An Intelligent Path Planning Scheme Autonomous Vehicles Platoon Using Reinforcement Learning*

*Note: This is a reproduction of the original paper

Kasra Khalafi
Electrical and Computer Engineering Department
Queen's University
Kingston, Ontario, Canada
kasra.khalafi@queensu.ca

Golnaz Bashirian
Electrical and Computer Engineering Department
Queen's University
Kingston, Ontario, Canada
golnaz.bashirian@queensu.ca

Abstract*—With the advancement in the Intelligent Transportation Systems and Artificial Intelligent, it is expected that in the near future roads are going to be filled with intelligent self-driving and autonomous vehicles, which have the ability to communicate with other intelligent autonomous vehicles as well as the environment. By using the platooning technology, where intelligent vehicles form a specific pattern to drive, it is possible to amend traffic efficiency and lower fuel consumption and as a result, reduce transport costs. In this paper, we tried to reproduce the original paper, where a path planning scheme was used to improve the driving efficiency of autonomous vehicular platoon by using Reinforcement Learning. It is worth mentioning that the original paper has used Reinforcement Learning instead of Deep Reinforcement Learning for path planning. First, the system model of autonomous vehicle platooning is given on the common highway. Following that, a joint optimization problem considering the task deadline and fuel consumption of each vehicle in the platoon is going to be presented. After that, two different path determination strategies employing reinforcement learning (Q-Learning and SARSA) are designed for the platoon. A case study is presented and the corresponding numerical results show that the proposed model could significantly reduce the fuel consumption of vehicle platoons while ensuring their task deadlines.

***Index Terms*—Path Planning, Platooning, Q-Learning, SARSA**

I. Introduction

With the development and expansion of urban areas in recent years, global ownership of cars has increased. As a consequence, issues like traffic congestion, accidents, and pollution, to name but a few, are growing and need to be addressed. For example, 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].

Vehicles in a platoon communicate with each other through wireless communication technologies [2]. The leader in the platoon starts to move and the following vehicles follow the leader automatically, i.e., followers speed up, speed down and try to follow the leader automatically. When vehicles drive close to each other, fuel consumption would reduce as it is going to improve the aerodynamics of all vehicles in the platoon [3]. As a test was implemented, the platoon can reduce up to 6% of fuel consumption for the leader and up to 10% for the followers [4]. Furthermore, finding and using an optimal path can lower traffic congestion, which directly affects fuel consumption [5]. Using path planning, we try to find this optimal path from starting point to the terminal point according to some evaluation standards such as the shortest or fastest path [6]. The shortest path is not always the optimal path because it does not consider real-time road conditions. For instance, the shortest path might have a long time to be passed, which is not energy optimal in this case. In the original paper, it is proposed to integrate edge computing and platoon computing together [8]. The system scheme is illustrated in the figure 1.

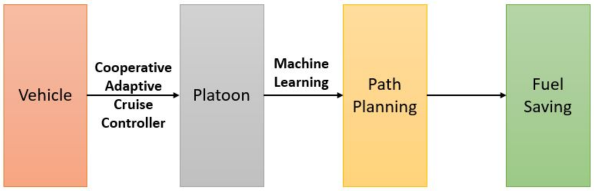


Fig. 1: System Scheme

The main contributions the original paper are:

- employing the platooning technology for autonomous vehicles to minimizing the waste of gasoline.
- To select the optimal path, Reinforcement Learning (Q-Learning and SARSA) to improve the driving efficiency and road utilization.

II. System Model

Using a platoon, the traffic density of the road can be controlled. Vehicles try to keep their distance from each other while following the leader and keeping their constant speed to reduce traffic density. Considering the total number of vehicles is N and there are m platoons. Vehicles which belong to the same platoon, are numbered consecutively from 1 to the number of vehicles in the corresponding platoon. Figure 2 illustrates i$^{\text{th}}$ vehicle in each platoon.

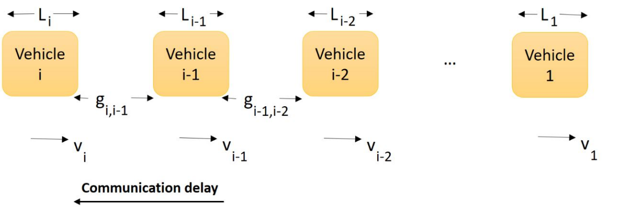


Fig. 2: Platoon System.

Distance between two vehicles when they are in a platoon like figure 2 is formulated in the equation 1:

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

in the equation 1, g refers to the distance, s refers to the position of the vehicle and L refers to the length of the vehicle.

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

equation 2 provides the velocity range for expected speed based on the space between vehicles. Based on the space between vehicles, speed of the platoon is going to be increased or decreased. g_{st} is the safe distance between the vehicles and g_{go} is the distance between vehicles when the space between them is sparse. In the paper, fuel consumption model and air resistance model for each distance is considered as function of the speed. Also, leader and followers all consume same amount of fuel. To calculate the air resistance for the vehicles, equation 3 is used:

$$F_{air} = \frac{1}{2} c_D \rho S_s v^2 \tag{3}$$

The air resistance reduction through platoon is the main reason for reducing the fuel consumption. In the equation 3, F_{air} is the air resistance. c_D is the resistance coefficient, ρ is the air density, S_s is the front area of the vehicle and v is the relative speed of the vehicle relative to the airflow. When the speed of the vehicle increase, air resistance can be notably increased. These variables were not provided in the original paper, but for c_D, ρ and S_s values 0.09, 1.225 and 2 respectively for the normal case. These values might be different from the original paper's implementation, but as it was not provided, these values were used in our implementation.

III. Routing Strategy

The vehicles are at the beginning state *(S)* and the the target is to reach the ending state *(E)* using the optimal fuel saving path according to the distances and the traffic condition. It is also important to improve the judgement time of vehicles and reduce the time for decision making.To achieve our goal, we have used Q-Learning and SARSA algorithms.

From the starting state to the ending state, the vehicle needs to take certain actions.By choosing each action, t gets a certain reward, which can be a positive reward or a negative one based on the conditions. Actions are selected according to policy π, which is ε-greedy strategy in this case.In each state, the set of actions is defined as the next possible places that the vehicle can go in the next state, so the set of action is different for each state based on the road network.The assumptions about the places that vehicle can go in each state are shown in Figure 3.

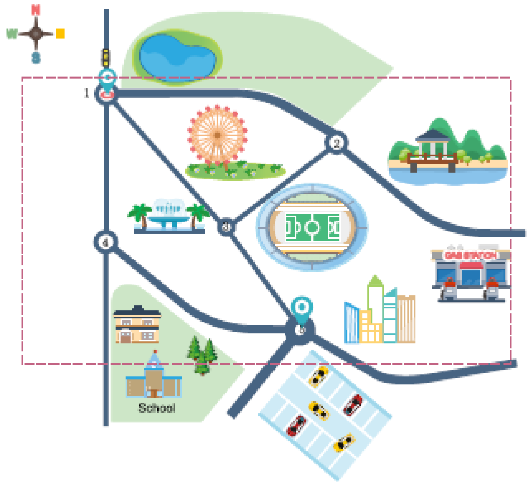


Fig. 3: Road Condition and Selection

In Figure 3, the area in dotted box is the path area vehicle can choose. A table ***Q(s$_t$, a$_t$)*** that contains the action values for each state in each time step. During the training process, the values of *Q*-table are updated many times to achieve the optimal values. The updates are made according the action taken in each state and the reward in the time-step after executing the action.In each time step, action A is taken based on policy π which can be a greedy, ε-greedy, etc. strategy. Algorithms are as follow [7]:

A. Q-Learning

- Initialize *Q(s, a)* for all states and actions
- Set current state to initial state;
- Choose action A using policy π;
- Take action A, observe R, S';
- Update *Q(S, A)*:

$$Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max{}_{\text{a}} Q(S', a) - Q(S, A)] \tag{4}$$

- $S \leftarrow S'$;
- Repeat steps 3, 4, and 5 until S is the terminal.

B. SARSA

- Initialize *Q(s, a)* for all states and actions
- Set current state to initial state;
- Choose action A using policy π;
- Take action A, observe R, S';

- Choose action A'from S'using policy π;
- Update *Q(S, A)*

$$Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \max{}_{a} Q(S', A') - Q(S,A)] \tag{5}$$

- $S \leftarrow S'$; $A \leftarrow A'$;
- Repeat steps 3, 4, and 5 until *S* is the terminal

γ is the discount rate and α is the learning rate. The term *R* in equations 4 and 5 to update the *Q*-table is the reward that evaluates the decision. Q-learning is an off-policy TD (Temporal Difference) control algorithm. In Q-Learning, the *Q* function directly approximates the optimal action values q^* independent of the policy. SARSA is an on-policy TD control algorithm. The algorithms uses the sequence (S_t,A_t,R_{t+1},S_{t+1},A_{t+1}) to make transition from one state action pair to another, which also gives the name *SARSA* for the algorithm [7].

The weight of the path is modeled as feedback r_t, the average time benefit, which reflects the influence of the road traffic condition on the path.r_d is the distance feedback that represents how close are places that can be reachable from each other.r_{goal} is the end-point feedback that is defined to ensure the vehicle drives to the destination [8]. The average time benefit r_t, road benefit r_d, and r_{goal} are calculated from (6), (7), and (8):

$$r_{\mathrm{t}} = \begin{cases} 10 & t \leq M \\ 6 & M < t \leq 2M \\ 2 & t \geq M \end{cases} \tag{6}$$

$$r_{\mathrm{d}} = \begin{cases} 5 & d \leq xkm \\ 0 & else \end{cases} \tag{7}$$

$$r_{\mathrm{goal}} = \begin{cases} 5 & Arrived\ Goal \\ 0 & else \end{cases} \tag{8}$$

M is the average passing time of all sections of the traffic network studied and *d* is the length of each path. The total reward *R* is calculated as:

$$R = \rho r_{\mathrm{t}} + (1-\rho) r_{\mathrm{t}} + r_{\mathrm{goal}} \tag{9}$$

ρ is the proportional coefficient which can be determined according to driver expects. We set discount rate γ to 0.9 and the learning rate α to 0.7. The *Q*-table is initialized to zero.

IV. A Case Study

The road network is represented in Figure 3. There are 5 nodes each of them is a place; The vehicle always starts from place 1 and the target is placed 5. During the path, there are many route choices and the optimal path is determined according to a greedy algorithm combined with Q-Learning in one case and SARSA in the second case. It is assumed that the road environment is ideal and static.

For Q-Learning and SARSA path planning methods, we need to update the *Q*-table for each step which requires test and correction of every possible pair of state and action *(s, a)* through feedback information for several episodes. To avoid a long learning and training process, a counting threshold *h* is added. By increasing the training time, the system is more improved and the probability of choosing the optimal path increases. If the number of nodes extends, more time is needed to find the optimal *Q*-table since there are more choices in each step.

In the process of learning, ε-greedy algorithm is used for action selection; this means that the probability of selecting paths randomly is less than that of a greedy strategy. The reason is that it will converge to the optimal *Q* values faster. The formula of ε-greedy action selection strategy is shown in (10):

$$\pi(x) = \begin{cases} an\ action\ a' \in A\ , & \varepsilon \\ arg\ Max_a\ Q(s',\ a)\ , & 1\ \text{-}\ \varepsilon \end{cases} \tag{10}$$

Parameter $\varepsilon \in$ [0, 1] shows the probability of exploring the actions.The greedy algorithm chooses the action with the greatest ^{Q}-value that is the best action for the next time step but it is not necessarily the best action in the long term since the future states and rewards are not considered and it does not guarantee that the algorithm is searching for the best path. In ε-greedy algorithm, action is selected randomly with probability ε, and the optimal path is selected with probability 1 - ε. With this algorithm, it is more probable that each action may be selected and different path planning strategies are considered. For simulation of traffic condition of the road and the distances, formula (6), (7), and (8) are used. In Matrix r_d. it is assumed that unreachable states are -1 so that the *R* value o of that section is greater than or equal to 0:

$$r_t = \begin{bmatrix} -1 & 6 & 10 & 6 & -1 \\ 2 & -1 & 10 & -1 & -1 \\ 6 & 1 & -1 & -1 & 6 \\ 0 & -1 & -1 & -1 & 10 \\ -1 & -1 & 5 & 0 & -1 \end{bmatrix}$$

$$r_d = \begin{bmatrix} -1 & 5 & 5 & 5 & -1 \\ 5 & -1 & 5 & -1 & -1 \\ 5 & 1 & -1 & -1 & 5 \\ 5 & -1 & -1 & -1 & 5 \\ -1 & -1 & 5 & 0 & -1 \end{bmatrix}$$

After running two algorithms for multiple times, the Q_{gain} matrices are:

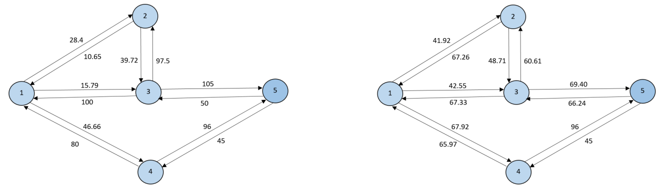
- Q-Learning

$$Q_{gain} = \begin{bmatrix} 0 & 28.4 & 15.79 & 46.66 & 0 \\ 10.65 & 0 & 39.72 & 0 & 0 \\ 100 & 97.5 & 0 & 0 & 105 \\ 80 & 0 & 0 & 0 & 96 \\ 0 & 0 & 50 & 45 & 0 \end{bmatrix}$$

- SARSA

$$Q_{gain} = \begin{bmatrix} 0 & 41.92 & 42.55 & 67.92 & 0 \\ 67.26 & 0 & 48.71 & 0 & 0 \\ 67.33 & 60.61 & 0 & 0 & 69.95 \\ 80 & 0 & 0 & 0 & 96 \\ 0 & 0 & 66.94 & 62.32 & 0 \end{bmatrix}$$

After continuous learning, matrix Q_{gain} converges to the optimal values. The benefit of transition state is shown in Figure 4.



(a) Q-Learning

(b) SARSA

Fig. 4: State Transition Benefits

V. Numerical Results

In this section, we are going to investigate the results and effectiveness of vehicle routing and air resistance. Vissim software to simulate the scene of a single-vehicle. For finding the optimal path, Q_{gain} is obtained using ϵ-Q-Learning and SARSA methods. Q-Learning and SARSA can compute multiple possible paths in case of emergency to prevent failure of the strategy change, which is an advantage over the Dijkstra algorithm [8]. Also to compare Q-Learning and SARSA with the k-shortest algorithm, they have lower costs and higher efficiencies [8]. Table 1 and Figure 5 are reproductions of Table 3 and figure 13 of the original paper respectively.

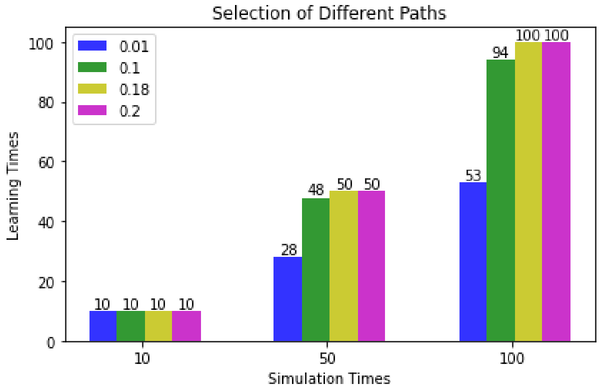


Fig. 5: Q-Learning:Relationship between learning times and simulation time

Table I illustrates the selection of different paths based on ϵ-Q-Learning when having 5 nodes (destinations). As shown in the Table I, as ϵ increases, vehicles try to choose the optimal path more often instead of choosing other paths for moving from node 1 to node 2. As we increase the ϵ, agents try to explore more instead of just exploiting. As an ϵ should be selected, it should not be too large to explore a lot and should not be too small to reduce the appearance of features. In the original paper, $\epsilon = 0.18$ is chosen, which is also a proper value in our implementation.

TABLE I: Selection of different paths using Q-Learning

path planning	ϵ	Times=10	Times=50	Times=100
1-3-4	0.01	10	28	53
1-4-5	0.01	0	10	27
others	0.01	0	12	20
1-3-4	0.1	10	48	94
others	0.1	0	2	0
1-3-4	0.18	10	50	100
others	0.18	0	0	0
1-3-4	0.2	10	50	100
others	0.2	0	0	0

TABLE II: Selection of different paths using SARSA

path planning	ϵ	Times=10	Times=50	Times=100
1-3-4	0.01	6	39	67
1-4-5	0.01	2	4	14
others	0.01	2	7	19
1-3-4	0.1	6	24	65
others	0.1	4	26	35
1-3-4	0.18	8	28	74
others	0.18	2	22	26
1-3-4	0.2	7	30	61
others	0.2	3	20	39

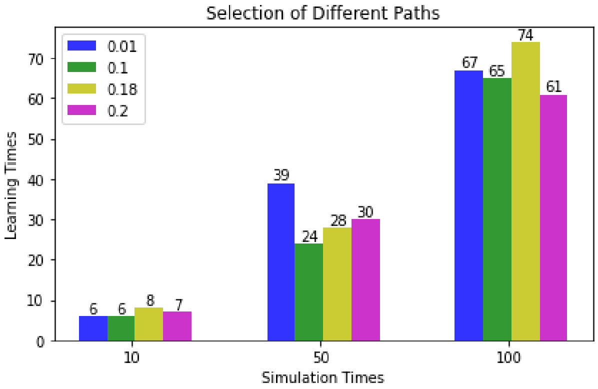
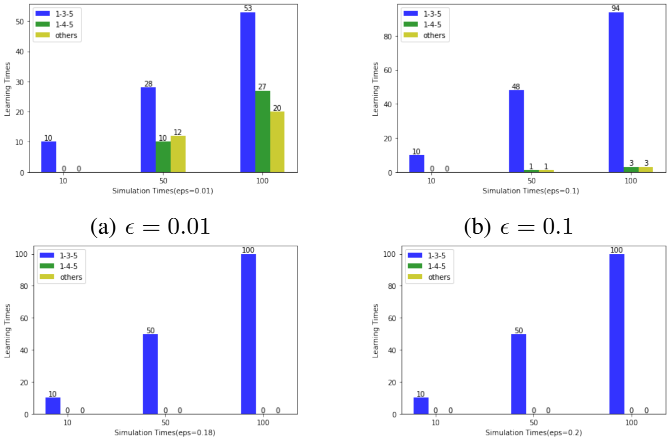


Fig. 6: SARSA:Relationship between learning times and simulation time

Table II and figure 6 are the same reproduction of table 3 and figure 13 of the original paper respectively, but this time SARSA method was used for path planning. As you can see, Q-Learning has performed better than SARSA in almost all cases. Using SARSA, $\epsilon = 0.18$ has also almost the best effectiveness in finding the optimal path.
figure 7 illustrates in detail the times the Q-Learning method chooses the optimal and other paths using different ϵ values 0.01, 0.1, 0.18 and 0.2 respectively. This figure is a reproduction of figure 14 of the original paper. To select the optimal, we use the highest value in Q-table for each state their corresponding possible actions,

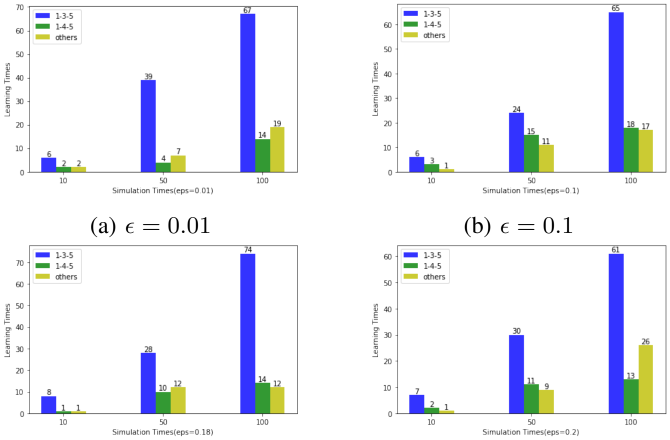
Figure 8 is similar to the previous figure, but this time SARSA method is going to be used to select the optimal path



(a) $\epsilon = 0.01$ (b) $\epsilon = 0.1$

(c) $\epsilon = 0.18$ (d) $\epsilon = 0.2$

Fig. 7: the influence of on path selection using Q-Learning method.



(a) $\epsilon = 0.01$ (b) $\epsilon = 0.1$

(c) $\epsilon = 0.18$ (d) $\epsilon = 0.2$

Fig. 8: the influence of on path selection using Q-Learning method.

using different values for the ϵ. This figure is also reproduction of the figure 14 from original paper.

Figure 7 depicts the time taken using ϵ-Q-Learning and SARSA for a single learning time when there are 5, 12 and 16 nodes respectively. This figure is reproduction of the figure 15 from original paper. The system used to implement the codes in our reproduction is Google Colab.

As can be seen in the figure 7, when learning times increase, the timing for the SARSA method witness a steady increase, while Q-Learning experiences a smoother increase. Also, timing in the SARSA method compared to the Q-Learning method is larger. Results provided by both methods are optimal choices of the algorithms.

Figure 10 and figure 11, which are reproduction of figure 16 in the original paper, illustrates the linear relationship between learning times and simulation time. The number of iterations differs based on the number of nodes. As illustrated, by increasing the number of nodes, simulation time increases. Figure 10 and figure 11 are by implemented using Q-Learning and SARSA respectively. Simulation time in SARSA is much higher than the Q-Learning method, which can be seen in the larger number of nodes.

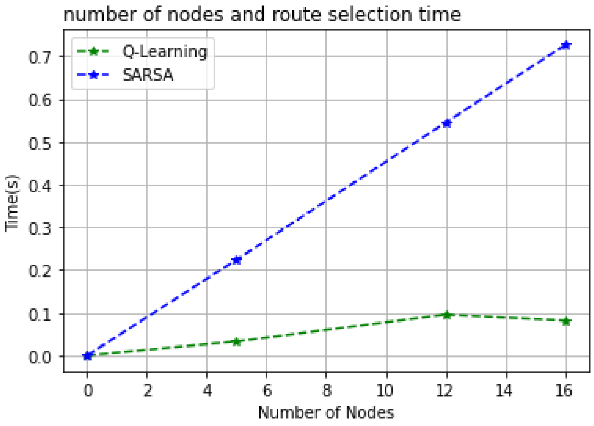


Fig. 9: relation between number of nodes at intersection and route selection time.

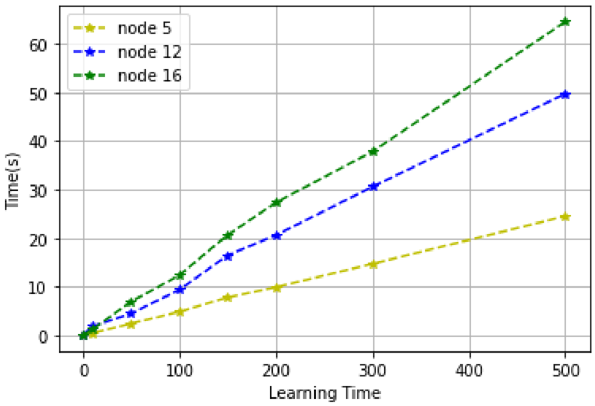


Fig. 10: Q-Learning: Relationship between learning times and simulation time.

To reproduce figure 18 of the original paper, the variables needed for the equations in the paper were not sufficient. In this regard, we considered some variables based on other papers and reproduced Air resistance for the vehicles. These variables can be found in the System Model Section. The simulated air resistance of the vehicle platoon is shown in the figure 12

The first graph in the figure 12 depicts a normal vehicle in the normal condition. Other graphs illustrate bigger vehicles with double greater size compared to the previous vehicle in two different environments. Graph 3 has a larger air density and resistance coefficient. As the distance between vehicles increases, air resistance increases and as a result, we would witness air-drag reduction, which is shown in figure 18 of the original paper.

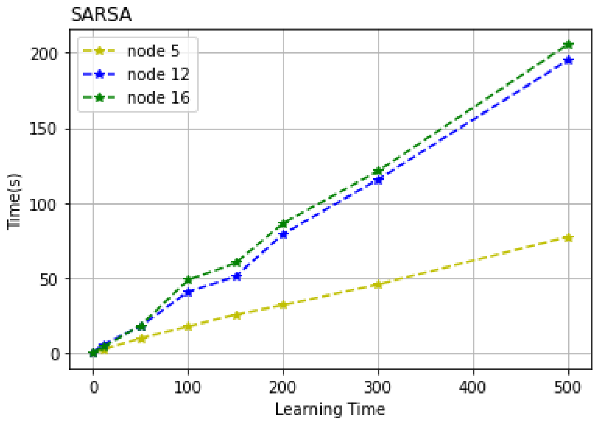


Fig. 11: SARSA: Relationship between learning times and simulation time.

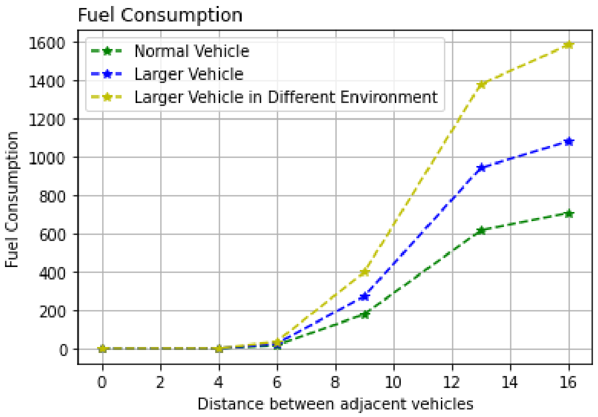


Fig. 12: Air resistance in three different conditions for different vehicle sizes.

VI. Conclusion

The purpose of this paper is to find the optimal path for a vehicle platoon to reduce fuel consumption. The platoon formation reduces the air resistance between vehicles and in turn, reduces the fuel consumption. All the vehicles in the platoon start travelling from the same point. Assuming that the starting point and the destination point are known, reinforcement learning is used to train the platoon to find the optimal path according to road traffic conditions and distances. For this purpose, Q-learning and SARSA algorithms are combined with ε-greedy strategy and by using the time, distance, and other measurements to reflect the road condition, a reliable algorithm is developed.

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] 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.

[3] 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.

[4] V. Turri, B. Besselink, 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.

[5] A. García Castro and M. D. C. Andrés, “Measuring the effects of traffic congestion on fuel consumption,” in in Proc. Transp. Res. Board Conf., TRB, Washington, DC, USA, 2014, pp. 19–22.

[6] R. Wang, Z. Xu, X. Zhao, and J. Hu, “V2 V-based method for the detection of road traffic congestion,” IET Intell. Transp. Syst., vol. 13, no. 5, pp. 880–885, May 2019.

[7] R.s. Sutton and A.G. Barto, ”Reinforcement Learning : An Itroduction’

[8] C. Chen, J. Jiang, N. Lv and S. Li, ”An Intelligent Path Planning Scheme of Autonomous Vehicles Platoon Using Deep Reinforcement Learning on Network Edge,” in IEEE Access, vol. 8, pp. 99059-99069, 2020, doi: 10.1109/ACCESS.2020.2998015