An Intelligent Path Planning Scheme of Autonomous Vehicles Platoon Using Deep Reinforcement Learning on Network Edge

CHEN CHEN (Senior Member, IEEE), JIANGE JIANG (Senior Member, IEEE), NING LV (Senior Member, IEEE), AND SIYU LI
State Key Laboratory of Integrated Service Networks, Xidian University, Xi’an 710071, China
Corresponding author: Jiange Jiang (m13340110704@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFE0126000, in part by the National Natural Science Foundation of China under Grant 61672131, in part by the Key Research and Development Plan of Shaanxi Province under Grant 2017ZDCXL-GY-05-01, Grant 2019ZDLGY13-04, and Grant 2019ZDLGY13-07, and in part by the Xi’an Key Laboratory of Mobile Edge Computing and Security under Grant 201805052-ZD3CG36.

ABSTRACT Recent advancements in Intelligent Transportation Systems suggest that the roads will gradually be filled with autonomous vehicles that are able to drive themselves while communicating with each other and the infrastructure. As a representative driving pattern of autonomous vehicles, the platooning technology has great potential for reducing transport costs by lowering fuel consumption and increasing traffic efficiency. In this paper, to improve the driving efficiency of autonomous vehicular platoon in terms of fuel consumption, a path planning scheme is envisioned using deep reinforcement learning on the network edge node. At first, the system model of autonomous vehicles platooning is given on the common highway. Next, a joint optimization problem is developed considering the task deadline and fuel consumption of each vehicle in the platoon. After that, a path determination strategy employing deep reinforcement learning is designed for the platoon. To make the readers readily follow, a case study is also presented with instantiated parameters. Numerical results shows that our proposed model could significantly reduce the fuel consumption of vehicle platoons while ensuring their task deadlines.

INDEX TERMS Path planning, platooning, fuel consumption, Q-learning.

I. INTRODUCTION
In recent years, with the development of economy and the advancement of urbanization, the global car ownership has gradually increased, and a series of problems such as traffic congestion, accidents, pollution, shortage of land resources have become increasingly prominent. Transportation industry is the basic need and precondition of national economic development. Road traffic has become the subject with the largest energy consumption and the fastest growth rate among all kinds of transportation modes. Transportation has accounted for more than 30% of China’s oil consumption, among which road transportation is the most energy-consuming mode (accounting for more than 70%) [1].

Novel semi-automated driving technologies, collectively referred to as Cooperative Adaptive Cruise Control (CACC) [2], enable vehicles to drive very close together as a platoon. From Figure 1, we can see that vehicles in a platoon are virtually linked and communicate with each other through wireless communication technology [3], [4]. The leading vehicle is manually driven at the first position of the platoon and automatically followed by one or more
following vehicles. This means that the following vehicles automatically brake, steer and (de)accelerate based on the actions of the leading vehicle [5]. Driving close together reduces fuel consumption as it improves the aerodynamics of all vehicles in the platoon [6]. Test vehicle experiments suggest savings of up to six percent for the leading vehicle and ten percent for the following vehicles [7].

On the other hand, using path planning to choose an optimal path for the platoon can reduce traffic congestion that directly leads to increasing fuel consumption [8]. Path planning is to find a path from the starting point to the terminal point in a particular network according to some evaluation standards (such as the shortest or the fastest, etc) [9], which is an important way to reduce fuel consumption. Traditional path planning algorithm can only solve the static shortest path problem [10], due to not considering the real-time road conditions such as traffic jams, accidents, etc., the shortest path is not the optimal path. Therefore, many scholars have carried out extensive research on the path planning problem [11], and made some progress, but mainly focused on solving the shortest path or the shortest time, the research on energy saving path planning is very little.

So it is our goal to train vehicles according to machine learning [12], so as to improve vehicle learning and judgment function, reduce the time required for judgment calculation, quickly select the optimal path [13]. Millions of vehicles are connected to the Internet, generating zillions bytes of data at the network edge. Driving by this trend, there is imperative to push the AI frontiers to the network edge so as to fully unleash the potential of the edge big data [14]. To meet this demand, edge computing [15], [16], an emerging paradigm that pushes computing tasks and services from the network core to the network edge [17], has been broadly recognized as a promising solution. In this paper, we propose integrating the edge computing and the platoon computing together, Figure 2 is a VANET model based on edge computing, and then use Q-learning algorithm for path planning. The system scheme is shown in Figure 3.

The non-platoon represents a free vehicle that is not in the platoon, and wired link means network cable, optical fiber and other channels that can transmit data.

The main contributions of our work are as follows:

- We employ the platooning technology for autonomous vehicles, which allows vehicles to electronically dock on the road and then follow completely automatically. And platooning can increase the driving stability and minimizing the waste of gasoline.
- Reinforcement learning is applied in our work to enable the vehicle to choose the optimal path at the known starting and ending points, so as to improve the driving efficiency and road utilization.
- To address the problem of limited storage and computing power of platoon, we put forward the mobile edge computing-based platoon to accelerate the planning of the optimal path.

The rest of our paper is organized as following. In Section II, some related work was reviewed. In Section III, We form platoon model according to the traffic information, and calculate the utility of platoon according to the analysis, so as to establish a model of energy saving. Section IV, we calculate the best path with reinforcement learning. In Section V, we propose a case study, in which the platoon can choose the most fuel-efficient route. Numerical results are given in Section VI with corresponding performance evaluations. Our paper is concluded in Section VII, with the frontier direction, development trend and challenges given.

For simplicity, the notations used in our paper are listed in Table 1.

II. RELATED WORK

A. INTELLIGENT PATH PLANNING

In path planning, many ways are used to study the strategy of path planning. The A* algorithm [18] is one of the most efficient algorithms for calculating a safe route with the shortest distance cost. However, the route generated by the conventional A* algorithm is constrained by the resolution of the map. Multi-agent path planning (MAPP) [19] is increasingly being used to address resource allocation problems in highly dynamic, distributed environments that involve autonomous agents. Dijkstra algorithm [20] has the defect of high search time without considering the target information in the global space. In addition, many intelligent algorithms, such as fuzzy logic [21], genetic algorithm (GA) [22], and neural network [23], [24] have been used for path planning of mobile robots [25], [26]. Among the existing methods to solve the global path optimization problem, GA shows strong robust optimization performance by simulating the natural evolution of the population. Imitation learning [27] can provide direct reference for every step of decision-making, so as to alleviate
the problem of delayed reward. Therefore, we will start the research on path planning through reinforcement learning. Q-learning is a branch of reinforcement learning, and it is mainly used to solve sequential decision-making problems and enable autonomous vehicles to learn a large number of behavioral skills with minimal human intervention.

B. PLATOONING EDGE COMPUTING

In the platoon, using reinforcement learning for path planning will inevitably lead to a large number of data for every vehicle, while the storage and computing capacity of the vehicle itself is limited. Mobile Edge Computing (MEC) is a new paradigm with potential to improve vehicular services through computation offloading in close proximity to mobile vehicles [28]. However, how to execute computing-intensive applications on resource-constrained vehicles still faces huge challenges. Ning et al. [30] and Juntao et al. construct an intelligent offloading system for vehicular edge computing by leveraging deep reinforcement learning. Vehicular edge computing (VEC) [31] is a new computing paradigm with a great potential to enhance vehicular performance by offloading applications from the resource-constrained vehicles to lightweight and ubiquitous VEC servers. Nevertheless, offloading schemes [32], where all vehicles offload their tasks to the same VEC server, can limit the performance gain due to overload. Multi-Access Edge Computing (MEC) [33] is an emerging network paradigm that can be exploited also in vehicular scenarios to foster a more effective and flexible service delivery. Therefore, we combine edge computing and Q-learning into the platoon.

### III. SYSTEM MODEL

In this section we first present the platooning model upon which the proposed controller is designed. Then, we combine the heuristic algorithm to optimize the overall fuel consumption problem. The procedure of this section is shown in Figure 4.

![FIGURE 4. Optimization process.](image1)

#### A. PLATOON MODEL

Assuming that $n$ autonomous vehicles drive at a constant speed $v$ on a highway with length $Z$, the number of vehicles on the highway, namely the traffic density, can be controlled through the entrance passage. Assuming that vehicles are in a platoon and there are $m$ platoons, with a total of $N$ vehicles. Vehicles belonging to the same platoon are numbered consecutively,

$$i = 1, \text{ starting at the head of the platoon, as shown in Figure 5. We use } v_i \text{ to represent the } i \text{th vehicle in each platoon.}$$

![FIGURE 5. Platoon system.](image2)

The distance between the two vehicles is as follows when the vehicles are in platoon as shown in Figure 5.

$$g_{i,i-1}(t) = s_{i-1}(t) - s_i(t) - L_i. \quad (1)$$

$g$ is the distance between two vehicles, $s$ is the position of the vehicle, $L$ is the length of the vehicle.

$$V(g) = \begin{cases} 
0, & g \leq g_{st} \\
\frac{v_{max}}{2} \left[ 1 - \cos \left( \pi \left( \frac{g - g_{st}}{g_{go} - g_{st}} \right) \right) \right], & g_{st} \leq g \leq g_{go} \\
v_{max}, & g \geq g_{go}
\end{cases} \quad (2)$$

Velocity range $V(g)$ gives the function relationship between expected speed and spacing. $g_{st}$ is the safe distance between vehicles in congested environment, and $g_{go}$ is the...
distance between vehicles in sparse environment, as shown in Figure 6(a).

\[
W(v_{i-1}) = \begin{cases} 
v_{i-1}, & v_{i-1} \leq v_{\text{max}} \\
v_{\text{max}}, & v_{i-1} > v_{\text{max}} 
\end{cases}
\]

Saturation function \(W(v_{i-1})\) describes the switching between \(v_{i-1}\) and \(v_{\text{max}}\) in normal cruise control mode, as shown in Figure 6(b).

When road load and road delay remain unchanged, the platooning model combined with the following driving can shorten the distance between vehicles, increase the road capacity, and improve vehicle arrival rate and passing rate.

A particularly important aspect of the optimal path is to coordinate routes so that platoon can benefit from the overlap of their routes [34]. Therefore, the route of a single vehicle includes not only the desired path, but also the position of their routes [34]. Therefore, the route of a single vehicle coordinate routes so that platoon can benefit from the overlap.

But the cars don’t necessarily go exactly as expected. If this happens, there will be two situations, one is that the motorcade arrives before the expected time, the other is that it arrives beyond the expected time. The distance the vehicle travels is

\[
\text{Din} := \sum_{i=1}^{N_e-1} Z(e[i]) + x^D - x^S = \sum_{i=1}^{N_e} v[i][\hat{t}[i+1] - \hat{t}[i])
\]

\(N_e\) is the number of intersections from the starting position to the terminal position.

When vehicles are traveling in a platoon, we ignore the size of the vehicle, so each platoon consists of a leader and followers. We use the fuel consumption model for each distance as a function of speed, and determing whether the vehicle follows the platoon. The leader of a motorcade consumes the same amount of fuel as a single vehicle, while the follower reduces fuel consumption. Assuming that vehicles are running in a platoon, we indicate that the fuel consumption for each distance is \(f : [v_{\text{min}}, v_{\text{max}}] \times \{0, 1\} \to R^+\):

\[
f(v, \text{follower}) = \begin{cases} 
f_0, & \text{if follower } = 0 \\
f_{\text{follower}}, & \text{if follower } > 0 
\end{cases}
\]

\(f_0\) is the consumption of fuel when the car is the leader in the platoon or when it drives alone. \(f_{\text{follower}}\) is the number of following vehicles, so \(\text{follower } > 0\) is the fuel consumed by the vehicle when it is a follower. Applicable data is analyzed from following functions.

\[
F_{\text{air}} = \frac{1}{2} c_D S_S v^2
\]

The main reason for reducing fuel consumption through platoon is the result of reduced air resistance. Air resistance \(F_{\text{air}}\) can be modeled as equation (6), where \(c_D\) is resistance coefficient, \(\rho\) is air density, \(S_S\) is front area of vehicle,
\( v \) is relative speed of vehicle relative to air flow. When the velocity \( v \) decreases, the air resistance can be significantly reduced.

It is necessary to reduce the speed adjustment, because the speed adjustment will lead to the increase of fuel consumption. For example, when a vehicle is required to maintain a constant 80 km/h, the air resistance of the following vehicle decreases and the required distance between vehicles can be maintained. So, the constraint on the speed of the vehicle is very important. In one experiment \([35]\), the total amount of fuel saved by vehicles running in the platoon was 1045 liters of diesel, and the carbon dioxide emission reduction was 2770 kg. The same goes for all types of vehicles. Therefore, the way to reduce the air resistance is to reduce the air resistance coefficient \( c_D \), which depends on the specific shape of the vehicle and the position of other vehicles. We know that when a vehicle travels at a certain speed, the greater the air resistance, the greater the starting power required, and the greater the fuel consumption. Therefore, reducing the resistance of vehicles in the process of driving is an important measure to reduce energy consumption.

\[
F_{\text{fuel}} = F_{\text{ins}} = p_1 F + p_2
\]

\[
F_{\text{fuel}} = \frac{1}{N_c D_m} \sum_{i=1}^{N} \int_{s_i}^{s_{i+1}} \phi_i \frac{1}{v} ds
\]

\( \phi_i \) is the instantaneous fuel consumption, \( F_{\text{fuel}} \) is the total fuel consumption per vehicle, \( p_1 \) and \( p_2 \) are the influence parameters of engine and gearbox. Specifically, \( p_2 \) is the fuel consumption when the engine idles, and \( P \) is the instantaneous power.

We set a following or leader plan for each vehicle to minimize fuel consumption. The total fuel consumption \( F(\phi_n, \pi_n) \) associated with vehicle \( n \) plan is derived from (5) fuel consumption during the journey.

\[
F(\phi_n, \pi_n) = \int_{t_n}^{t_n^A} f(\phi_n(t), \pi_n(t)\phi_n(t)) dt
\]

\( \phi_n \) is the time distribution of vehicle speed, \( \pi_n \) is the trajectory, \( t_n^A \) is the platooning time of \( n \) vehicles, \( t_n \) is the starting time.

\[
F_c = \sum_{N \in N_c} F(\phi_n, \pi_n)
\]

\( F_c \) is the fuel consumption in the target state of the minimum platoon.

\[
\Delta F(n, N_c) = F(\phi_n, \pi_n) - F(\phi_n, \pi_n, 1)
\]

When fuel savings is \( \Delta F \),

According to the vehicle routing model \( E_c = (N_c, e_c, \Delta F) \) and the team vehicle \( N_l \subset N_c \), We conclude that the total amount of the largest fuel savings is formula (12), which is the optimal state.

\[
F_w = \sum_{i \in N_l \setminus N_l} \max \Delta F(n, N_c)
\]

We can use the cluster algorithm to iterate the result until we get the result \( F_w \) which will not continue to raise. Input a directed graph, output is a group of coordination leaders \( N_c \). Initially, \( N_c \) is an empty set, and each iteration selects a node \( i \). If the target function \( \Delta F \) is added to or removed from \( N_c \), the target function \( \Delta F \) is added to the node and \( N_c \) is updated accordingly.

The clustering algorithm guarantees convergence in a limited time. This is explained by the fact that the number of vehicles in the subset of \( N_c \) is limited, so the number of assigned values is also limited. In each iteration, \( \Delta F(n, N_l) \) increases strictly, so it changes in each iteration, and the same assignment as \( N_l \) never is repeated. Therefore, in the worst case, the clustering algorithm traverses all subsets before termination.

The calculation of vehicle driving plan is divided into four stages:

- Route calculation \( n \in N_c \): Route calculation method for calculating road network is used.
- Platoon calculation for two vehicles: When calculating plans involving two vehicles, the most fuel-efficient routes in these driving processes are recorded as road network models \( E_c = (N_c, e_c) \).
- Selection of motorcade plan: A leading vehicle \( N_l \subset N_c \) is chosen as the consistent subset of the plan calculated in the previous stage, i.e. coordinating the leader. A leading vehicle \( N_l \subset N_c \) is selected in the consistent subset of the driving plan calculated in the previous stage, that is, the coordination leader.
- Joint optimization: The selected motorcade plan is optimized to reduce fuel consumption. The vehicles of platoon are calculated in step 3.

IV. ROUTING STRATEGY

Vehicles are lined up according to their driving paths. Q-learning algorithm is used to train vehicles according to their learning, so as to improve the ability of vehicle robot to learn and judge, reduce the time needed to judge the most fuel saving path, and improve the reaction efficiency. In order to achieve the goal, we generally use state income function and state - action income function as evaluation criteria. In reinforcement learning, all system and environment states are considered Markovian. This assumption simplifies the decision-making process.

When a vehicle changes from the starting state to the ending state (\( S \)), certain actions must be taken during its movement (\( A \)). Every time we take an action, we will get a certain reward. \( R \) is the reward we get after executing the action, positive reward or negative reward is determined by our action. Strategy sets \( \pi \) depend on our action sets. According to the probability of greedy algorithm \( \epsilon \) [36], the return we get will determine our value. Our task is to select strategies to maximize our value.

As shown in Figure 8, we make assumptions about the selected area that the vehicle will encounter before driving. The area in the dotted box is the path area that the vehicle
FIGURE 8. Road condition selection area.

FIGURE 9. State transition income diagram.

C. Chen et al.: Intelligent Path Planning Scheme of Autonomous Vehicles Platoon

will choose. Figure 9 is the state transition income diagram. We make the assumption that the conditions of arrival are the same and the length of the section is the same.

Define a function $Q(s_t, a_t) = \max R_{t+1}$, in which action $A$ can be executed in state $s_t$ to maximize the benefits after the action is completed.

$$\pi(s) = \arg\max_a Q(s, a) \quad (13)$$

$$Q_{gain}(s, a) = r + \gamma \max_{a'} Q(s', a') \quad (14)$$

$\gamma$ is learning rate.

Algorithm is as follows:

- Set the current state = the initial state;
- Starting from the current state, find the action with the highest Q value;
- Set the current state = the next state;
- Repeat steps 2 and 3 until the current state = the target state.

The above algorithm will return to the specific process of the state sequence from the initial state to the target state, as shown in Figure 10.

Feedback $r$ is the basis of evaluating behavior. The weight of the path is modeled as feedback $r_t(s, a)$, with average time feedback $r_t$. The average time feedback is designed to truly reflect the influence of road traffic conditions on vehicle path planning. With the assistance of vehicle GPS, this optimization method can be used for real-time prediction. $r_d(s, a)$ distance feedback is the inherent feedback that reinforcement learning needs to design. $r_{goal}(s, a)$ is the end-point feedback to ensure that the vehicle will eventually drive to its destination. Feedback enables the vehicle to reach its destination in the shortest time (or the shortest distance). Feedback should be able to truly reflect the impact of traffic conditions on vehicles. Traffic conditions are usually described by factors such as the average time benefit $r_t$ and distance benefit $r_d$ of roads, so the benefit is assumed to be (15) (16) (17) (18).

$$r_t = \begin{cases} 10 & t \leq M \\ 6 & M < t \leq 2M \\ 2 & t > 2M \end{cases} \quad (15)$$

$$r_d = \begin{cases} 5 & d \leq xkm \\ 0 & \text{else} \end{cases} \quad (16)$$

$$r_{goal} = \begin{cases} 5 & \text{ArriveGoal} \\ 0 & \text{else} \end{cases} \quad (17)$$

$$r_m = \rho r_t + (1 - \rho) r_d + r_{goal} \quad (18)$$

$M$ is the average passage time of all sections of the traffic network studied; $d_{ij}$ is the length of each section; $\rho$ is the proportional coefficient, which can be determined according to the way the driver expects and set to be priority. The adjustment of $\rho$ will affect the path planning. We set the learning rate $\gamma$ equal to 0.9 and the initial matrix $Q$ as a
zero matrix. In the process of modeling, intersections can be regarded as nodes, and roads can be regarded as lines connecting two nodes. The weight on the line represents the road traffic condition. When the road network model and the path selection algorithm are available, the best path can be chosen.

V. A CASE STUDY

We assume that the road network is in the selected area of the dashed line frame shown in Figure 8, the selected area that the vehicle will encounter before driving, and we assume that the road environment is ideal and static, the vehicle starts from place 1, and 5 is the target node, there are many route choices in the process of driving obviously. We can determine the optimal path according to the above path selection algorithm, this paper uses greedy algorithm combined with Q-learning algorithm for path planning.

When applying the Q-learning algorithm to the path planning problem, we need to improve the \(Q_{goal}(s, a)\) step by step through trial and error, which requires the system to test and correct every possible \((s, a)\) through the feedback information for many times, so as to get a more suitable control strategy. If the system depends on the feedback from the environment step by step to improve its actions, it means that it will experience a very long learning and training process. To solve the problem of update slow, a counting threshold \(h\) is added to the algorithm in this paper. The update of Q value is determined by the cumulative access times of \((s, a)\). When the cumulative access times reach \(h\), the Q value of \((s, a)\) is updated. It has the ability of multi-step prediction. It can take into account the influence of multiple \((s, a)\) on the Q value in the future, making the learned decision more reasonable and greatly reducing the computational complexity. In order to compare the convergence rate of the algorithm, the following Figure 11 is given for illustration.

![Figure 11. Comparison of convergence rate of Q-learning algorithm.](image)

It can be found that with the increase of training time, the probability of the improved system to find the optimal path increases significantly faster and more quickly. Before the improvement, it will not converge until about 900 seconds, and after the improvement, it will converge at about 500 seconds.

In the process of learning, the probability of randomly selected strategies at the beginning is greater than that of greedy strategies. As the number of iterations increases, the probability of greedy strategy decreases, so the agent chooses action \(a_i\) according to the probability. The formula is shown in (19):

\[
\pi(x) = \begin{cases} 
\text{an action } a' \in A, \\
\epsilon \\
\arg\max_a Q(s, a'), 1 - \epsilon 
\end{cases} \quad (19)
\]

Parameter \(\epsilon \in [0, 1]\) indicates that the probability of search action is empirical value. According to the mechanism of greedy strategy search, simple Q-learning can effectively search for the best action value, but its drawback is that it only chooses the action corresponding to the maximum Q value in the set of alternative actions, and it is impossible to determine whether the choice is the best one or not, which may lead to Q-algorithm unable to search for the optimal path. According to \(\epsilon\)− greedy strategy, we randomly select one of all actions with \(\epsilon\) probability, and select the current optimal action with \(1 - \epsilon\) probability, so that each action may be selected, and multiple sampling will produce different selection paths.

\[
Q(s, a) = \begin{cases} 
(1 - \alpha)Q(s, a) + \alpha[R(s, a) + \gamma \times \max\{Q(\text{next state, all actions})\}], & s \neq \text{goal} \\
Q(s, a) + \gamma \times \max\{Q(\text{next state, all actions})\}, & s = \text{goal}
\end{cases}
\]

where \(\alpha\) is the learning rate, \(\gamma\) is the discount rate. When the vehicle arrives at state 5 (the target), we observe its behavior. We assume that the unreachable section is \(-1\), that is, the \(R\) value of the interconnected section is greater than or equal to 0. Assuming that time and path are equally important, we set the parameter \(\rho\) to 0.5. In order to simulate the real-time changes of road conditions, the distance between road nodes and congestion conditions are randomly set up. So we can deduce from formula (15) (16) (17) (18) and Figure 9:

Because node 5 is the destination, every time to 5 is a cycle until all updates are made. Every update is a learning process. According to the results of multiple iterations, the convergent matrix \(Q_{gain}\) is:

\[
Q_{gain} = \begin{bmatrix}
0 & 40.8 & 44.2 & 41.8 & 0 \\
38.8 & 0 & 44.2 & 0 & 0 \\
40.8 & 38.8 & 0 & 0 & 45.8 \\
28.4 & 0 & 0 & 0 & 45.3 \\
0 & 0 & 44.2 & 41.3 & 0
\end{bmatrix}
\]

Because the matrix \(Q_{gain}\) is initialized to 0, the Q matrix is obtained after several iterations. When the next state 5 (target state) becomes the current state, the iteration is stopped.
After continuous learning, matrix $Q$ will eventually converge. Once the matrix $Q_{gain}$ tends to converge, the optimal path to the target state is learned, and the highest reward value can be obtained according to the sequence of the optimal state. After continuous convergence optimization, it can be concluded that when the starting point is 1 and the target is node 5, the benefit of transition state is shown in Figure 12.

![Figure 12. State transition benefits.](image)

From the graph, we can see that the $Q_{gain}(state, action)$ corresponding value of path 1-3-5 is the largest. The traditional Q-learning algorithm will choose this path, but there is contingency. If the benefits of two paths are the same, then the system will choose the path randomly. When $\epsilon$-greedy strategy is added, with $\epsilon$ probability randomly chooses one action from all actions, and with $1 - \epsilon$ probability chooses the current optimal action, different path selection tables will appear, and the optimal result will be set to the top according to the result.

Intelligent vehicles use the above algorithms to learn from experience. Each experience is equivalent to one training session. In each training session, the agent explores the environment and gets a reward once it reaches the target state. The purpose of training is to enhance the brain of an agent, expressed in matrix $Q_{gain}$. The more training, the better matrix $Q_{gain}$ will be obtained. In this case, if the matrix $Q_{gain}$ has been optimized, then the vehicle will not blindly explore everywhere, but will find the fastest route to reach the target state. When the vehicle encounters similar situations again, the trend that this strategy will be selected, i.e. the probability will increase, whereas the trend of choosing this strategy will weaken, similar to the principle of conditioned reflection. Therefore, we can get the desired strategy by combining the vehicle platoon algorithm. We choose the most fuel-efficient route, and we can get the desired results by platooning in an energy-efficient way.

VI. NUMERICAL RESULTS

In this section, python simulation is used to verify the feasibility and effectiveness of the method in vehicle routing planning and the feasibility of saving fuel. Firstly, we use vissim software to simulate the scene of a single vehicle and test each road. Then, in a simple static environment, the optimal matrix $Q_{gain}$ is obtained by using $\epsilon$-Q-learning algorithm. Finally, the optimal matrix $Q_{gain}$ is applied to path planning in different dynamic environments. Compared with Dijkstra algorithm, Q-learning can compute multiple candidate paths to avoid the failure to change strategies rapidly in case of emergency. Compared with the k-shortest algorithm, Q-learning algorithm has higher efficiency and lower cost. According to the stability comparison of the path planning algorithm in Table 2, the stability of reinforcement learning is obviously higher than that of the other two algorithms. Compared with ant colony algorithm, Q-learning is more stable and reduces the random test results. It does not need too many tests, saves costs and reduces the random selection.

| Algorithm          | Stability |
|--------------------|-----------|
| Q-learning algorithm | 0.9       |
| Dijkstra algorithm  | 0.57      |
| k-shortest algorithm | 0.77     |

Table 3 shows the path selection based on $\epsilon$-Q-learning algorithm when the road condition is like Figure 8. When $\epsilon$ increases, the probability of choosing the main path 1-3-5 decreases and that of other paths increase. As can be seen from Figure 13, minor changes in $\epsilon$ will not have a significant impact, but will highlight the results. According to the experimental results, an appropriate $\epsilon$ value should be
selected without affecting the results. $\epsilon$ should not be too large, or the advantage will be reduced, and $\epsilon$ should not be too small, otherwise it may reduce the appearance of features, so this paper chooses $\epsilon = 0.18$.

Column Figure 14 shows the times our strategy chooses the best path according to $\epsilon$ value when the number of choices is 1, 10, 50, and 100, and $\epsilon$ is 0.01, 0.1, 0.18, 0.20, respectively. It can be seen that the main choice path is provided by the maximum probability selection. Figure 15 shows the time taken by traditional Q-learning and $\epsilon$-Q-learning when the number of intersection nodes is 5, 12, and 16. The number of iterations varies according to the number of nodes. In Figure 16, the simulation results show that the more learning times, the more nodes, the longer the simulation time. In the simulation process, we found that the learning results have been stable after several times, so there is no need to learn too many times. This can improve the efficiency of calculation and reduce unnecessary calculations.

Figure 16 shows a linear relationship between learning times and simulation time. Node5, node12, and node16 in the figure respectively represent the situation that the number of nodes is 5, 12, and 16. The number of iterations varies according to the number of nodes. In Figure 16, the simulation results show that the more learning times, the more nodes, the longer the simulation time. In the simulation process, we found that the learning results have been stable after several times, so there is no need to learn too many times. This can improve the efficiency of calculation and reduce unnecessary calculations.

Figure 17 shows that the time required for K-shortest planning increases linearly with the scale of the network, while the time required for multi-path planning of $\epsilon$-Q-Learning increases with the logarithm of the scale of the network, and the larger the scale of the network, the more obvious the advantages. In addition, the reliability of Q-learning multi-path planning can be guaranteed to be 0.8, but the reliability of K-shortest planning cannot be guaranteed.

When a motorcade travels at a speed of 80 kilometers per hour, the air drag reduction varies according to the distance between vehicles. From Figure 18, we can see that $V_1$, $V_2$, and $V_3$ are the leader, the first follower, and the second follower respectively, and the car body is the same.
Distance indicates the distance between cars. Figure 18 shows that the reduced air resistance of the following vehicle varies according to the relative position of the vehicle relative to the leader and the distance between the vehicles, which will affect the fuel consumption of each vehicle in the platoon, and then affect the overall fuel consumption of the platoon.

Figure 19 shows the impact of the length of platoon on fuel savings. Under the condition that the vehicle speed is 80km/h, the random state in the graph represents the fuel consumption savings of random driving, the actual represents the fuel savings in the ideal driving process of platooning, and the platooning perfect condition represents the excellent savings in the actual driving process of platoon vehicles. As shown in the figure, when the length of platoon is small, fuel savings increase rapidly with the number of tasks. As the length of platoon increases, the trend of fuel economy will decrease or even stagnate. Ideally, assuming that the vehicles are the same, as the length of platoon tends to be infinite, it gradually approaches the maximum fuel savings, because almost every car is a follower throughout the journey. Through the joint optimization of vehicle schemes, we can see obvious fuel savings.

VII. CONCLUSION

The purpose of this paper is to reduce fuel consumption. And for that, we use the platoon mode to reduce the air resistance between vehicles. When the vehicle is traveling, the system will form a platoon with the same terminal point, then take the beginning of overlapping sections as the starting point. Using known starting point and target point combined with reinforcement learning to train platoon to form the planned route quickly. And finally we achieve the purpose of reducing the total fuel consumption with platooning technology. On the other hand, from the point of view of path planning, we should not only control platoon energy saving and emission reduction, but also take control from the general direction, so this paper proposes to use greedy algorithm combined with Q-learning algorithm for path planning, using time, distance and other factors to design combination weight to ensure the reliability of the results. In order to make the route of platoon more likely to be applied to the autonomous vehicles in the real world, the future work needs to study the path planning problem in the unknown dynamic environment.

REFERENCES

[1] W. Liu and B. Lin, “Analysis of energy efficiency and its influencing factors in China’s transport sector,” J. Cleaner Prod., vol. 170, pp. 674–682, Jan. 2018.
[2] S. Bae, Y. Kim, J. Guanetti, F. Borrelli, and S. Moura, “Design and implementation of ecological adaptive cruise control for autonomous driving with communication to traffic lights,” in Proc. Amer. Control Conf. (ACC), Jul. 2019, pp. 4628–4634.
[3] G. Guo and W. Yue, “Autonomous platoon control allowing range-limited sensors,” IEEE Trans. Veh. Technol., vol. 61, no. 7, pp. 2901–2912, Sep. 2012.
[4] B. Wang, Y. Sun, Z. Sheng, H. M. Nguyen, and T. Q. Duong, “Inconspicuous manipulation for social-aware relay selection in flying Internet of Things,” IEEE Wireless Commun. Lett., vol. 8, no. 5, pp. 1394–1397, Oct. 2019.
[5] B. Wang, Y. Sun, S. Li, and Q. Cao, “Hierarchical matching with peer effect for low-latency and high-reliable caching in social IoT,” IEEE Internet Things J., vol. 6, no. 1, pp. 1193–1209, Feb. 2019.
[6] L. Salati, P. Schito, and F. Cheli, “Wind tunnel experiment on a heavy truck equipped with front-rear trailer device,” J. Wind Eng. Ind. Aerodynamics, vol. 171, pp. 101–109, Dec. 2017.
[7] V. Turri, B. Bessellink, and K. H. Johansson, “Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning,” IEEE Trans. Control Syst. Technol., vol. 25, no. 1, pp. 12–28, Jan. 2017.
[8] A. García Castro and M. D. C. Andrés, “Measuring the effects of traffic congestion on fuel consumption,” in Proc. Transp. Res. Board Conf., TRB, Washington, DC, USA, 2014, pp. 19–22.
[9] R. Wang, Z. Xu, X. Zhao, and J. Hu, “V2V-based method for the detection of road traffic congestion,” IET Intell. Transp. Syst., vol. 13, no. 5, pp. 880–885, May 2019.
[10] S. Sundar, A. Basu, A. Kar, R. Kumar, and H. M. Kittur, “Shortest path establishment approaches for static and dynamic mobile nodes with quality of service,” Electron. Government, Int. J., vol. 14, no. 1, pp. 1–15, 2018.
[11] K. Chu, J. Kim, K. Jo, and M. Sunwoo, “Real-time path planning of autonomous vehicles for unstructured road navigation,” Int. J. Automot. Technol., vol. 16, no. 4, pp. 653–668, Aug. 2015.
[12] M. Nazari, A. Oroojlooy, L. Snyder, and M. Takács, “Reinforcement learning for solving the vehicle routing problem,” in Proc. Adv. Neural Inf. Process. Syst., Sep. 2018, pp. 9830–9849.
[13] E. Larsson, G. Sennott, and J. Larson, “The vehicle platooning problem: Computational complexity and heuristics,” Transp. Res. C, Emerg. Technol., vol. 60, pp. 258–277, Nov. 2015.
[14] J. Du, L. Zhao, J. Feng, and X. Chu, “Computation offloading and resource allocation in mixed Fog/Cloud computing systems with min-max fairness guarantee,” IEEE Trans. Commun., vol. 66, no. 4, pp. 1594–1608, Apr. 2018.
Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise
S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas, “Fuel-optimal
C. Chen, L. Liu, T. Qiu, K. Yang, F. Gong, and H. Song, “ASGR: An artificial Spider-Web-Based geographic routing in heterogeneous vehicular networks,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 5, pp. 1604–1620, May 2019.
R. Song, Y. Liu, and R. Bucknall, “Smoothed A* algorithm for practical unmanned surface vehicle path planning,” Appl. Ocean Res., vol. 83, pp. 9–20, Feb. 2019.
R. Shi, P. Sreenkiste, and M. Veloso, “SC-M*: A multi-agent path planning algorithm with soft-collision constraint on allocation of common resources,” Appl. Sci., vol. 9, no. 19, p. 4037, Sep. 2019.
M. Enayattabar, A. Ebrahimeinjad, and H. Motameni, “Dijkstra algorithm for shortest path problem under interval-valued pythagorean fuzzy environment,” Complex Intell. Syst., vol. 5, no. 2, pp. 93–100, Jun. 2019.
L. Cherroun, M. Boumechra, and A. Koutouz, “Mobile robot path planning based on optimized fuzzy logic controllers,” in New Developments and Advances in Robot Control. New York, NY, USA: Springer, 2019, pp. 255–283.
M. Nazarahari, E. Khammirza, and S. Doostie, “Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm,” Expert Syst. Appl., vol. 115, pp. 106–120, Jan. 2019.
M. Luo, X. Hou, and S. X. Yang, “A multi-scale map method based on bioinspired neural network algorithm for robot path planning,” IEEE Access, vol. 7, pp. 142682–142691, 2019.
Z. Duan, Y. Yang, K. Zhang, Y. Ni, and S. Baigain, “Improved deep hybrid networks for urban traffic flow prediction using trajectory data,” IEEE Access, vol. 6, pp. 31820–31827, 2018.
J. Guan, R. Lai, A. Xiong, Z. Liu, and L. Gu, “Fixed pattern noise reduction for infrared images based on cascade residual attention CNN,” Neurocomputing, vol. 377, pp. 301–313, Feb. 2020.
J. Zhang, K. Zhang, Z. Qin, H. Yin, and Q. Wu, “Sensitive system calls based packed malware variants detection using principal component initialized MultiLayers neural networks,” Cybersecurity, vol. 1, no. 1, pp. 1–13, Dec. 2018.
J. M. Pfeiffer, S. Shukla, M. Turchetta, C. Cadena, A. Krause, R. Siegwart, and J. Nieto, “Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations,” IEEE Robot. Autom. Lett., vol. 3, no. 4, pp. 4423–4430, Oct. 2018.
K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, “Optimal delay constrained offloading for vehicular edge computing networks,” in Proc. IEEE Int. Conf. Commun. (ICC), May 2017, pp. 1–6.
Z. Ning, P. Dong, J. J. P. C. Rodrigues, F. Xia, and X. Wang, “Deep reinforcement learning for vehicular edge computing: An intelligent offloading system,” ACM Trans. Intell. Syst. Technol., vol. 10, no. 6, pp. 1–24, 2019.
G. Juntao, R. Lai, and A. Xiong, “Learning spatiotemporal features for single image stripe noise removal,” IEEE Access, vol. 5, no. 1, pp. 1–8, 2019.
Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, “Joint load balancing and offloading in vehicular edge computing networks,” IEEE Internet Things J., vol. 6, no. 3, pp. 4377–4387, Jun. 2019.
C. Chen, J. Hu, T. Qiu, M. Atiquzzaman, and Z. Ren, “CVCG: Cooperative V2 V-aided transmission scheme based on coalitional game for popular content distribution in vehicular ad-hoc networks,” IEEE Trans. Mobile Comput., vol. 18, no. 12, pp. 2811–2828, Dec. 2019.
Z. Laaroussi, R. Morabitio, and T. Taleb, “Service provisioning in vehicular networks through edge and cloud: An empirical analysis,” in Proc. IEEE Conf. Standards Commun. Netw. (CSCN), Oct. 2018, pp. 1–6.
C. Chen, L. Liu, T. Qiu, D. O. Wu, and Z. Ren, “Delay-aware grid-based geographic routing in urban VANETs: A backbone approach,” IEEE/ACM Trans. Netw., vol. 27, no. 6, pp. 2324–2337, Dec. 2019.
S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas, “Fuel-optimal centralized coordination of truck platooning based on shortest paths,” in Proc. Amer. Control Conf. (ACC), Jul. 2015, pp. 3740–3745.
Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 19, no. 1, Jan. 2007, pp. 153–160.

CHEN CHEN (Senior Member, IEEE) received the B.Eng., M.Sc., and Ph.D. degrees in electrical engineering and computer science (EECS) from Xidian University, Xi’an, China, in 2000, 2006, and 2008, respectively. He was a Visiting Professor with the Department of EECS, The University of Tennessee, and the Department of CS, University of California. He is currently an Associate Professor with the Department of Telecommunication, Xidian University. He is also a member of the State Key Laboratory of Integrated Service Networks, Xidian University. He is also the Director of the Xi’an Key Laboratory of Mobile Edge Computing and Security, and the Intelligent Transportation Research Laboratory, Xidian University. He has authored or coauthored two books, over 90 scientific articles in international journals and conference proceedings. He has contributed to the development of five copyrighted software systems and invented over 80 patents. He is also a Senior Member of the China Computer Federation (CCF). He is also a member of ACM and the Chinese Institute of Electronics. He serves as the General Chair, the PC Chair, the Workshop Chair, or a TPC Member for a number of conferences.

JIANGE JIANG received the B.Eng. degree in electronic and information engineering from Nanchang Hangkong University, in 2019. She is currently pursuing the master’s degree with Xidian University. Her research interests include machine learning, data fusion, intelligent transportation, mobile edge computing, and the Internet of Vehicle.

NING LV received the B.Eng. degree in electrical engineering and automation from Xi’an Jiaotong University, in 2000, and the M.Sc. degree in circuit and system engineering from Xidian University, Xi’an, China, in 2004. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent system with Xidian University.

SIYU LI received the B.Eng. degree in communication engineering from Heilongjiang University, in 2016, and the M.Sc. degree in electronic and communication engineering from Xidian University, Xi’an, China, in 2019. Her research interests include vehicular ad hoc networks, intelligent transportation, mobile edge computing, and the Internet of Things. 

VOLUME 8, 2020