms [10], fuzzy logic-based algorithms [14], neuro-fuzzy-based algorithm [2], and machine learning-based algorithms [13]. These algorithms focus on how to reduce the waiting time upon real-time traffic volume information, with either an isolated intersection or network of intersections considered. VANET has also been used in traffic signal scheduling to collect detailed vehicle information, including id, speed, and position. With such information, accurate and efficient scheduling can be achieved [7], [13].

Besides traffic signal scheduling, autonomous vehicle controlling via agent [4] or maneuver manipulation [6], [9] have also been studied. Without using a traffic signal, such approaches calculate the optimal trajectory for each vehicle so that vehicles can safely pass the intersection without colliding with each other. Since the speed and position of each vehicle need to be accurately calculated, the optimization is very complex, especially when the number of vehicles is large.

Our traffic control system also assumes autonomous vehicles. However, different from the existing works, our algorithm adopts neuro-fuzzy network to schedule vehicles and optimize scheduling strategy. With the powerful reasoning and learning ability of neuro-fuzzy network, our traffic control system is more efficient and more adaptive to real-time traffic condition.

III. System Model

A. Intersection and Lanes

We consider a typical intersection with four directions, i.e., north, south, east, and west, as shown in Fig. 1. In each direction, there are two lanes, for going forward and turning left, respectively. Obviously, the path of a vehicle in the intersection area is determined by the lane it is in.

The small dashed rectangle represents the core area of the intersection. A vehicle in this area is called to be "passing" the intersection. The large dashed rectangle represents the queue area. A vehicle in this area is viewed as in the waiting queue to pass the intersection.[Comp: Please set the below definition as per style of theorem.]

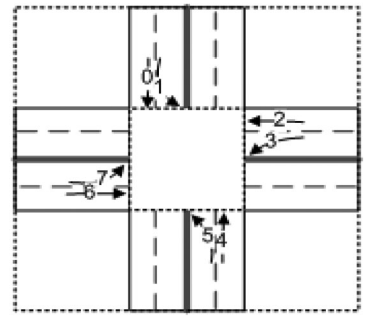


Fig. 1. Intersection model.

Definition 1: [*The concurrency/conflict relationship*]. According to road rules, vehicles with crossing paths have to pass the intersection mutual exclusively. Such vehicles are said to be "conflicting." Accordingly, the lanes of conflicting vehicles are also conflicting with each other. On the other hand, vehicles with noncrossing paths can pass the intersection simultaneously. These vehicles and their lanes are to be "concurrent." In Fig. 1, each lane has two concurrent lanes, e.g., l_0, l_1, l_4 are concurrent.

B. Vehicles and Controller

Each vehicle has a unique id, which can be the license plate number. A vehicle can get the knowledge of its lane number based on the digital map or other methods. It is assumed to be able to detect the boundary of the queue/core area when it crosses the boundary. (This can be realized via on road sensors or positioning system like GPS.) Each vehicle is equipped with anticollision or collision prevention component, so that emergent braking is conducted automatically in case that vehicles scheduling encounters failures.

There is an intersection controller deployed in the center (or vicinity) of the intersection. Both vehicles and the controller are equipped with wireless communication device and they can communicate with each other as defined in wireless access in vehicular environments/dedicated short range communications standards.

We assume that the transmission range of the communicating device is larger than the length of the queue area. That is, the vehicles inside the queue area constitute a one-hop network and each vehicle can communicate with the controller directly. Wireless links are reliable and no packets will be lost.

IV. Proposed Traffic Control System

A. System Architecture

Our neuro-fuzzy traffic control system is shown in Fig. 2. It consists of two parts: vehicle grouping and group scheduling. The first part is in charge of dividing vehicles into groups. Since vehicles arrive at the intersection sequentially, grouping is done in an incremental way. That is, when a vehicle arrives, the controller makes a decision, according to fuzzy rules, on whether this vehicle should be included into the end group of the lane. The group scheduling part is used to schedule vehicles to pass the intersection based on fuzzy rules, in terms of vehicle groups,

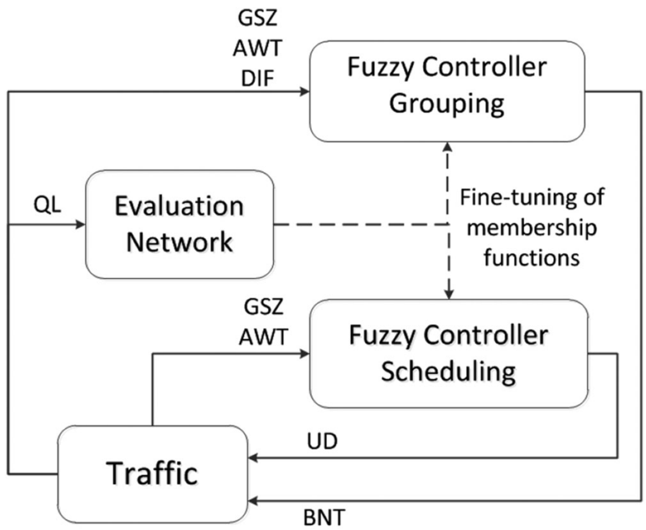


Fig. 2. Architecture of the neuro-fuzzy traffic control system.

i.e., the permit of passing intersection is always granted to the head group of a lane.

There are three neural networks in our system. The evaluation network is a two-layer feedforward neural network which gathers information about vehicles in the intersection and evaluates the performance actions selected by fuzzy controller networks at last time step. The learning algorithm used in our system mostly follows GARIC [1].

Both grouping and scheduling adopt a five-layer feedforward neural fuzzy network, as shown in Fig. 2. According to the output of evaluation network, these fuzzy controller networks fine-tune fuzzy membership functions by updating their weight parameters. At the same time, the evaluation network adjusts its weight parameters.

B. Evaluation Network

The evaluation network is used to evaluate the goodness of the action of fuzzy controller. It is a standard two-layer feedforward network with h hidden layer cells and n input cells from the environment. Each input cell measures the queue length of waiting vehicle at each lane. Each hidden layer cell collects weighted inputs from the first layer and computes activated output using a sigmoidal function:

$$y_i = g\left(\sum_{j=1}^{n} a_{\mathrm{ji}}{}^{*} QL_j\right) \tag{1}$$

where

$$g(s) = \frac{1}{1+\ e^{-s}}\ . \tag{2}$$

The output layer of the evaluation network receives input values from both the input layer and the hidden layer. The output v is a measurement of the goodness of the network, i.e., prediction of future reinforcement [1]

$$v = \sum_{i=1}^{n} b_i{}^{*} QL_i + \sum_{j=1}^{h} c_j{}^{*}\ y_j. \tag{3}$$

The prediction of future reinforcement is combined with external performance measure to compute internal reinforcement $\hat{r}$:

$$\hat{r}(t) =\ r(t) + \gamma v(t) - v(t-1)\,. \tag{4}$$

In (4), $r(t)$ is the change in AWT between two successive learning cycles, and $\Upsilon (0 \le \Upsilon \le 1)$ indicates the discount rate to set less significance on v at time t than that at the previous time step. The internal reinforcement is used to guide the fuzzy controller network in decision making. For example, if the system moves from a state with low v to a state with high v, the positive change can reinforce the selection of the action that caused this move.

Learning in evaluation network adopts the gradient descent algorithm, as in common neural networks. If a positive (negative) internal reinforcement is received, network weights are rewarded (punished) by changes in the direction that increases (decreases) its contribution to the total sum. The weights of the links connecting input and output are updated according to the following:

$$b_i\,[t+1] =\ b_i\,[t] +\ \beta \hat{r}\,[t+1]\, QL_i\,[t] \tag{5}$$

where $\beta = 0.1$ is a constant and $\hat{r}[t+1]$ the internal reinforcement at time $t+1$.

The weights of the connections between the hidden layer cells and the output cell are updated as follows:

$$c_i\,[t+1] =\ c_i\,[t] + \beta \hat{r}\,[t+1]\, y_i\,[t]\ \ . \tag{6}$$

The weights of the connections between input and hidden:

$$\begin{aligned} a_{\mathrm{ij}}\,[t+1] \ =\ a_{\mathrm{ij}}\,[t] +\ \beta_h \hat{r}\,[t+1]\, y_i\,[t]\ \ (1 -\ y_i\,[t]) \\ \operatorname{sgn}(c_i\,[t])\, QL_i\,[t] \end{aligned} \tag{7}$$

where $\beta_h 0.3$ and *sgn()* is a sign function.

C. Vehicle Grouping

The goal of intersection control is to reduce AWT and improve fairness. Then, vehicle groups should have the following properties.

1) Groups at concurrent lanes should have similar size. This can improve the utility of intersection space and reduce AWT.
2) The waiting time of vehicles in the same group should be similar. This can help achieve high fairness.

To deal with the complexity and variation of traffic volume at intersections, we adopt a neuro-fuzzy network to grouping vehicles, as shown in Fig. 3. By updating the weight parameters through reinforcement learning, the neuro-fuzzy network can fit various traffic conditions.

1) Variables and Membership Functions: In vehicle grouping, three fuzzy variables are used as input, which reflect the current traffic condition at the intersection.

1) Group size (GSZ): the number of vehicles currently in the ending group.
2) AWT: the AWT of the vehicles in the ending group.
3) Difference from concurrent groups (DIF): the difference in size between the current group and its concurrent groups.

The output of the fuzzy logic is benefit.

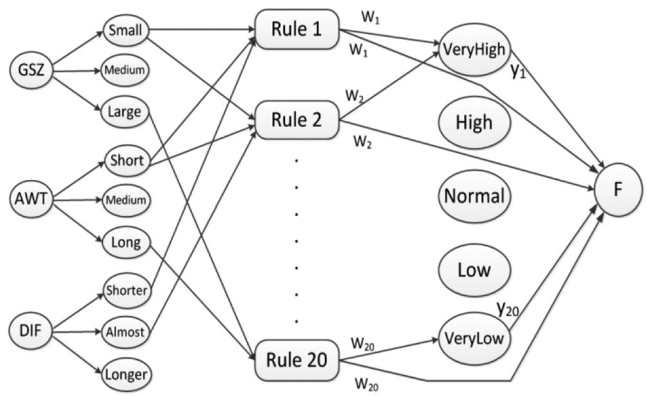


Fig. 3. Neuro-fuzzy network for grouping.

4) Benefit (BNT): the benefit of including the current vehicle into the ending group. Here, "benefit" refers to benefit may be obtained in efficiency and fairness.

As shown in Fig. 3, each of the input variables has three linguistic values, and the output fuzzy variable has five linguistic values. The membership functions of all these linguistic values are of the form of triangular function. By fine-tuning the position of three corners of the triangular, the shape and location of the membership function may change.

When vehicle i enters the queue area of intersection, it sends a request message to the controller node. The request message carries two data items: vehicle id and lane id. Upon receiving the request message from vehicle i, the controller needs to decide on which group i should be included into, based on the current traffic status of the whole intersection.

First, the traffic controller examines current traffic condition and gets measurement of three fuzzy input variables, i.e., GSZ, AWT, and DIF. Then, the controller computes BNT through neuro-fuzzy network with predefined fuzzy rule base. If the output value is higher than a threshold value, the new vehicle will be included into the ending group. Otherwise, a new group is established for that vehicle. The fuzzy rule base is described in the following section.

2) Fuzzy Rules Base and Fuzzy Inference: Table I shows the fuzzy rule base for grouping, which infer the benefit of including a newly arrived vehicle into the ending group. The underlying idea in designing these rules is as follows.

1) Groups in concurrent lanes should have similar sizes.
2) Vehicles in the same group should have similar waiting time, but different groups should have quite different AWT.

For example, Rule 1: "if GSZ is Small and AWT is Short and DIF is Smaller, then BNT is VeryHigh" indicates to increase the current group so as to reduce the difference in GSZ. At the same time, since AWT is short, the vehicles in the current group arrived not a long time ago, and including the newly arrived vehicle into the group will not affect fairness much. Therefore, in such a case, the newly arrived vehicle should join the current group rather than create new group.

Fuzzy inference is applied to combine these rules into a mapping from fuzzy input set to fuzzy output set. As described in

TABLE I
FUZZY RULES FOR GROUPING

No.	Input		Output	
	GSZ	*AWT*	*DIF*	*BNT*
1	Small	Short	Smaller	VeryHigh
2	Small	Short	Almost	VeryHigh
3	Small	Medium	Smaller	VeryHigh
4	Small	Medium	Almost	VeryHigh
5	Small	Long	Smaller	VeryHigh
6	Small	Long	Almost	VeryHigh
7	Medium	Short	Smaller	VeryHigh
8	Medium	Short	Almost	Normal
9	Medium	Short	Longer	Low
10	Medium	Medium	Smaller	High
11	Medium	Medium	Almost	Normal
12	Medium	Medium	Longer	Low
13	Medium	Long	Smaller	High
14	Medium	Long	Almost	Normal
15	Medium	Long	Longer	Low
16	Large	Short	Almost	Low
17	Large	Short	Longer	Low
18	Large	Medium	Almost	Low
19	Large	Medium	Longer	VeryLow
20	Large	Long	Any	VeryLow

Section III, for each rule in the rule base, cell in layer 3 computes the firing strength by combining all the membership degrees of antecedent labels in the rule through softmin operation. For the consequent part, cell in layer 4 computes the defuzzified value according to the firing strength supplied to it. Local mean-of-maximum is used as the defuzzification method. Finally, the output is the sum of all the defuzzified values, weighted by rule firing strength values.

3) Learning in Vehicle Grouping: As mentioned earlier, we adopt reinforcement learning as the learning algorithm of our neuro-fuzzy network for vehicle grouping. In our system, the output of evaluation network v is a measurement of the performance of our system. Thus, the goal of vehicle grouping is to maximize v. The action taken by the neuro-fuzzy network can be denoted as $F_p(x)$, where x is system input and $\boldsymbol{p}$ is the vector of weight parameters of the network, i.e., the weight parameters on the connection between layers 1 and 2 and those between layers 3 and 4. Hence, the objective of learning is fine-tuning $\boldsymbol{p}$ so as to maximize v. The can be done by gradient descent, which estimates the derivative $\partial v/\partial \mathrm{p}$, and uses the following learning rule to update the parameter value

$$p_{\mathrm{new}} = p + \eta \frac{\partial v}{\partial p} = p + \eta \frac{\partial v}{\partial F} \frac{\partial F}{\partial p}. \tag{8}$$

The dependence of v on F is indirect, because both of them are state specific. Since $\partial v/\partial \mathrm{F}$ is hard to compute, it is approximated as follows, where *sgn()* is a sign function

$$\frac{\partial v}{\partial F} = \mathrm{sgn}\left(\frac{v(t) - v(t-1)}{F(t) - F(t-1)}\right). \tag{9}$$

Since F is known and differentiable, $\partial F/\partial \mathrm{p}$ is much easier to compute. In the following formula, $\mathrm{Con}(R_j)$ and $\mathrm{Ant}(R_j)$ are the consequent and antecedent labels used by rule j. A label V is parameterized by p_v, which represents any one of the parameters of the membership function of that label. F is a weighted sum of all individual rule output as follows:

$$F = \frac{\sum_i w_i f_i}{\sum_i w_i}. \tag{10}$$

where f_i and w_i denote the output and firing strength of rule *I*, respectively.

As discussed in Section III, several rules may use the same linguistic value as their consequent label. For consequent labels *V* with parameter p_v, all rules *i* which use *V* in their consequent part have to be taken account:

$$\begin{aligned}\frac{\partial F}{\partial p_v} &= \sum_{v\, \in \,\mathrm{Con}(R_i)} \frac{\partial F}{\partial f_i}\frac{\partial f_i}{\partial p_v} \\ &= \frac{1}{\sum_{j=1}^{20} w_j} \sum_{v\, \in\, \mathrm{Con}(R_i)} w_i \frac{\partial f_i}{\partial p_v}. \end{aligned}\tag{11}$$

In (11), $\sum_{j=1}^{20} w_j$ is the summation of firing strength of all the rules in rule base and $\partial f_i/\partial \mathrm{p_v}$ is based on the form of membership function and the defuzzification method.

For antecedent labels, the calculation is similar but it requires a few more pass of the derivative of chain rule. The action depends on the degrees w_i, which in turn depend on the membership degree μ_i generated in layer 2

$$\frac{\partial F}{\partial p_v} = \frac{\partial F}{\partial \mu_v}\frac{\partial \mu_v}{\partial p_v} = \left(\sum_{v\, \in\, \mathrm{Ant}(R_i)} \frac{\partial F}{\partial w_i}\frac{\partial w_i}{\partial \mu_v}\right)\frac{\partial \mu_v}{\partial p_v} \tag{12}$$

$$\frac{\partial F}{\partial w_i} = \frac{f_i + w_i \frac{\partial f_i}{\partial w_i} - F}{\sum_{j=1}^{20} w_j}. \tag{13}$$

Similarly, $\sum_{j=1}^{20} w_j$ is the sum of firing strength of all the rules in rule base. $\partial f_i/\partial \mathrm{w_i}$ depends on defuzzification method and $\partial \mu_v/\partial \mathrm{p_v}$ is based on the form of membership function. Since we use softmin operation to get firing strength, $\partial w_i/\partial \mu_\mathrm{v}$ can be computed as follows:

$$\frac{\partial w_i}{\partial \mu_v} = \frac{e^{-k\mu_v}\left(1 + k\left(w_i - \mu_v\right)\right)}{\sum_i e^{-k\mu_i}} \tag{14}$$

where $\sum_i e^{-k\mu_i}$ is calculated by going through membership degrees of all the antecedent linguistic labels in rule *i*.

D. Group Scheduling

The group scheduling of our algorithm is used to grant permit of passing intersection. The pseudocode of the algorithm is listed as Algorithm 1. Obviously, only the head groups at different lanes are candidates for the next passing. The metric of urgency degree (UD) is used to select next group.

The algorithm first computes UD of each head group using fuzzy logic and the head group with the highest UD is selected as the next to pass. The UDs of two concurrent groups are then compared and the one with higher degree is also granted with permit.

Two fuzzy variables are used as input in group scheduling, i.e., GSZ and AWT. The output is UD. The fuzzy rule base is shown in Table II. The structure and learning algorithm for the neuro-fuzzy is similar to that of vehicle grouping.

After the next group and its concurrent group are selected, the controller will broadcast a PERMIT message to vehicles. This message contains the list of two groups of vehicles being granted permit, denoted by *plt*.

When the vehicles receive the PERMIT message, those in *plt* will start to pass the intersection. Of course, vehicles in the same group need to pass intersection one by one. The list *plt* also shows which vehicle is the last one in the group.

Then, when the last vehicle in the granted group exits from the intersection, it will send an EXIT message to the controller. After the controller receives EXIT from the last vehicle in each group granted, one pass is completed and the next group will be scheduled. In this way, the controller is capable of adapting the traffic signal phase sequentially according to the changing traffic flow condition.

ALGORITHM 1:
GROUP SCHEDULING.

For Vehicles

```
//For each vehicle i
On Receiving PERMIT(plt):
if (i ∈ plt)
    move and pass the intersection;
On exiting form the intersection
if (i is the last in the group of plt)
    send EXIT(i, lid) to the controller;
```

For the controller

```
On Receiving EXIT(i, lid):
delete group of i from plt;
if (plt is empty)
    go to ScheduleNext;
ScheduleNext:
for (each head group)
    compute UD using fuzzy logic;
select the head group with highest UD
select the concurrent group with higher UD
construct plt;
broadcast PERMIT(plt);
```

TABLE II
FUZZY RULE BASE FOR UD

No.	Input		Output
	GSZ	*AWT*	*UD*
1	Small	Short	VeryLow
2	Small	Medium	Low
3	Small	Long	High
4	Medium	Short	Low
5	Medium	Medium	Normal
6	Medium	Long	VeryHigh
7	Large	Short	Normal
8	Large	Medium	High
9	Large	Long	VeryHigh

V. PERFORMANCE EVALUATION

We examined the performance of our proposed algorithm by simulations. Since our approach is based on VANET, we conduct

TABLE III
FUZZY RULE BASE FOR UD

Parameters	Values
Territory of Intersection	100 m $*$ 100 m
Transmission range	120 m
Communication protocol (MAC)	IEEE 802.11 p
Time of passing core area	3 s
Capacity of intersection	64 vehicles/min
Volume-to-capacity ratio (v/c)	0.5–0.9
Simulation time	18 000 s

simulations using the popular network simulator ns3. In our simulation, we implemented four different algorithms: 1) the proposed neuro-fuzzy group-based algorithm with reinforcement learning (FuzzyGroupLearning); 2) group-based control algorithm without learning (FuzzyGroup); 3) fuzzy logic-based algorithm without grouping (NoGroup); and 4) an adaptive traffic light control algorithm (AdaptiveLight) [18]. In the AdaptiveLight algorithm, the traffic light facility is simulated by a centralized control node and the permit to pass intersection is scheduled according to green signal time assigned. We choose this algorithm as the baseline because it is more efficient than other detection-based ones. We do not simulate intelligent algorithms because they assume quite different models from our work and hard to implement.

A. Simulation Setup

We simulated an intersection with eight lanes as in Fig. 1, with IEEE 802.11 p as the communication protocol. The major parameters involved are listed in Table III. The area of intersection is set to be 100 m $\times$ 100 m. The transmission range of the communication device is set to be 120 m. All the vehicles in intersection area can communicate with each other directly. The control node is deployed in the center of the intersection. The simulation time is set to 5 h, which is long enough to show the difference in performance of the four algorithms.

Same as in existing works, we vary the volume-to-capacity ratio v/c of each road to examine performance under different traffic load levels, and the capacity of the intersection is set to be 64 vehicles/min. The v/c value in our simulations is ranged from 0.5 to 0.9, a quite large range of traffic load.

Besides traffic load level, we also set two different traffic patterns based on the observation of real world traffic. The first one is the uniform pattern, where all the lanes have the same traffic load level. This pattern is the simplest and popular in existing works. The second pattern is backbone road pattern, where the horizontal (or west–east) road is 50% higher than the vertical one. Such an imbalanced pattern is more complex than the uniform one.

Moreover, to make the traffic more reasonable, vehicles arrive in a random way, following the Poisson distribution, with the mean value set according to different traffic patterns.

B. Simulation Results

Following existing works, performance of traffic control algorithms is measured using two metrics for efficiency and fairness, respectively.

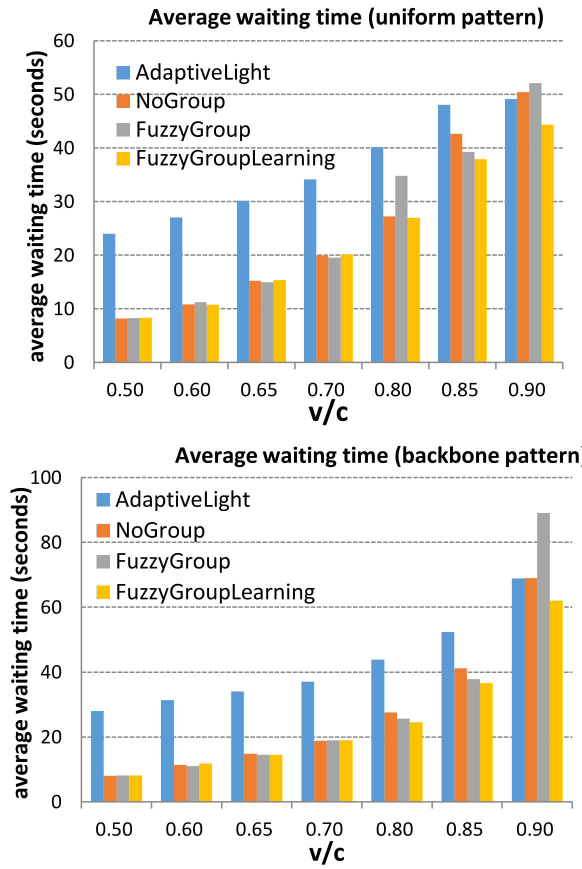


Fig. 4. Average waiting time

Fig. 4. AWT.

1) AWT: the average time duration from the arrival a vehicle to the moment it gets permit to pass intersection.
2) Waiting time variance (WTV): the variance of waiting time of vehicles. This metric is used to show the fairness of the traffic control algorithm.

1) AWT:

The results of AWT under different traffic load levels and traffic patterns are shown in Fig. 4. Roughly, a vehicle needs to wait for tens of seconds to pass the intersection. The waiting time increases with the increase of traffic load. This is expected. High traffic load will certainly cause more vehicles to wait for passing and then longer waiting time. Traffic pattern also affects the value of AWT significantly. Comparing the two figures in Fig. 4, we can see AWT in uniform pattern is smaller than backbone pattern. This indicates that nonuniform pattern is complex and difficult to handle.

Now, let us compare the four algorithms. In most cases, AdaptiveLight performs the worst, while the fuzzy group with learning algorithm is the best. This shows the benefit of VANET-based approaches. AdaptiveLight estimates traffic volume based on data from detectors. With VANET, accurate traffic volume data rather than estimation can be obtained and better scheduling is done.

However, AdaptiveLight is not always the worst. The performance difference among different algorithms is significantly affected by traffic volume level. More precisely, under low traffic levels, the disadvantage of AdaptiveLight is much more obvious than that under high traffic levels. This can be explained as follows. Under low traffic volumes, the error of the volume estimation in AdaptiveLight is large, so the scheduling of AdaptiveLight is inefficient. With the vehicle arrival rate increasing, the accuracy of volume estimation also increases and the difference between AdaptiveLight and other algorithm is reduced.

More interestingly, when v/c reaches 0.9, the performance of AdaptiveLight is even better than FuzzyGroup and NoGroup. This indicates that, under heavy traffic load, predefined membership functions for scheduling in FuzzyGroup and NoGroup cannot work efficiently. On the other hand, the FuzzyGroupLearning algorithm always outperforms others, which confirms the benefit of adaptive membership functions.

Comparing FuzzyGroup with NoGroup, we can see that FuzzyGroup works better than NoGroup in most cases. The difference between these two algorithms becomes more obvious under high traffic load levels. With more vehicles arrive, the waiting queue becomes longer. The proposed grouping algorithm divides the vehicles into different groups so that groups at concurrent lanes have similar size. As discussed in Section IV, similar GSZ at concurrent lanes improves the utility of intersection space and reduces AWT. The simulation result clearly validates the benefit of grouping vehicles. More importantly, FuzzyGroupLearning fine tunes the membership functions of fuzzy controllers according to real-time traffic condition. This adaptive grouping ability increases the efficiency of a traffic controller. This is why, FuzzyGroupLearning performs better than other scheduling algorithms.

2) WTV (Fairness): The results of WTV, i.e., fairness are plotted in Fig. 5. Same as AWT, WTV also increases with the increase of traffic load level in all traffic patterns. This is also expected and easy to understand. The effect of traffic pattern is also obvious. AWT in backbone pattern is larger than that in uniform patterns. Because the traffic volume in backbone roads is much higher, vehicles in these roads have to wait longer, resulting in larger WTV.

Like AWT, WTV of NoGroup and FuzzyGroup is better than AdaptiveLight under low traffic volume. With grouping, vehicles with similar waiting time are schedule together, so the variance of waiting time is much smaller. However, under higher traffic volume, AdaptiveLight outperforms NoGroup and FuzzyGroup. The reason for this is similar: predefined membership functions are not suitable under various traffic conditions. With reinforcement learning, FuzzyGroupLearning adapts its grouping and scheduling strategies to deal with real-time traffic condition. The WTV of FuzzyGroupLearning is smaller than other algorithms, which clearly shows the effectiveness of adaptive grouping and scheduling in fairness.

3) Summary of Simulations: The simulation results have shown that our proposed neuro-fuzzy group-based intersection control algorithm is effcient and fair. It outperforms AdpativeLight, NoGroup, and FuzzyGroup in all cases of traffic levels and traffic patterns. Especically, the difference between NoGroup and our algorithm directly show the effectiveness of vehicle grouping.

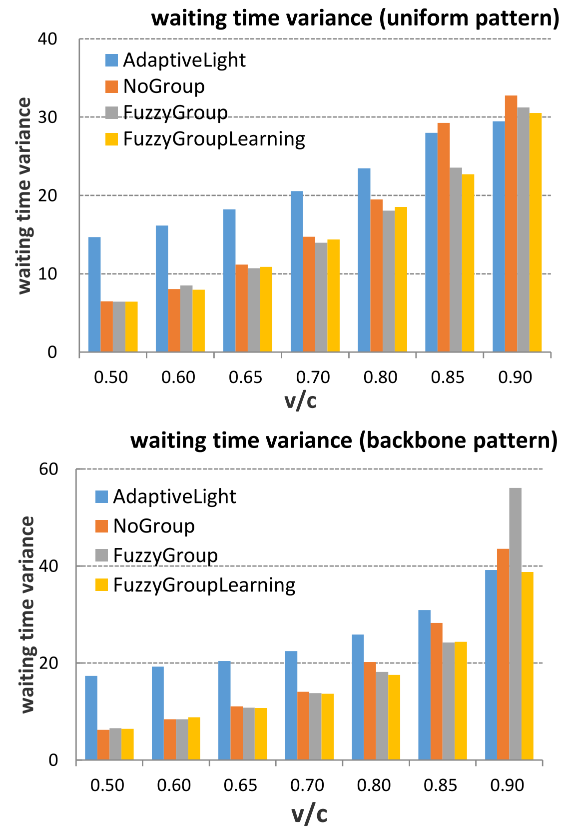


Fig. 5. WTV.

VI. Conclusion

In this paper, we study the intersection control problem for smart transportation systems in smart cities. Different from existing works, we propose to divide the waiting vehicles in a lane into different groups. The permission of passing intersection is then granted in terms of vehicle groups, via V2I communications. The major challenge lies in determining the appropriate groups based on real-time traffic conditions, with respect to requirements of AWT and fairness. We adopt neuro-fuzzy network to do the grouping and also the group scheduling. Furthermore, we apply reinforcement learning to fine-tuning the parameters of the network and make it adaptive to various traffic conditions. Grouping vehicles makes our intersection control algorithm efficient and fair, especially in high dynamic traffic flows. Such advantages have been validated via extensive simulations.

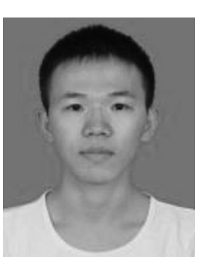
Group-based intersection control is a novel approach and more efforts need to be made for better solutions. Possible directions include improving the grouping fuzzy rules with other machine learning techniques, considering more complex scenarios of multiple intersections.

References

[1] H. R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 724–740, Sep. 1992.

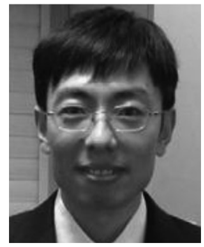
[2] E. Bingham, "Reinforcement learning in neurofuzzy traffic signal control," *Eur. J. Oper. Res.*, vol. 131, no. 2, pp. 232–241, 2001.

[3] K. Y. Chan, S. Khadem, T. S. Dillon, and V. Palade, "Selection of significant on-road sensor data for short-term traffic flow forecasting using the Taguchi method," *IEEE Trans. Ind. Informat.*, vol. 8, no. 2, pp. 255–266, May 2012.
[4] K. M. Dresner and P. A. Stone, "A multiagent approach to autonomous intersection management," *J. Artif. Intell. Res.*, vol. 31, pp. 591–656, 2008.
[5] M. Ferreira and P. M. d'Orey, "On the impact of virtual traffic lights on carbon emissions mitigation," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 284–295, Jan. 2012.
[6] S. Glaser B. Vanholme, S. Mammar, D. Gruyer, and L. Nouveliere, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 589–606, Sep. 2010.
[7] V. Gradinescu, C. Gorgorin, R. Diaconescu, V. Cristea, and L. Iftode, "Adaptive traffic lights using car-to-car communication," in *Proc. IEEE 65th Veh. Technol. Conf.*, 2007, pp. 21–25.
[8] S. E. Li, S. Xu, X. Huang, B. Cheng, and H. Peng, "Eco-departure of connected vehicles with V2X communication at signalized intersections," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5439–5449, Dec. 2015.
[9] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 81–90, Mar. 2012.
[10] P. Lertworawanich, M. Kuwahara, and M. Miska, "A new multiobjective signal optimization for oversaturated networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 967–976, Dec. 2011.
[11] C. Li and S. Shimamoto, "An open traffic light control model for reducing vehicles' emissions based on ETC vehicles," *IEEE Trans. Veh. Technol.*, vol. 61, no. 1, pp. 97–110, Jan. 2012.
[12] W. H. Lin and C. Wang, "An enhanced 0-1 mixed-integer LP formulation for traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 238–245, Dec. 2004.
[13] L. A. Prashanth and S. Bhatnagar, "Reinforcement learning with function approximation for traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 412–421, Jun. 2011.
[14] J. Qiao, N. Yang, and J. Gao, "Two-stage fuzzy logic controller for signalized intersection," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 1, pp. 178–184, Jan. 2011.
[15] F. Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.
[16] W. Wu, J. Zhang, A. Luo, and J. Cao, "Distributed mutual exclusion algorithms for intersection traffic control," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 1, pp. 65–74, Jan. 2015.
[17] D. Zhao, Y. Dai, and Z. Zhang, "Computational intelligence in urban traffic signal control: A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 485–494, Jul. 2012.
[18] B. Zhou, J. Cao, X. Zeng, and H. Wu, "Adaptive traffic light control in wireless sensor network-based intelligent transportation system," in *Proc. IEEE 72nd Veh. Technol. Conf.*, 2010, pp. 1–5.



Jialang Cheng received the B.Sc. degree in electronic engineering and the M.Sc. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2012 and 2015, respectively.

His research interests include distributed system, vehicular network and its application in intelligent transportation system.



Weigang Wu (M'07) received the B.Sc. and M.Sc. degrees in computer science from Xi'an Jiaotong University, Xi'an, China, in 1998 and 2003, respectively, and the Ph.D. degree in computer science from Hong Kong Polytechnic University, Hung Hom, Hong Kong, in 2007.

He is currently a Professor at Sun Yat-sen University, Guangzhou, China. His research interests include distributed systems and wireless networks, especially cloud computing platforms and ad hoc networks. He has published more than 60 papers in conferences and journals.

Dr. Wu has served as a member of editorial board of two international journals: *Frontiers of Computer Science* and *Ad Hoc & Sensor Wireless Networks*. He is also an organizing/program committee member for many international conferences.

Jiannong Cao (F'15) received the B.Sc. degree from Nanjing University, Nanjing, China, in 1982, and the M.Sc. and Ph.D. degrees from Washington State University, Pullman, WA, USA, in 1986 and 1990, respectively, all in computer science.

He is currently the Head and Chair Professor of the Department of Computing, Hong Kong Polytechnic University, Hung Hom, Hong Kong. His research interest includes parallel and distributed computing, networking, mobile and wireless computing, fault tolerance, and distributed software architecture. He has published more than 200 technical papers in the above areas. His recent research has focused on mobile and pervasive computing and mobile cloud computing.

Dr. Cao is a Senior Member of the China Computer Federation, and a member of the ACM. He has served as a member of editorial boards of several international journals, and also as an organizing/program committee member for many international conferences.

Keqin Li (F'15) received the B.S. degree in computer science from Tsinghua University, China, in 1985, and the Ph.D. degree in computer science from the University of Houston, Houston, TX, USA, in 1990. He is a SUNY Distinguished Professor of computer science. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU–graphics processing unit (GPU) hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file-sharing systems, mobile computing, service computing, Internet of things and cyberphysical systems. He has published more than 430 journal articles, book chapters, and refereed conference papers, and has received several best paper awards.

Dr. Li is currently serving or has served on the editorial boards of IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Cloud Computing, IEEE Transactions on Services Computing, and *Elsevier Journal on Parallel and Distributed Computing*.