Traffic & Transportation

Improving Traffic Efficiency in a Road Network by Adopting Decentralised Multi-Agent Reinforcement Learning and Smart Navigation

Hung Tuan TRINH[1], Sang-Hoon BAE[2], Duy Quang TRAN[3]

Original Scientific Paper
Submitted: 15 Mar. 2023
Accepted: 19 July 2023

[1] trinhhung@pukyong.ac.kr, Smart Transportation Lab, Department of Spatial Information Engineering, Pukyong National University

[2] Corresponding author, sbae@pknu.ac.kr, Smart Transportation Lab, Department of Spatial Information Engineering, Pukyong National University

[3] duytq@ntu.edu.vn, Faculty of Civil Engineering, Nha Trang University

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT

In the future, mixed traffic flow will consist of human-driven vehicles (HDVs) and connected autonomous vehicles (CAVs). Effective traffic management is a global challenge, especially in urban areas with many intersections. Much research has focused on solving this problem to increase intersection network performance. Reinforcement learning (RL) is a new approach to optimising traffic signal lights that overcomes the disadvantages of traditional methods. In this paper, we propose an integrated approach that combines the multi-agent advantage actor-critic (MA-A2C) and smart navigation (SN) to solve the congestion problem in a road network under mixed traffic conditions. The A2C algorithm combines the advantages of value-based and policy-based methods to stabilise the training by reducing the variance. It also overcomes the limitations of centralised and independent MARL. In addition, the SN technique reroutes traffic load to alternate paths to avoid congestion at intersections. To evaluate the robustness of our approach, we compare our model against independent-A2C (I-A2C) and max pressure (MP). These results show that our proposed approach performs more efficiently than others regarding average waiting time, speed and queue length. In addition, the simulation results also suggest that the model is effective as the CAV penetration rate is greater than 20%.

KEYWORDS

multi-agent reinforcement learning (MARL); multi-agent advantage actor-critic (MA-A2C); deep reinforcement learning (DRL); deep neural network (DNN); connected and autonomous vehicles (CAVs); traffic signal control.

1. INTRODUCTION

With rapid socio-economic development, cities are becoming more and more developed and are continuously expanding. Personal cars are becoming a big challenge, putting much pressure on traffic management [1]. Currently, the problem of traffic congestion in large urban areas causes many issues for road users, including environmental pollution and economic losses [2]. Now, there are many studies focused on solving the problem of traffic congestion. These studies have provided new solutions to increase the efficiency of existing transport infrastructure without expanding the road network. To improve traffic performance, the leading solutions focus on the problems by a) optimising signal lights at intersections and b) navigating routes (rerouting vehicles) to avoid congested areas.

One of the direct ways to reduce traffic congestion is to manage signal lights at intersections. It is an effective method for areas where lanes cannot be widened. Currently, most cities use traditional fixed-time signal control (FTSC) to manage traffic at intersections [3]. This method applies a fixed phase and signal cycle length. However, it becomes less effective for highly dynamic traffic environments. It especially cannot solve congestion in areas with many intersections [4]. Therefore, we must develop a reasonable way to control a multi-intersection network in a highly dynamic traffic environment.

As cities expand, smart solutions are needed to improve traffic efficiency. In recent years, reinforcement learning (RL) has been studied and applied in several fields, such as robots or game management [5]. Many researchers utilised RL and multi-agent systems to control traffic at signalised or unsignalised intersections [6–9]. RL, formulated according to the Markov decision process (MDP), is an emerging solution to solve traffic problems at intersections based on actual traffic experience. Unlike traditional approaches, it is not based on heuristic assumptions or formulas. Instead, it directly learns optimal controls based on its actual experiences as agents act in the environment [10].

Although deep reinforcement learning (DRL) is successfully applied at an isolated intersection, its application to multi-intersections is still a big challenge. In the real world, the congestion at intersections is different and intersections affect each other. Each intersection cannot know the traffic states of other intersections because it only observes a part of the traffic environment. Multi-agent reinforcement learning (MARL) algorithm is applied to manage traffic at multi-intersections, and the most significant challenge is the coordination of agents between them [11]. The simple model is independent Q-learning (IQL), where each local agent learns its policies independently. IQL is scalable, but it is difficult to converge because the environment is non-stationary. They update models based on stationary transitions, which is unlikely at intersections. In contrast, the policy--based method updates the model directly without using a value function. Advantage actor-critic (A2C) is a hybrid architecture that combines policy-based and value-based methods [12]. This new solution overcomes the aforementioned limitations and has outstanding advantages, such as stability during training, faster training and reduced variance.

Besides managing intersections by RL algorithms, the smart navigation (SN) technique is also an effective solution to reduce congestion. In case of congestion at intersections, the controller systems adjust the CAVs to alternate paths with less traffic. The rerouted vehicles choose another route that may be longer to avoid congestion. With the development of wireless communication, traffic controllers can easily receive signals and navigate vehicles to increase traffic efficiency.

Recently, big companies like Google [13] or TomTom [14] have started using infrastructure-based data to navigate traffic. However, these solutions are not really effective. Their framework is still based on the shortest route algorithm without considering current impacts such as traffic jams or waiting times due to red lights. Traffic information such as average speed, waiting time or traffic impacts of signal lights has not been utilised to optimise the route. In addition, another consequence is that if too many vehicles use a route, there will be congestion at that route while the other directions are clear. However, this situation can be avoided by using SN to balance traffic on all routes.

In this paper, we combined a novel MA-A2C algorithm with SN to optimise traffic performance on a hypothetical road network in mixed traffic conditions. The model results are evaluated with two optional benchmarks, independent advantage actor-critic (I-A2C) and max pressure (MP). Our main contributions to this paper are as follows:

- We proposed a novel MARL model to solve the traffic congestion problem in a road network under mixed traffic flow. Our model is based on the A2C algorithm, which combines the advantages of value-based and policy-based methods. This new solution stabilises the training by reducing the variance. In addition, it also overcomes the limitations of centralised and independent MARL. Our model results are significantly improved compared with two optional standards, I-A2C and MP.
- We also controlled and rerouted CAVs to avoid congested areas. SN technique is used to balance the traffic before it approaches intersections. When congestion occurs at intersections, the controller assigns CAVs to alternate paths to reduce traffic load on the congested intersections.
- We also considered the effect of the CAV penetration rate on the model's performance. We have conducted experiments with CAV penetration rates of 10, 20, 40, 60, 80 and 100%, respectively. Model performance is improved with a high market rate.

Our paper consists of 6 sections. Section 2 introduces the literature review. Section 3 shows the methodology. Section 4 presents the experimental setup. Section 5 is about the results of traffic simulation and evaluations. The last section makes a conclusion on the experimental results.

2. LITERATURE REVIEW

2.1 Traffic signal control for road network

As we all know, traffic management plays a critical role in improving traffic efficiency. Traditional intersection management methods (fixed time or stop signs) are gradually being replaced. The new adaptive traffic signal controller (ATSC) uses loop detectors and their algorithms to control traffic at intersections. When traffic conditions change, ATSC automatically adjusts the cycle and phase duration according to the given rules. The performance of traffic management depends on the optimisation algorithm applied. In addition, there are other algorithms to manage intersections, such as evolutionary algorithms [15], MP [16], self-organisation [17], the split-cycle offset optimisation technique (SCOOT) [18] and the Sydney coordinated adaptive traffic system (SCATS) [19]. However, these methods depend on the reliability and accuracy of traffic sensors. And their optimal algorithms are largely heuristic and sub-optimal.

Several DRL models have been proposed for traffic signal management because of their large state space capabilities [20]. Deep Q-learning is a combination of two algorithms, DNN and Q-learning. These models have been used in many studies to optimise signal lights and minimise waiting time [21, 22]. These models are only applied to a single intersection without considering the influence of adjacent intersections. Despite the power of the DQN model, its results are often too over-optimistic. Also, they can only handle discrete action space. If the number of actions is large, the algorithm seems to converge to the maximum local value rather than the best value.

Fortunately, the policy gradient (PG) algorithm (i.e. proximal policy optimisation (PPO)), deep deterministic policy gradient (DDPG) and trust region policy optimisation (TRPO) from a different branch of RL can manipulate the continuous action space. Based on policy optimisation, Tran et al. utilised PPO for training autonomous vehicles and increasing traffic performance in mixed traffic conditions [23]. However, the PG-based method has high variance, slow convergence and requires many samples [24]. Because of the shortcomings of the two methods based on PG and value, in some papers, the actor-critic algorithm is proposed to combine the advantages of the two methods [25, 26]. Simulation results have demonstrated the effectiveness of the proposed A2C model compared to other methods.

Previous studies demonstrated the applicability of DRL in traffic management at a single intersection. To overcome the shortcomings of single-agent issues, MARL is developed to solve real-world problems. The major challenge in MARL is the cooperative problem between agents, where agents react to the environment individually, and their behaviour may not be optimal [27]. MARL studies are generally classified into main categories: independent, centralised and decentralised algorithms. Following the first approach, the models in [28] assume that agents learn independently in local environments without coordinating with other neighbouring agents. Each agent gets local information and only controls the signal lights at this intersection. This assumption simplifies the problem and reduces the computation time but also decreases the model performance. In another approach, Prashanth et al. used centralised control to train a global agent of a road network [29]. The disadvantage of this approach is that the size of the state and action space grows exponentially as the number of actors increases.

To verify these algorithms, authors in [30] introduced two modes of traffic management: the independent mode and the integrated mode, with coordination between agents. Moreover, Ge et al. mentioned a cooperative deep Q-network with Q-value transfer for adaptive multi-intersection signal control, where agents share their information about the last action [31]. In [7], the authors developed a decentralised MARL algorithm with an A2C algorithm to optimise road networks. The MA-A2C is compared against independent-A2C and independent Q-learning (IQL) algorithms in terms of delay time and queue length. This model is more efficient than other MARL algorithms. They provided a flexible and convenient way to solve complex traffic problems at multi-interchange networks.

2.2 Traffic navigation

As mentioned in the previous part, traffic navigation is also crucial for increasing traffic efficiency. The first solutions for vehicle navigation based on the shortest route were developed from Dijkstra [32] or A* algorithms [33]. They give the shortest paths without considering the influence of other factors such as average velocity, waiting time or congestion. Therefore, they are not suitable when applied to signalised or congested

areas. Another way to navigate vehicles is to use the ant colony algorithm [34]. This algorithm is also adapted and applied in other research based on distinctive preferences to give the optimal routes. However, it has the limitation that the number of agents (ants) is correlated directly with the algorithm performance. Therefore, the model results are not good in the case of limited agents. With the development of machine learning technologies, many researchers have applied DRL to solve the problem of finding optimal routes. In [35], the authors proposed a navigation framework based on DRL to find the best route and avoid congestion. Thus, the paper [35] develops a method to build a real-time intelligent vehicle navigation system and test its algorithm in nine real scenarios. However, all scenarios in the model are simple without considering the influence of traffic control systems such as signal lights. In another approach, the authors give a new model to predict congestion and then select vehicles to reroute [36]. However, determining thresholds for predicting congestion and traffic navigation is complex, and model performance is highly dependent on these values.

2.3 Limitations

It can be seen that traffic management in the road network is very critical. MARL and traffic navigation have been studied extensively and have achieved great success in many areas, including transportation. However, these models do not always work well in multi-intersection networks. The problems can be defined as follows: (1) some models are only suitable for a single intersection and do not fit networks with many actions and grid states; (2) some reward functions in some models have no relation to the state space; (3) The road users often choose the shortest path based on their estimation. When the number of vehicles on the road exceeds its capacity, it will cause local congestion; and (4) most studies often assume that the vehicles are CAVs, and data on all vehicles are fully collected. This is not true in practice. To differ from that, we tested models in mixed traffic conditions.

To solve these problems, we combine a novel MA-A2C model with SN to solve the traffic congestion problem in mixed traffic conditions. In which, the MA-A2C model controls the signal lights at intersections while the SN technique reroutes CAVs to avoid the congested area. Our MA-A2C model integrates the benefits of a value-based approach and a policy-based approach and achieves higher performance.

3. METHODOLOGY

3.1 Research architecture

In our models, we combined two methods, MARL to optimise signal lights and SN to balance traffic across routes. MARL uses a systematic framework to solve complex high-dimensional tasks and could be applied to dynamic traffic environments. Through the learning process, agents choose actions to maximise rewards, i.e. the controller selects phase and green time at each intersection to alleviate the waiting time of all vehicles. Our model is based on the A2C algorithm, which combines the advantages of value-based and policy-based methods. The environment in the model is the traffic network, including vehicles, signalised intersections and

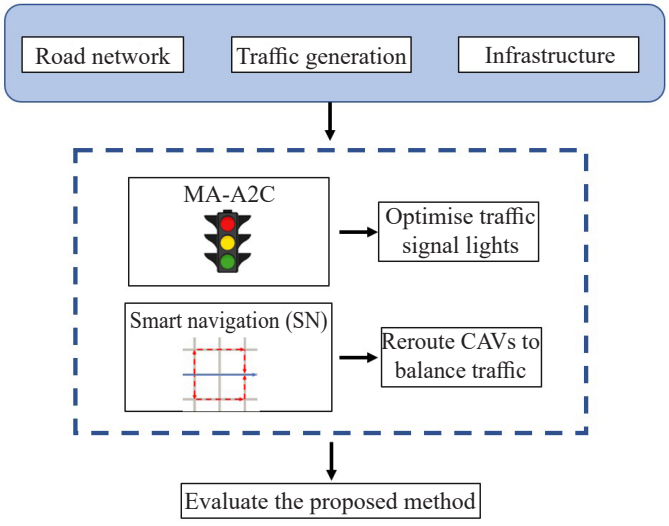


Figure 1 – Framework of the proposed traffic management approach

other road infrastructures. Signalised intersections are consider as agents in our models. To further improve the traffic performance, we also implemented SN to avoid congestion and increase throughput. The framework of our model is expressed in *Figure 1*.

3.2 Reinforcement learning (RL)

RL is a type of machine learning technique that allows an agent to learn an optimal policy in an interactive environment by using feedback from its own actions and experiences. The best optimal policy is defined as the one that receives the most expected cumulative rewards. It uses rewards and punishments as signals for positive and negative behaviours. RL is applied in many fields, such as control theory, optimisation, multi-agent systems, etc. Parameters in our models are shown in *Table 1*.

Table 1 – Parameters used in our models

Symbol	Meanings
s_t and S	Agent's state at time t and state space S
a_t and A	Agent's action at time t and action space A
$R_a(s,s')$	The immediate reward when the agent transitions from the state S to the new state S' with action A
$P_a(s,s')$	The probability of transition (at time t) when the agent changes from the state S to the new state S' after performing action A
π	Policy
γ	Discount rate
$V_\pi(s)$	State function
$Q_\pi(s,a)$	State-action function under policy π
$G(V,E)$	The multi-agent network where V, E are two sets of agents and edge space
$k \in V$	An agent k (signalised intersection) in the multi-agent network
$a_k \in A$	Action of agent k
$e^{kj} \in E$	Edge in the multi-agent network between agent k and adjacent agent j
N_k	Neighbourhood of agent k
V_k	Local region of agent k ($V_k=N_k \cup k$)
$L(\omega_k)$	Value loss function of agent k
$L(\theta_k)$	Policy loss function of agent k
F	Advantage function

The main characters of RL include the agent and the environment. In RL, the environment is often modelled as MDP with the interactions between the agent and environment expressed by a four tuple M = (S, A, R, P).

The purpose of RL is to learn the optimal policy to maximise the reward. At time t, the agent interacts with its environment and takes an action from the set of available actions. The environment moves from state s_t to a new state s_{t+1} and the agent takes a reward r_{t+1} corresponding to transition (s_t, a_t, s_{t+1}). Based on reward and punishment, the agent continues to learn until it reaches the highest reward. The agent tries to learn a policy $\pi : A x S \rightarrow [0,1], \pi(a,s) = Pr(a_t = a | s_t = s)$ to maximise the reward. In each time period, the agent performs an action to control traffic according to a specific rule, and it interacts with the environment for a certain period of time. The goal of the agent is to optimise the reward accumulated over the time T.

3.3 Actor-critic

This is the solution to reduce the variance of RL and train agents faster and better. As shown in *Figure 2*, this algorithm combines two main components: actor-network and critic-network. The actor-network employs the

policy-gradient algorithm to generate actions and interact with the environment. On the other hand, the critic network utilizes Q learning to assess the actor's performance and provide guidance for the actor to the next action.

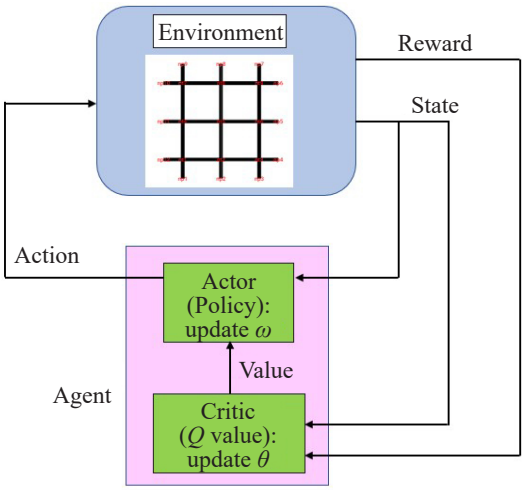


Figure 2 – Actor-critic reinforcement learning framework

At each time step t, the agent is at state s_t and passes it as input through the actor and critic network. The policy takes the state and outputs an action a_t. The critic takes action a_t and uses state s_t to compute the value of taking that action at this state $\hat{q}_\omega(s_t, a_t)$. The agent gets a reward r_{t+1} and changes to the new state s_{t+1}.

3.4 Multi-agent reinforcement learning (MARL)

Road network management in the real world involves optimising multiple intersections simultaneously. The issue of cooperation between intersections (agents) is important because each intersection (agent) takes actions affecting the traffic at the adjacent intersections. Good cooperation makes vehicles go through intersections faster, reducing waiting time when stopping at red lights. MARL is used to solve cooperation issues between agents.

We model a road network $G(V,E)$ consisting of many intersections (agents), in which each agent executes a discrete action $a_k \in A_k$ and connects to surrounding agents via edges $e^{kj} \in E$ (edge between agent k and the neighbor agent j) and shares the global reward $r(s,a)$. Centralized RL is infeasible because of the size of the joint action space. This size increases exponentially with the number of agents. It is assumed that the global Q value function is decomposed for each agent as follows $Q(s,a) = \sum_{k \in V} Q_k(s,a)$ and each local agent is able to observe the global state.

Global cooperation is not always necessary, and it depends on traffic volume. When the traffic density is low, the optimal policy is to implement decentralized greedy control at each agent. When the traffic density is high, cooperation between agents is necessary to optimize the traffic flow.

Global cooperation is not always necessary, and it depends on traffic volume. When the traffic density is low, the optimal policy is to implement decentralised greedy control at each agent. When the traffic density is high, cooperation between agents is necessary to optimise the traffic flow.

3.5 Independent A2C (I-A2C)

In the case of IA2C, each agent learns its own policy π_θ and the corresponding value function V_{w_i} independently without considering the effects of other agents. It is a simple and popular approach to solve the problems of MARL. It is assumed that the global reward and state are shared between agents. I-A2C is the extension of centralized A2C, in which the local return of agent k is defined as follows:

$$R_{t,k} = \hat{R}_t + \gamma^{t_B - t} V_{w_k} - (s_{t_B} \mid \pi_{\theta_{-k}}) \tag{1}$$

Each return $\hat{R}_t = \sum_{\tau=0}^{t_B-1} \gamma^\tau r_\tau$ is sampled from the same stationary policy π_{θ^-} that makes the value gradient $\nabla L(w_k)$

consistent. The value function $V_{w_k}: \mathcal{S} \times A_k \rightarrow R$ is the estimation of the marginal impact of policy to calculate policy gradient $\nabla L(\theta_k)$. The identical value gradient $\nabla L(w_k)$ targets the global value function V^{π} rather than the local one $V^{\pi k} = \mathrm{E}_{\pi - k} V^{\pi}$. If the weights θ_{-k} is fixed, the policy converges to the optimal policy $\pi^*_{\theta_k}$ with $\theta_{-k} = \theta^*_{-k}$ after updating. Conversely, if θ_{-k} is updated, the policy gradient $\nabla L(\theta_k)$ cannot be consistent as the advantage function depends on changing the policy $\pi_{\theta_{-k}}$. Communication is assumed to be limited in each local region υ_k, meaning that the policy and value regressor use $s_{t,\upsilon_k} = \{s_{t,j}\}_{j \in \upsilon_k}$ as input state rather than s_t.

The value loss function is expressed below:

$$L(w_k) = \frac{1}{2|B|} \sum_{t \in B} \left(R_{t,k} - V_{w_k}(s_{t,\upsilon_k}) \right)^2 \tag{2}$$

The policy loss function:

$$L(\theta_k) = -\frac{1}{|B|} \sum_{t \in B} \log \pi_{\theta_k}(a_{t,k} \mid s_{t,\upsilon_k}) \left(R_{t,k} - V_{w_k^-}(s_{t,\upsilon_k}) \right) \tag{3}$$

3.6 Multi-agent A2C (MA-A2C)

In this approach, data of neighboring polices is included to enhance the observation of each local agent. This is the main difference of this method compared to the previous method (I-A2C). Accordingly, the latest sample policies of neighbors are included in the input of DNN with the current state. The sampled local policy is defined as

$$\pi_{t,k} = \pi_{\theta_k^-}(. \mid s_{t,\upsilon_k}, \pi_{k-1}, N_k) \tag{4}$$

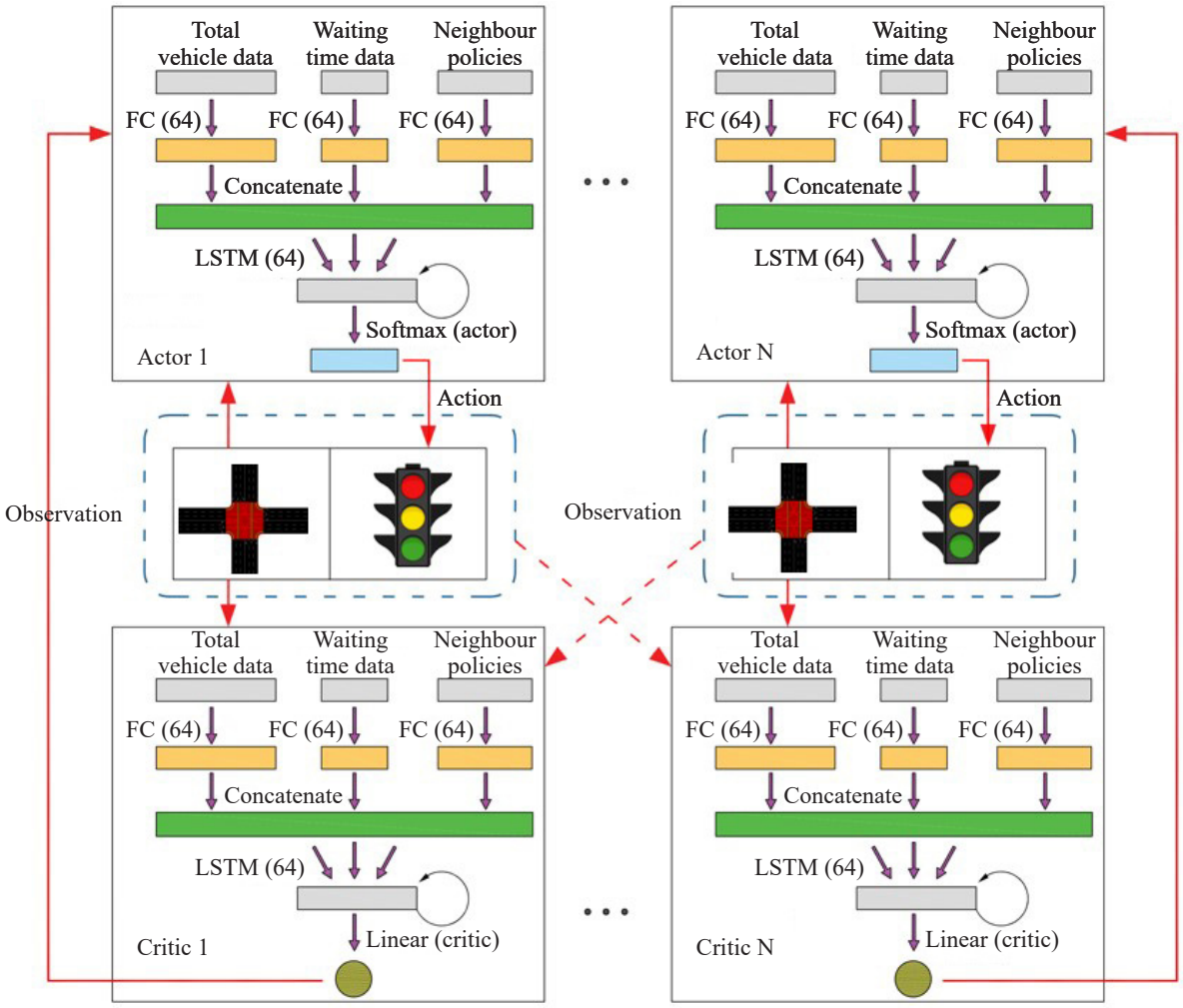


Figure 3 – MA-A2C Framework

The real-time current neighborhood policy is transmitted to each local agent. This is based on the fact that (1) the current policy is similar to the previous policy because the traffic state changes slowly. (2) the traffic state is MDP.

The global reward is assumed to decompose into local rewards in global cooperation as $r_t = \sum_{k \in V} r_{t,k}$. The global reward for agent k is adjusted with the spatial discount factor α.

$$r_{t,k} = \sum_{d=0}^{D_k} \left(\sum_{j \in V | d(k,j)=d} \alpha^d r_{t,j} \right) \tag{5}$$

The value loss function in *Formula 2* becomes:

$$L(\omega_k) = \frac{1}{2|B|} \sum_{t \in B} \left(\tilde{R}_{t,k} - V_{\omega_k}(\tilde{s}_{t,\upsilon_k}, \pi_{t-1}, N_k) \right)^2 \tag{6}$$

The policy loss function in *Formula 3* becomes:

$$L(\theta_k) = -\frac{1}{|B|} \sum_{t \in B} \left(\log \pi_{\theta_i}(a_{t,k} | \tilde{s}_{t,\upsilon_k}, \pi_{t-1}, N_i) \tilde{F}_{t,k} - \beta \sum_{a_k \in A_k} \pi_{\theta_k} \log \pi_{\theta_k}(a_k | \tilde{s}_{t,\upsilon_k}, \pi_{t-1}, N_k) \right) \tag{7}$$

The actor and critical DNNs are trained separately as shown in *Figure 3*.

3.7 Smart navigation technique

In addition to traffic signal management, route navigation is also especially crucial to improve model efficiency. This approach can be implemented by taking advantage of vehicle-to-infrastructure (V2I) communication technique. In this model, this technique is applied to CAVs that have not yet reached the congested area but are coming towards it. As shown in *Figure 4*, from west to east CAVs have 3 routes to go from original point to destination point. One route is a straight line (cyan line), and the other 2 routes are rerouting paths (red line). From the data collected by CAVs, the controller can request CAVs to reroute to avoid congestion if it occurs.

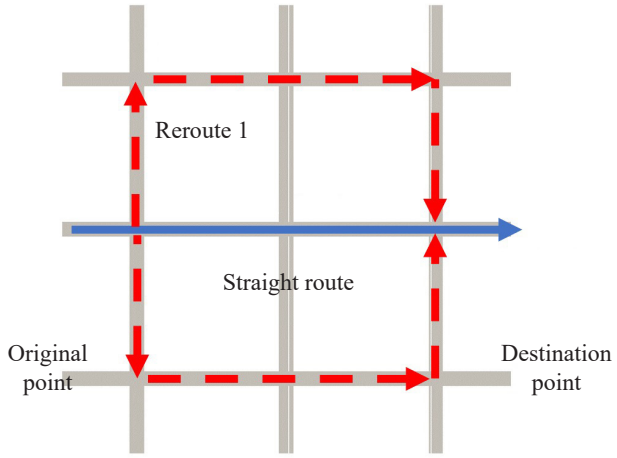


Figure 4 – An example of rerouting for vehicles travelling from west to east

Vehicle navigation is implemented based on the principle that vehicles choose the fastest routes instead of the shortest routes. Typically, the travel time per route is defined as the key parameter for vehicle rerouting. First, we took the original destination (OD) matrix of the traffic and determined the possible routes from the original to destination points. The shortest route is the line directly connecting those two points according to the Dijkstra algorithm [32]. Vehicles always choose the shortest route to travel in normal conditions. In the case of congestion, the system shifts CAVs to alternate routes with the shortest travel time instead of the default route.

In traffic engineering, the travel time of a vehicle is the amount of time required to travel from the original point to the destination point on a given route. It is defined as the sum of running time and traffic signal delay.

The speed of each vehicle varies because of congestion and waiting time at intersections. The average travel time on each route can be computed as follows:

$$Average\ travel\ time = \frac{\sum travel\ time}{Total\ vehicles} = \frac{1}{N_{veh}} \sum_{k=0}^{N_{veh}} (-t_{k,start} + t_{k,end}) \tag{8}$$

where N_{veh} is the total vehicles on the route, $t_{k,start}$ and $t_{k,end}$ are the time of the vehicle k that enters and exits the route.

4. EXPERIMENTAL SETUP

In this paper, we use DRL to manage traffic lights in the road network with the aim of reducing waiting line length and delay time at intersections. Traffic data at intersections are collected through vehicular network at each intersection according to different environments. Details of MA-A2C implementation, including the setting of state, actions, rewards and DNN are presented in this section.

4.1 Road, lane and configuration

We simulated the road network by simulation of urban mobility tool (SUMO) with different scenarios [37]. It is the open-source microscopic traffic simulator used in the world. The proposed algorithms are implemented by Python programming and the Tensorflow module. The traffic control interface (TraCi) protocol uses a TCP--based client/server architecture to access and get values from SUMO. Vehicle communication is implemented through TraCi to get the data needed to train the traffic management models. We use a detector-based method to simulate vehicle communication as it can change the data collection range (50 m around each intersection) by varying the detector length. Data from CAVs are collected from virtual detectors and they are processed at the traffic centre. Then, they will be transmitted to the CAVs via vehicle-to-infrastructure (V2I) communication.

The road network consists of 9 signalised intersections, and each road has 2 lanes (one lane for turning left, and one shared lane for going straight and turning right), as shown in *Figure 5*. The distance between intersections is 300 m and the width of each lane is 3.2 m. Mixed traffic flow is used in the model, and it includes two components, CAVs and HDVs. Through V2I, only data from CAVs are collected during training.

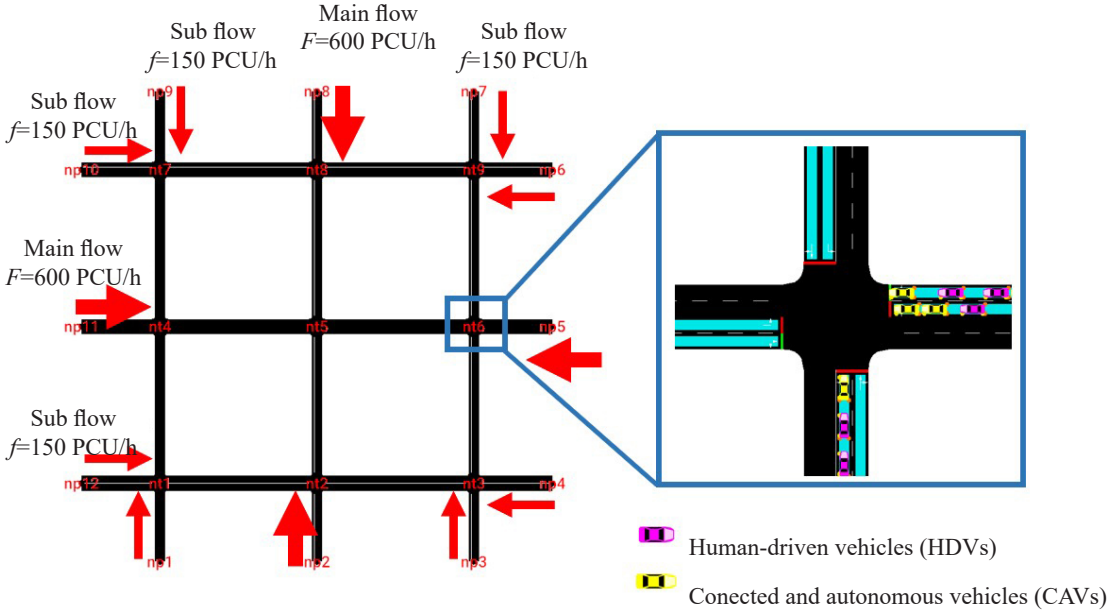


Figure 5 – Road network with 9 signalised intersections

To consider the effect of CAVs on model results, we take several simulation scenarios with different CAV penetration rates and traffic flow. Traffic generation is an important issue affecting model performance. In our model, CAVs and HDVs are generated simultaneously in certain proportions to ensure random traffic flow. Traffic volume gradually increases and reaches a maximum and then decreases towards the end. This process is done within 1 hour with a total traffic of 3,600 vehicles for the whole network.

4.2 MA-A2C settings

State space

The state s_t of an agent is a representation of the environment at a given time t and relates to the reward function. Depending on the purpose of the models, the state needs to provide enough information about traffic conditions in all directions so that the agent can optimise signal lights effectively. Redundant data increases the computational cost and training time of the model. On the other hand, not enough data makes the model results inaccurate. The local state at each intersection in our models is defined as

$$s_{t,i} = \{waiting\_time_t[lane], total\_vechiles_t[lane]\} \tag{9}$$

where *lane* is each coming lane of intersection *i*, *waiting_time* [s] measures the cumulative waiting time of the first vehicle, *total_vehicle* [veh] measures the total number of approaching vehicles along each incoming lane.

Information data of CAVs collected near the intersection are more important than information data far away. In addition, redundant data increase the calculation and take more time to train. Therefore, in our model, data are collected within 50 m around the intersection.

Action space

In our traffic management approach, actions are defined as the signal phase of traffic lights. This definition allows the agent to manage actions more flexibly. Local actions at each intersection are defined based on signal phase and are shown in *Figure 6*. Based on the state information of each agent, the green time of each action is also different.

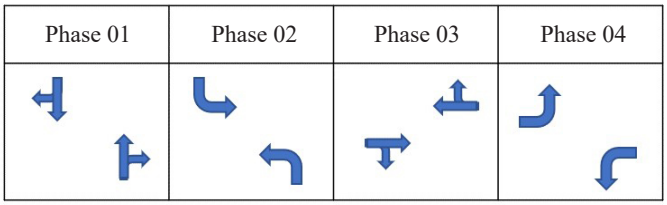


Figure 6 – Traffic signals control phase at intersections

In this approach, we set the reaction time between agents and environment as Δ_t. At each time step Δ_t, the agent chooses one action from the set of actions to perform. If the action taken is the same, the green time is extended. Otherwise, the signal phase turns to yellow time, and the agent chooses the new action to perform. The duration of Δ_t directly affects the green time of each action. If Δ_t is small, it increases the computation cost and shortens the green time. On the contrary, if it is too large, it makes the green time of each phase longer and prolongs the queue length in other directions.

Reward definition

The reward is an important part of the model because it defines the training goal. Through the reward, the agent understands the outcome of the action and will improve the model for the next choice. The purpose of the model is to optimise traffic and improve performance. Therefore, the reward is determined based on the waiting time and the queue length of all vehicles. It is defined as follows:

$$r_{t,i} = -\sum(queue\_length_{t+\Delta t}[lane] + v.waiting\_time_{t+\Delta t}[lane]) \tag{10}$$

where *queue_length* [veh] is the queue length of vehicles along each incoming lane, v [veh/s] is the trade-off coefficient, *waiting_time* [s] is the cumulative waiting time of the first vehicle, the queue length and waiting time values are measured at time $t+\Delta t$, *lane* is each incoming lane of intersection *i*.

DNN structures

In real life, if the agent only knows the current state, the MDP can become non-stationary. If all history states were inputted to A2C, it would increase state size significantly and reduce the focus of the model on the current state. We use long-short term memory (LSTM) in our model as the last hidden layer to extract representation from various types of states. The input data of the model MA-A2C include *waiting_time, total_vehicle*

and *neighbour policies*. They are processed by fully connected layers (FC) and concatenated in an array. Then they all are input to the LSTM layer. Next, the output of the LSTM is used as the input to the actor and critic networks. The final output includes the *Softmax* function for the actor and the *Linear* function for the critic. These activation functions are used to transform the summed weighted input into output. The actor uses the *Softmax* function to produce an output with a stochastic probability distribution.

In our model, each agent with its actor and critic network learns its policy separately, instead of sharing the last layers (LSTM). The input data of each agent are the local agent's observation at the intersection according to *Equation 9*. Based on the action taken, each agent receives a reward from the environment (*Equation 10*). The difference between MA-A2C and I-A2C structure is the appearance of the *neighbour* unit. In I-A2C, agents only learn their own policies independently. It means that the model input consists of only *waiting_time* and *total_vehicle* states. In MA-A2C, model input adds *neighbour* policies to improve the observability of each agent.

Hyperparameters are important parameters used to control the training process. They need to be identified before training the models. There are no specific rules for determining these hyperparameters. Therefore, we conducted the trial-and-error method to determine the influence of each parameter on the model performance. Some parameters have a significant impact on the results. For example, gamma changing has a more significant impact than layer changing in neural networks. The gamma (γ) determines the importance of future rewards by multiplying them with gamma. In our model, γ=0.99, it makes the agent strive for a long-term reward. Epsilon (ε) is the trade-off factor between exploration and exploitation. We utilise RMSprop to optimise the gradient in DNN. It can adjust the learning rate adaptively in our models.

Episode_length is the period where the agents are trained in the environment. The training time in each episode ends when it reaches a given time (3,600 s). The number of episodes in our models is 600 because it is enough for the models to converge. Different random seeds are generated during model training and evaluation.

We used normalisation to transform features to be on a similar scale. It improves the performance and stability of the model. Information of *waiting_time* and *total_vehicle* states is collected to get an appropriate normalisation. All states and rewards are normalised and clipped to [0,2] and [-2,2], respectively. In *Equation 7*, β balances optimal exploration and advantage PG and is set to 0.01. As if it is too large, the system may diverge. For MA-A2C and I-A2C, states, actions, next states and rewards are stored in batch size for each agent i: $B_i=\{(s_t,u_t,s_{t+1},r_t)\}$. Each batch reflects the experience trajectory of agent i. We defined other hyperparameters $|B|$=120, $\eta_\theta=\eta\omega$=2.5e-4 to train and evaluate models. Choosing the number of layers is challenging in neural network architecture. These layers do not interact directly with the external environment but also affect the model results. Selecting too many layers can make the model over-fitting. Conversely, too few layers cause under-fitting and incorrect results. The final hyperparameters of our models are shown in *Table 2*.

Table 2 – Parameters of models

Parameter	Value
Discount factor (gamma γ)	0.99
Epsilon (ε)	Decaying from 1 to 0.001
Num_LSTM	64
Num_FC	64
Batch_size	120
Total episode	600
Episode_length_second	3600
β	0.01
$\|B\|$	120
$\eta_\theta=\eta\omega$	2.5e-4
Agent	MA-A2C and I-A2C
Interval_second	5
Yellow_interval_second	3

4.3 Route navigation in network

SUMO is a traffic simulation tool only for HDVs. Through TraCi, we utilise loop detectors to extract CAVs' data. At each route, 2 multi-entry-exit detectors are placed at 2 lanes. The total number of multi-entry-exit detectors used for navigating vehicles in the network is 48. These detectors provide information about the number of vehicles, mean speed and travel time during the last time step. These data are analysed every 60 s to determine which CAVs must be rerouted.

We applied two types of route assignment in our model. The initial routes are determined based on the O-D matrix of the vehicles according to Dijkstra's algorithm [32] when the mode is initialised. In the beginning time, vehicles choose these routes to travel. When traffic increases and congestion begins to appear, the controller activates the dynamic route assignment based on a comparison of travel times of these routes. CAVs are navigated to choose the route with the shortest travel time. During training, while MA-A2C manages the signal lights, the SN mode reroutes traffic to avoid congestion.

5. RESULTS AND EVALUATION

To compare the efficiency of traffic control algorithms, we conducted four models with different algorithms: MA-A2C+SN, MA-A2C, I-A2C and MP. All methods use the same actions, states, rewards and traffic volume. Note that the experiments were performed with a CAV penetration rate of 60% as at a low CAV penetration rate, this improvement is not significant.

MA-A2C: All intersections are controlled by traffic light controller and A2C algorithm is used to optimise the traffic signal lights. The agents in the network coordinate with neighbour agents to improve the performance.

I-A2C: Each intersection is optimised by an independent agent. The A2C is applied to the optimal traffic light control policy. No exchange of information between agents in the road network.

Max pressure (MP): It is a method to control signalised intersections based on stabilising the queue length and maximising throughput. This method does not need to know the current and future traffic volume in the network. It only needs real-time data on the queue length in all directions of intersections [12].

MA-A2C +SN: We used the A2C algorithm to optimise the traffic signal lights. The agents in the network coordinate with neighbour agents to improve traffic performance. We also incorporated SN to alleviate congestion and increase throughput.

5.1 Performance of 4 scenarios (MA-A2C +SN, MA-A2C, I-A2C and MP)

The cumulative reward curves

The cumulative reward curves used to evaluate the performance of the models are shown in *Figure 7*. If the value of the reward curve is small, the model has low performance.

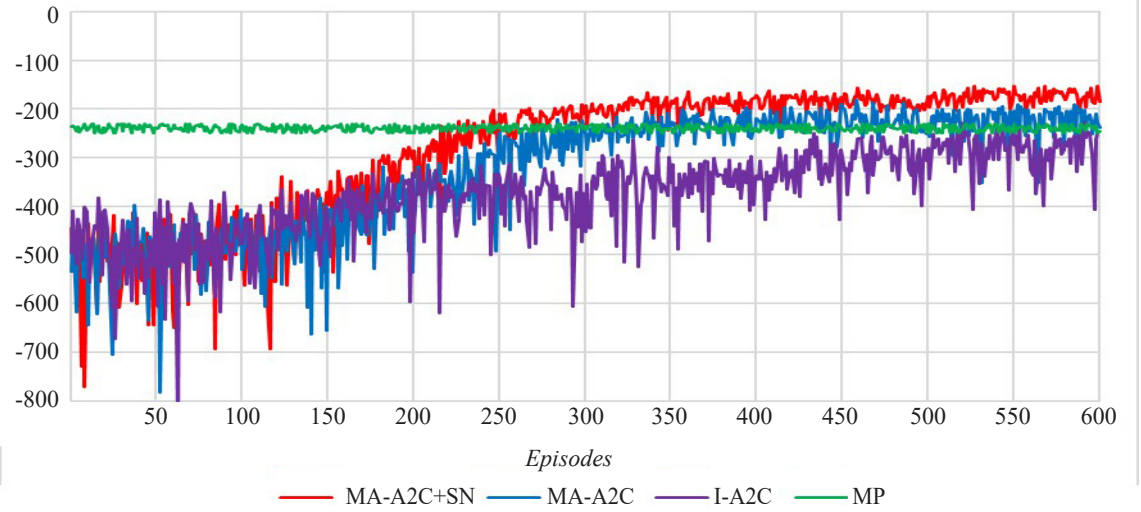


Figure 7 – Cumulative rewards of four scenarios (MA-A2C+SN, MA-A2C, I-A2C and MP)

During training, the reward curve gradually increases for RL based approaches. It means that agents in the models also learn gradually, the queue length and the waiting time of all vehicles decrease over time. This result proves that the models are effective for network optimisation. All models improve traffic efficiency, but the MA-A2C +SN model achieves higher results. After about 300 episodes, this model converges and reaches a constant value of -180. The combination of the 2 approaches (MA-A2C+SN) improves traffic efficiency more than the model based solely on MA-A2C. The I-A2C model has the worst performance and fluctuates greatly when training. After 440 episodes, this model also converges at -300.

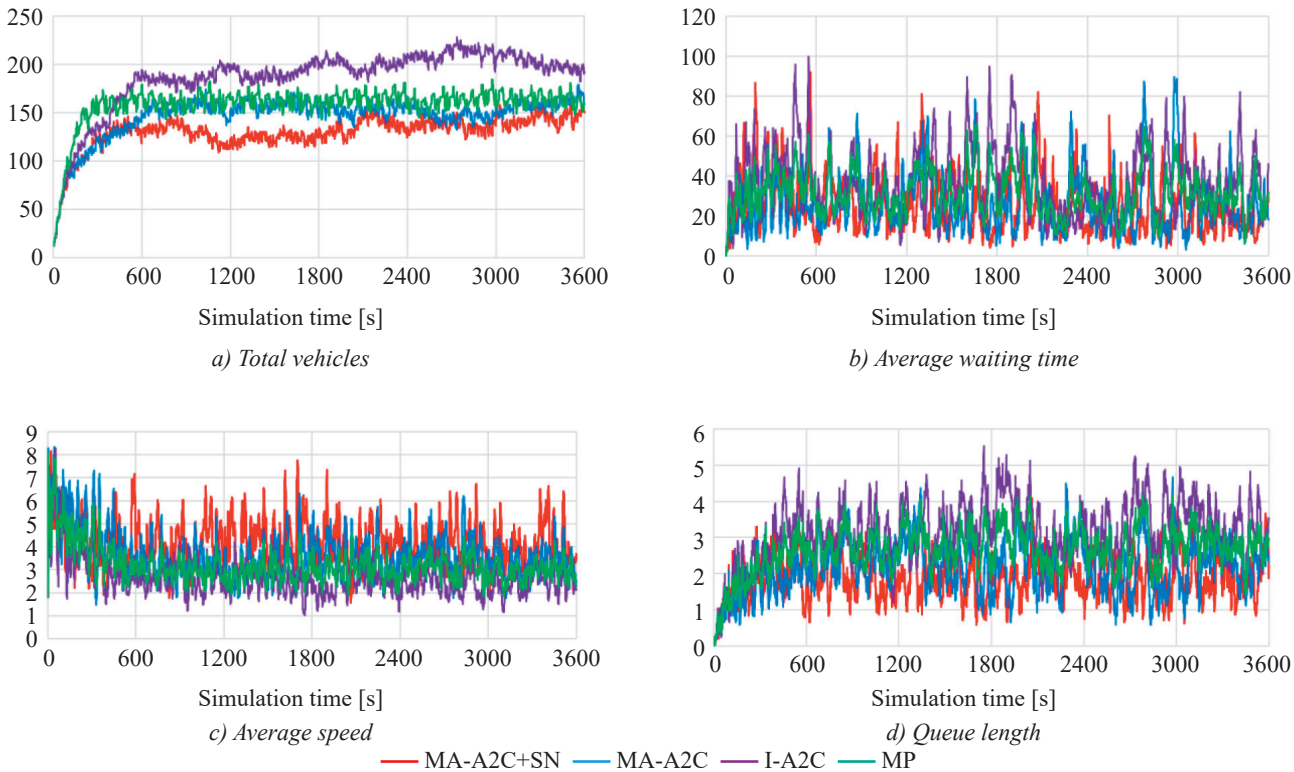
As we can see, the fluctuation of the reward curve also decreases, but the reward curve of the MA-A2C scenario fluctuates less than that of the I-A2C scenario. This can be explained by the fact that the MA-A2C algorithm is more efficient. During the training process, agents in this model share information with the neighbouring agents to help the model quickly achieve the best results. Each agent in MA-A2C is not only aware of its policy but also aware of the policy of its neighbours. This method increases stability during learning by allowing communication between adjacent agents.

In contrast, agents in the I-A2C method only optimise traffic locally at their intersections without communicating with neighbour agents. And its reward curve is unstable with high fluctuation. This difference affects the stability and results of the two models. After training, the reward curve increases by 60% and 30% compared to that before training for MA-A2C+SN and I-A2C cases, respectively.

The results of three RL-based models (MA-A2C+SN, MA-A2C and I-A2C) are improved during training. However, the MP model's results do not change significantly as the number of episodes increases. Because it controls the traffic network based on a predefined algorithm without learning, and its curve reward ranges from -240 to -220 over 500 episodes.

Measures of effectiveness (MOE)

In this section, we present graphs showing the performance metrics in 4 scenarios. The total vehicles, average waiting time, speed and queue length for one hour are shown in *Figure 8*.



8 – Results for the four implemented models (MA-A2C+SN, MA-A2C, I-A2C and MP)

Since the I-A2C method has the lowest reward curve, the efficiency of this model is also the lowest. At the initial time, as the traffic volume in the models is small, the values of average waiting time, speed and queue length in the 4 modes are not much different. These two methods (MA-A2C+SN and MA-A2C) are more effi-

cient due to the coordination between agents in the road network. The I-A2C method has independent agents, so it only optimises the local intersections without paying attention to global optimisation. Therefore, when the traffic volume decreases, the waiting time and the queue length of all vehicles are still high. Other performance results of the four models are shown in *Table 3*.

Table 3 – Performance of four models (MA-A2C+SN, MA-A2C, I-A2C and MP)

Method	Average speed [m/s]	Average duration [s]	Average waiting time per vehicle [s]	Average fuel consumption [mg/s]	Average $C0_2$-emission [mg/s]
MA-A2C+SN	4.15	387.54	47.1	712.36	2,058
MA-A2C	3.86	425.71	53.1	769.69	2,258
I-A2C	2.63	558.56	58.3	857.78	2,684
MP	3.59	481.67	55.2	790.23	2,380

In *Table 3*, average waiting time of MA-A2C+SN mode is reduced by 11% and 18% compared to that of MP and I-A2C mode. The average speed of vehicles of the MA-A2C+SN mode is also improved by 10% and 36% compared to that of the MP and I-A2C modes, respectively.

5.2 Effect of CAV penetration rate in MA-A2C+SN scenario

To test the impact of CAV rate on MA-A2C+SN, we conducted experiments with different penetration rates. We compared 6 scenarios with the same total traffic but CAV rates of 10%, 20%, 40%, 60%, 80% and 100%, respectively. The curve rewards are used to evaluate the efficiency of our models. The model results are shown in *Figure 9*.

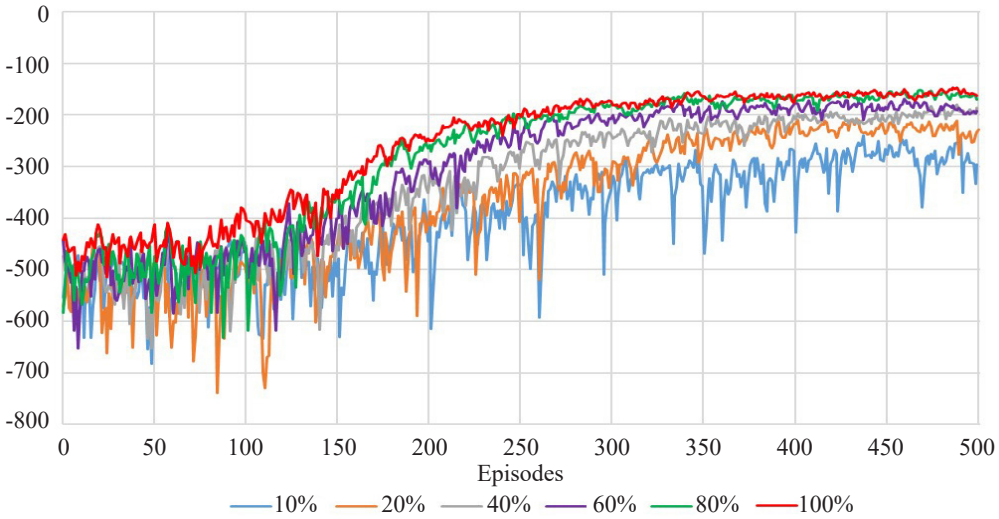


Figure 9 – Cumulative rewards of the MA-A2C + SN model with different CAV penetration rates

It can be seen that model efficiency is proportional to CAV penetration rate. At a low CAV penetration rate, this improvement is not significant. The training process is based on data collected from CAVs. If this rate is too small, it is not enough for the learning process. In addition, when this rate is small, the fluctuation degree of the reward curve is also high. As this rate increases, the information collected from the CAVs also increases, making the learning process improve gradually. The model is most effective when the CAV penetration rate is 100%, meaning that all vehicles in the model are CAVs.

The results also show that the model efficiency is greatly improved when the penetration rate rises from 20% to 40% and the curve reward increases from -300 to -200. Next, this penetration rate accelerates from 40% to 100% and the model efficiency increases from -200 to -150.

In this section, we expressed the performance of CAVs and HDVs under different penetration rates to clarify the advantages of CAVs when the rate increases from 10% to 100%. The results are shown in *Table 4*.

Table 4 – Performance of CAVs and HDVs under different CAV penetration rate

Rate		10%	20%	40%	60%	80%	100%
Average waiting time per vehicle [s]	CAVs	61.52	52.09	48.52	46.65	45.31	43.02
	Total	63.45	55.6	48.84	47.1	45.72	43.02
	HDVs	63.99	57.8	50.09	48.5	47.47	
Average speed [m/s]	CAVs	3.44	3.65	3.86	4.05	4.17	4.13
	Total	3.4	3.61	3.85	4.03	4.15	4.13
	HDVs	3.38	3.58	3.84	4	4.04	

First of all, when the CAV penetration rate increases, the model efficiency also increases. That is, the overall waiting time reduces from 63.45 s to 43.02 s as this rate increases from 10% to 100%; and the total speed also increases from 3.4 to 4.13 m/s. It has also been confirmed that the higher the CAV rate is, the more information is collected, and the model is better optimised.

Another finding is that even though the model is optimal for all vehicles, the performance of CAVs is better than that of HDVs. Because when CAVs approach the intersections, the trained agent switches phases to reduce the waiting time for CAVs. For the approach of HDVs, the trained agent does nothing. It can be seen that agents at intersections only receive information from CAVs and only interact with CAVs.

6. CONCLUSIONS

In this paper, we proposed a novel MA-A2C-based approach combined with SN to solve the traffic congestion problem in a road network in mixed traffic conditions. Data of CAVs are collected through loop detector areas mounted in all directions of intersections. Our A2C model integrates the benefits of a value-based approach (DQN) and a policy-based approach (PG) and achieves higher performance. This new solution stabilises the training by reducing the variance. In addition, it also overcomes the limitations of centralised and independent MARL. Our research results show that MARL is a promising approach for traffic optimisation in a road network.

This proposed method is tested by a micro-simulation model on a network under mixed traffic conditions. To compare the efficiency of traffic control algorithms, we conducted four models with different algorithms: MA-A2C+SN, MA-A2C, I-A2C and MP. After training, the reward curve increases by 60% and 30% compared to that before training for MA-A2C+SN and I-A2C cases, respectively. Our experimental results demonstrate the outstanding advantages of applying DRL in traffic management, similar to other papers [15-19].

However, we used a modern model to optimise the network, so our results are better. Compared with the model (DQN) in [20-21], our model has the advantage of being able to simulate continuous actions. This is very important in traffic simulation because it is possible to describe the actions of agents accurately and fully. In contrast, the original DQN models have worse results because they can only perform discrete actions. Another finding is that the MA-A2C +SN model is more efficient at the initial stage, so it also converges more quickly.

Our model results express that as the CAV penetration rate rises, the model efficiency also improves. At a low CAV penetration rate, this improvement is not significant. The training process is based on data collected from CAVs. If this rate is too low, it is not enough for the learning process. The model is most effective when the CAV penetration rate is 100%, meaning that all vehicles in the model are CAVs.

Our models are tested under ideal conditions. The data are fully collected from CAVs and used as input in the model. No data loss or delay during data transmission. The traffic flow consists of two modes: CAVs and HDVs. To simplify the model, we ignored the influence of other modes (bus, bicycle) and pedestrians. Therefore, to apply the model in practice, additional adjustment steps are required.

For future work, we plan to deploy the model in a road network with real data to evaluate the model's performance. To increase the efficiency of our model, we also consider simulating pedestrians at intersections in a road network. In addition, we will conduct new scenarios in case of communication latency.

ACKNOWLEDGEMENTS

This work was supported by a Research Grant from Pukyong National University (2021).

REFERENCES

[1] Downs A. *Stuck in traffic Coping with peak-hour traffic congestion*. The Lincoln Institute of an Policy Cambridge, Massachusetts; 1992. DOI: 10.1177/0739456X9301200312.
[2] Bilbao-Ubillos J. The costs of urban congestion: Estimation of welfare losses arising from congestion on cross-town link roads. *Transportation Research Part A: Policy and Practice*. 2008;42(8):1098-1108. DOI: 10.1016/j.tra.2008.03.015.
[3] Chin YK, et al. Multiple intersections traffic signal timing optimization with genetic algorithm. *IEEE International Conference on Control System, Computing and Engineering, 2011, Penang, Malaysia*. 2011. DOI: 10.1109/ICCSCE.2011.6190569.
[4] Mondal MA, Rehena Z. Priority-based adaptive traffic signal control system for smart cities. *SN Computer Science*. 2022;3:417. DOI: 10.1007/s42979-022-01316-5.
[5] Lewis FF, Liu D. *Reinforcement learning and approximate dynamic programming for feedback control*. IEEE Press; 2012. DOI: 10.1002/9781118453988.
[6] Mannion P, Duggan J, Howley E. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In: McCluskey T, et al. (eds) *Autonomic road transport support systems.* Birkhäuser, Cham; 2016. p. 47-66. DOI: 10.1007/978-3-319-25808-9_4.
[7] Chu T, Wang J. Multi-agent deep reinforcement learning for large-scale traffic signal control. IEEE Transactions on Intelligent Transportation Systems. 2020;21(3). DOI: 10.1109/TITS.2019.2901791.
[8] Guo J, Cheng L, Wang S. CoTV: Cooperative control for traffic light signals and connected autonomous vehicles using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*. 2023;24(10): 10501-10512. DOI: 10.1109/TITS.2023.3276416.
[9] Miletic M, Ivanjko E, Greguric M, Kusic K. A review of reinforcement learning applications in adaptive traffic signal control. *IET Intelligent Transport Systems*. 2022;16:1269-1285. DOI: 10.1049/itr2.12208.
[10] Kiran BR, et al. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*. 2022;23(6):4909-4926. DOI: 10.1109/TITS.2021.3054625.
[11] Ge H, et al. Multi-agent transfer reinforcement learning with multi-view encoder for adaptive traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*. 2022;23(8):12572-12587. DOI: 10.1109/TITS.2021.3115240.
[12] Kuutti S, et al. End-to-end reinforcement learning for autonomous longitudinal control using advantage actor critic with temporal context. *IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand*. 2019. DOI: 10.1109/ITSC.2019.8917387.
[13] Google Maps, Google. https://www.google.com/maps. [Accessed 14 Mar. 2023].
[14] TomTom-Mapping and Location Technology, Tom Tom Technology. https://www.tomtom.com/. [Accessed 14 Mar. 2023].
[15] Branke J, Goldate P, Prothmann H. Actuated traffic signal optimisation using evolutionary algorithms. *Proceedings of the 6th European Congress and Exhibition on Intelligent Transport Systems and Services (ITS07), Jun 2007 Aalborg, Denmark*. 2007.
[16] Varaiya P. Max pressure control of a network of signalized intersections. *Transp. Res. Part C Emerg. Technol*. 2017;36:177-195.
[17] Ferreira M, et al. Self-organized traffic control. *Proceedings of the seventh ACM international workshop on vehicular internetworking, Chicago, IL, USA*. 2010. p. 85-90. DOI: 10.1145/1860058.1860077.
[18] Bretherton RD. Scoot urban traffic control system — Philosophy and evaluation. *IFAC Proceedings Volumes.* 1990. p. 237-239. DOI: 10.1016/S1474-6670(17)52676-2.
[19] Lowrie PR. SCATS: Sydney Co-Ordinated Adaptive Traffic System: A traffic responsive method of controlling urban traffic. Darlinghurst, NSW, Australia: Roads and traffic authority NSW; 1990.
[20] Greguric M, Vujic M, Alexopoulos C, Miletic M. Application of deep reinforcement learning in traffic signal control: An overview and impact of open traffic data. *Applied Sciences*. 2020;10(11). DOI: 10.3390/app10114011.
[21] Trinh TH, Bae SH, Duy QT. Deep reinforcement learning for vehicle platooning at a signalized intersection in mixed traffic with partial detection. *Applied Sciences*. 2022;12(19). DOI: 10.3390/app121910145.
[22] Liang X, Du X, Wang G, Han Z. A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology*. 2019;68:1243-1253. DOI: 10.1109/TVT.2018.2890726.
[23] Tran DQ, Bae SH. Proximal policy optimization through a deep reinforcement learning framework for multiple autonomous vehicles at a non-signalized intersection. *Applied Sciences*. 2020;10(16). DOI: 10.3390/app10165722.
[24] Schölkopf B, Platt J, Hofmann T. Advances in neural information processing systems 19: Proceedings of the 2006 Conference. *The annual Neural Information Processing Systems (NIPS) Conference, Vancouver*. 2006.
[25] Ma D, Zhou B, Song X, Dai H. A deep reinforcement learning approach to traffic signal control with temporal traffic pattern mining. *IEEE Transactions on Intelligent Transportation Systems*. 2022;23(8). DOI: 10.1109/TITS.2021.3107258.

[26] Sun QW, et al. Deep reinforcement-learning based adaptive traffic signal control with real-time queue lengths. *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Prague, Czech Republic*. 2022. DOI: 10.1109/SMC53654.2022.9945292.
[27] Wong A, et al. Deep multiagent reinforcement learning: Challenges and directions. *Artificial Intelligence Review*. 2022. DOI: 10.1007/s10462-022-10299-x.
[28] Wiering M. Multi-agent reinforcement learning for traffic light control. *Proceedings 17th ICML*. 2000.
[29] Prashanth LA, Bhatnagar S. Reinforcement learning with function approximation for traffic signal control. *IEEE Transactions on Intelligent Transportation Systems.* 2011;12(2). DOI: 10.1109/TITS.2010.2091408.
[30] El-Tantawy S, Abdulhai B, Abdelgawad H. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown Toronto. *IEEE Transactions on Intelligent Transportation Systems.* 2013;14(3). DOI: 10.1109/TITS.2013.2255286.
[31] Ge H, et al. Cooperative deep q-learning with Q-value transfer for multi-intersection signal control. *IEEE Access*. 2019;7:40797-40809. DOI: 10.1109/ACCESS.2019.2907618.
[32] Dijkstra EW. A note on two problems in connexion with graphs. *Numer. Math*. 1959;1(1). DOI: 10.1007/BF01386390.
[33] A start algorithm. https://en.wikipedia.org/wiki/A*_search_algorithm. [Accessed 14 Mar. 2023].
[34] Dorigo M, Maniezzo V, Colorni A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern*. 1996;26(1):29-41. DOI: 10.1109/3477.484436.
[35] Koh S, et al. Real-time deep reinforcement learning based vehicle navigation. *Applied Soft Computing*. 2020;96:106694. DOI: 10.1016/j.asoc.2020.106694.
[36] Claes R, Holvoet T, Weyns D. A decentralized approach for anticipatory vehicle routing using delegate multiagent systems. *IEEE Transactions on Intelligent Transportation Systems.* 2011;12(2):364-373. DOI: 10.1109/TITS.2011.2105867.
[37] Lopez PA, et al. Microscopic traffic simulation using SUMO. *IEEE, 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, Maui, USA*. 2018. DOI: 10.1109/ITSC.2018.8569938.

Hung Tuan Trinh, Sang-Hoon Bae, Duy Quang Tran

Cải thiện hiệu quả giao thông trong mạng lưới đường bộ bằng cách áp dụng Học tập tăng cường đa tác nhân phi tập trung và Điều hướng thông minh

Trong tương lai, luồng giao thông hỗn hợp sẽ bao gồm phương tiện do con người điều khiển (HDV) và phương tiện tự hành được kết nối (CAV). Quản lý giao thông hiệu quả là một thách thức toàn cầu, đặc biệt là ở các khu vực đô thị có nhiều nút giao thông. Nhiều nghiên cứu đã tập trung vào giải quyết vấn đề này để tăng hiệu suất của mạng lưới. Học tập tang cường (RL) là một cách tiếp cận mới để tối ưu hóa đèn tín hiệu giao thông, khắc phục những nhược điểm của phương pháp truyền thống. Trong bài báo này, chúng tôi đề xuất một phương pháp tiếp cận tích hợp kết hợp Multi-agent Advantage Actor-Critic (MA-A2C) và điều hướng thông minh (SN) để giải quyết vấn đề tắc nghẽn giao thông trong mạng lưới đường bộ trong điều kiện giao thông hỗn hợp. Thuật toán A2C kết hợp các ưu điểm của phương pháp dựa trên giá trị và dựa trên chính sách để ổn định quá trình đào tạo bằng cách giảm phương sai. Nó cũng khắc phục những hạn chế của phương thức MARL tập trung và độc lập. Ngoài ra, kỹ thuật SN định tuyến lại lưu lượng giao thông sang các đường thay thế để tránh tắc nghẽn tại các giao lộ. Để đánh giá mức độ mạnh mẽ của phương pháp của mình, chúng tôi so sánh mô hình của mình với thuật toán A2C độc lập (I-A2C) và Max Pressure (MP). Những kết quả này cho thấy phương pháp đề xuất của chúng tôi thực hiện hiệu quả hơn các phương pháp khác về thời gian chờ trung bình, tốc độ và độ dài hàng đợi. Ngoài ra, kết quả mô phỏng cũng cho thấy mô hình hiệu quả khi tỷ lệ thâm nhập CAV lớn hơn 20%.

Từ khóa
Học tăng cường đa tác nhân (MARL); Multi-agent advantage actor-critic (MA-A2C); Học tăng cường sâu (DRL); Mạng nơ-ron sâu (DNN); Xe tự lái và được kết nối (CAV) và Điều khiển tín hiệu giao thông.