Distributed Multi-Agent Reinforcement Learning for Collaborative Path Planning and Scheduling in Blockchain-Based Cognitive Internet of Vehicles

Huigang Chang, Yiming Liu, *Member, IEEE*, and Zhengguo Sheng, *Senior Member, IEEE*

***Abstract*—The collaborative path planning and scheduling can overcome the limitations of single vehicle intelligence to obtain a globally optimal decision strategy in cognitive internet of vehicles (CIoVs). The collaboration of vehicles necessitates the exchange of environmental and decision information, generating massive collaborative computing tasks with strict latency requirements. Leveraging mobile edge computing (MEC) technology, computing tasks can be processed near the vehicles to reduce latency. However, traffic congestion and computational load imbalance seriously affect traffic efficiency and computational latency. In hybrid driving scenarios, it is challenging to fulfill the diverse service requirements of vehicles with different intelligence levels. Moreover, non-collaborative tend to result in traffic congestion due to vehicle aggregation effects, while centralized solutions lack flexibility and have high computational complexity. To address these concerns, a distributed multi-agent reinforcement learning (DMARL) algorithm is proposed for collaborative path planning and scheduling in a blockchain-based collaboration framework. In this framework, we model the communication, traffic situation and task processing of the system and formulate a joint optimization problem to minimize both travel time and computation latency. Last, we convert the scheduling problem for different types of vehicles into Markov decision processes (MDPs) and propose Q-learning-based DMARL algorithm to achieve proactive load balancing of both road infrastructures and MEC nodes (MECNs). Simulation results demonstrate that the proposed approach outperforms the comparison schemes in terms of load balance indexes of roads and MECNs, travel time, and computation latency.**

***Index Terms*—Cognitive Internet of vehicles, mobile edge computing, path planning and scheduling, multi-agent reinforcement learning, load balancing.**

I. Introduction

THE cognitive Internet of vehicles (CIoVs) introduce cognitive engines to perceive the traffic environment and network status which can assist the path planning and scheduling of vehicles [1]. The collaborative path planning and scheduling of different types of intelligent vehicles requires analysis of a large amount of onboard and environment perception data for decision making, which will generate a large number of computing tasks and consume numerous computing resources. Meanwhile, the intelligent vehicles with computing power need to efficiently handle complex autonomous driving tasks to improve traffic efficiency and reduce traffic congestion. However, vehicles with constrained storage and computing resources are not sufficient to handle these explosive computing tasks. Due to the limited capacity and high energy consumption of vehicles, a large number of computing tasks need to be offloaded to mobile edge computing nodes (MECNs) deployed at the roadside for processing [2].

In hybrid driving scenarios, the vehicles with different levels of intelligence, such as connected automated vehicles (CAVs) and connected ordinary vehicles (COVs) have different requirements and priorities for computing and traffic services [3]. Furthermore, the traffic efficiency and computing task processing latency are closely affiliated with the traffic situation and the computing load on the MECNs. However, the imbalanced distribution of vehicles and computation loads often leads to road congestion or overload of MECNs in CIoVs, which will seriously affect the traffic efficiency and computational latency of vehicles [4], [5]. In addition, the unbalance distribution and mismatch of resources and demands will also cause underutilization of resources, thus affecting system performance. To improve the performance of intelligent transportation systems (ITSs) powered by CIoVs, the intelligent path planning and scheduling of vehicles are essential to reduce the travel time and task processing latency for various types of vehicles [6], [7].

On the other hand, due to the selfish behavior of individuals, the independent decision making approach may cause secondary congestion by navigating excessive vehicles to non-congested road sections. It will lead to an aggregation effect of vehicles thus causing the traffic congestion and computational performance degradation. The decision of vehicles affects the environment state, which in turn influences the decision of other vehicles. Therefore, the non-collaborative approach cannot obtain a globally optimal scheduling policy. The collaborative decision making among vehicles provides a solution for obtaining global optimal path planning and scheduling

Manuscript received March 26, 2023; revised November 14, 2023; accepted December 9, 2023. This work was supported by the National Natural Science Foundation for Young Scientists of China under Grant 62001050, the Science and Technology Development Fund (SKL-IOTSC(UM)-2021-2023) and the State Key Laboratory of Internet of Things for Smart City (University of Macau) (Ref. No.: SKL-IoTSC(UM)-2021-2023/ORPF/SA02/2022), the Fundamental Research Funds for the Central Universities (Grant No. 2023RC95), and European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Grant Agreement, United Kingdom, (101006411). *(Huigang Chang, Yiming Liu contributed equally to this work and should be considered co-first authors. Corresponding author: Yiming Liu.)*

H. Chang and Y. Liu are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China. H. Chang is also with China Intelligent and Connected Vehicles (Beijing) Research Institute Co.,Ltd, Beijing 100176, China (e-mail: changhuigang@bupt.edu.cn; liuyiming@bupt.edu.cn).

Z. Sheng is with the Department of Engineering and Design, University of Sussex, Brighton, BN1 9RH, United Kingdom (e-mail: z.sheng@sussex.ac.uk).

strategy [8]. However, centralized solutions increase computational complexity and mismatch with the flexibility needs and temporal relevance of decisions. In addition, effective vehicle collaboration requires ensuring trustworthiness of shared information. Blockchain with the characteristics of decentralization, immutability, auditability, and anonymity, ensuring that the information on the chain cannot not be tampered with, thus providing a credible information base for vehicle collaboration and traffic accident determination [9], [10]. Leveraging blockchain technology to perform secure information sharing can facilitate distributed collaborative decision making for ITSs [11], [12]. However, blockchain with high computation and communication overhead is difficult to apply directly to CIoVs considering the cost and efficiency issues. Furthermore, current research efforts are mainly focused on reducing travel time and lacks joint optimization with the computation latency in CIoVs, ignoring the diverse service requirements of different types of vehicles in the hybrid driving scenarios.

To address these issues, we consider a hybrid driving scenarios and propose a distributed multi-agent reinforcement learning (DMARL) algorithm for collaborative path planning and scheduling in blockchain-based CIoVs. We utilize blockchain to build a collaborative framework based on our prior work [13] to facilitate distributed collaborative decision making in CIoVs. In the framework, we build a communication model for information transmission and analyze the factors influencing the consensus latency of vehicle collaboration. In hybrid driving scenarios, we analyze traffic efficiency and computational performance of different types of vehicles to satisfy diverse service requirements for ITSs. Then, we formulate a joint optimization problem to minimize both travel time and computation latency. To solve the problem, we propose a Q-learning-based DMARL algorithm to find the globally optimal path and scheduling strategy for proactive load balancing of road infrastructure and MECNs. The main contributions of this paper are as follows:

- We propose a blockchain-based collaboration framework to support global collaborative decision optimization of vehicles. The collaborative decisions are recorded as transactions in the block, which facilitates the credible collaboration and helps in traffic incident tracing and investigation.
- We consider a realistic hybrid driving scenario and analyze the collaborative consensus latency of the distributed decision process. For the traffic and network environments, we model the communication, traffic situations and computational task processing to satisfy the diverse service requirements of different types of vehicles.
- We formulate a joint optimization problem based on the established models to minimize travel time and computation latency. Furthermore, we devise the scheduling problem as Markov decision processes (MDPs) and propose a Q-learning-based DMARL algorithm for collaborative path planning and scheduling to achieve proactive load balancing of both road infrastructure and MECNs.
- Extensive experiments are conducted and discussed to evaluate the effectiveness of the proposed algorithm compared with other benchmark schemes regarding load balancing indexes, travel time, and computation latency.

The rest of our work is organized as follows. Section II introduces related works. Section III presents the collaboration framework for collaborative path planning and scheduling. Section IV discusses the system model and analyzes the collaborative consensus latency for problem formulation. In Section V, the proposed solution of Q-learning-based DMARL algorithm for collaborative path planning and scheduling is presented in detail. Section VI discusses the simulation results. Finally, we conclude this paper in Section VII.

II. Related Works

In this section, we investigate the related works on path planning and load balancing, blockchain for information sharing, and multi-agent reinforcement learning in vehicular networks, respectively.

A. Path Planning and Load Balancing in Vehicular Networks

To avoid road congestion, Sun *et al*. [6] proposed a path planning algorithm to obtain the shortest travel time and maintain global load balancing based on the prediction of average travel velocity. Lin *et al*. [14] proposed a social vehicle route selection algorithm to find the optimal route for vehicles to reduce traffic congestion and manage traffic flow. Li *et al*. [4] proposed a proactive load balancing approach to enable efficient cooperation among mobile edge servers based on the predicted traffic situation in vehicular networks. Pan *et al*. [15] proposed a distributed rerouting system for congestion avoidance in which a large portion of the rerouting computations are offloaded on vehicles to accelerate the rerouting process. Xie *et al*. [16] investigated a joint optimization problem of vehicle data offloading and route selection to improve the throughput and reduce the travel time of vehicles. Dai *et al*. [17] proposed an offloading scheme considering the load balancing and resource allocation of the vehicular edge computing system to maximize the system utility. Li *et al*. [18] proposed an online reinforcement learning based load balancing method in vehicular networks by observing the characteristic of urban traffic flow. Most of the works investigated path planning and load balancing methods to avoid road congestion. However, they mainly considered the performance optimization of vehicle traffic efficiency, which cannot meet the service requirements of different vehicles in hybrid driving scenarios. In addition, due to the lack of information interaction and analysis of the impact of vehicle decisions on the environment state, individual decision cannot effectively address the aggregation effect of vehicles.

B. Blockchain for Information Sharing in Vehicular Networks

Blockchain provides an effective solution to facilitate the sharing of information in the distributed vehicular networks [3], [19]. A lot of works have focused on knowledge and information sharing based on blockchain in vehicular networks [20]–[22]. Blockchain enables intelligent vehicles to collaborative inferences and makes decisions by sharing information

and knowledge in a secure and privacy-preserving manner. Fu *et al*. [12] utilized vehicular blockchain to support the knowledge transfer of deep reinforcement learning (DRL) for autonomous lane-changing systems, which improved the driving security of vehicles. Song *et al*. [3] proposed a blockchain-based positioning error sharing model to improve the positioning accuracy of common vehicles through a deep neural network (DNN) algorithm. Chai *et al*. [20] proposed a hierarchical blockchain framework for knowledge sharing in distributed vehicular networks, which facilitated vehicles learning environmental data through machine learning methods and sharing the learning knowledge. He *et al*. [21] employed blockchain to build a decentralized machine learning system that enables CAVs to obtain a better global model. Jiang *et al*. [22] proposed a blockchain-enabled distributed deep learning model sharing approach assisted by MEC to improve the object detection performance of autonomous driving systems. Blockchain improves the trusted sharing of information in CIoVs. However, these works did not consider the collaborative path planning and scheduling in vehicular networks, which are still in the preliminary stage.

C. Multi-Agent Reinforcement Learning in Vehicular Networks

Multi-agent reinforcement learning based algorithms facilitate intelligent vehicles to make the correct action strategies to improve the performance of ITSs in the complex traffic environment. Lin *et al*. [23] designed a deep Q-learning (DQN) based distributed multi-agent reinforcement learning model to solve the online routing decision problem of vehicles in the software-defined IoVs. Guillen *et al*. [24] proposed a multi-agent deep reinforcement learning (MADRL) method that enables the collaborative movement of CAVs at intersections to enhance the capacity of autonomous intersection management systems. Ren *et al*. [25] proposed a multi-agent reinforcement learning model based on the encoder-decoder framework for vehicle path planning, which optimized the route length and travel time simultaneously. Qin *et al*. [26] proposed a multi-agent reinforcement learning approach based on an actor-critic algorithm to perform dynamic transportation task assignments for vehicles in the urban transportation system. Kwon *et al*. [27] proposed a multi-agent deep deterministic policy gradient (MADDPG) approach to improve the throughput of connected vehicles in vehicular networks. Although the above works have achieved satisfactory results in improving the performance of advanced applications in vehicular networks, they did not consider the interaction among intelligent vehicles with autonomous decision making capabilities, which cannot satisfy the low latency and flexibility requirements of large-scale collaboration scenarios in CIoVs.

III. Distributed Collaboration Framework for Collaborative Path Planning and Scheduling

This section introduces the distributed collaboration framework for collaborative path planning and scheduling in blockchain-based CIoVs, including the system architecture and collaboration and consensus processes.

TABLE I
Main Notations

Notation	Definition
$\mathcal{N}, \mathcal{M}, \mathcal{V}$	Set of vehicles, MECNs, and collaborative vehicles
C_n^{pro}	Computation resources provided by vehicle n
C_b^{req}	Computation resources required for block generation
C_m^{ava}	Available computing resources of MECN m
N_g	Number of vehicles on road section g
Γ_{nm}	Communication rate of vehicle n access to MECN m
$h_{n,m}$	Channel gain between vehicle n and MECN m
B_m	Total bandwidth
a_n	Node selection factor
D_b	Data volume of the generated block
$\mathfrak{D}_g$	Vehicle density of road section g
v_g	Maximum velocity limit of road section g
F_g	Vehicle traffic flow of road section g
a_n^j	selection factor of CAV n for task j
b_{nm}	Selection factor of MECN m for CAV n
$C_{\min}^j$	Minimum computation resources allocated to task j
N_a	Number of collaborative vehicles

A. System Architecture

Fig. 1 shows the system architecture for collaborative path planning and scheduling in blockchain-based CIoVs. Collaborative path planning and scheduling not only selects the optimal path for different types of vehicles in hybrid driving scenarios, but also optimizes road congestion and MECN load through a reasonable scheduling strategy to achieve proactive load balancing of both road and MECNs. The vehicle set is denoted by $\mathcal{N} = \{1, 2, \ldots, n, \ldots, N\}$, and the MECNs set is represented as $\mathcal{M} = \{1, 2, \ldots, m, \ldots, M\}$. In hybrid driving scenarios, the CAVs and COVs act as mobile terminals with different communication, computing, and control capabilities, have different driving and service requirements. Among them, CAVs have a large number of computing tasks with low latency requirements to support secure autonomous driving systems [28]. Meanwhile, COVs with human drivers are more focused on improving the driving experience and reducing travel time. The CAVs and COVs set are denoted as $\mathcal{V} = \left\{\left(V_1^{cav}, V_i^{cav}, \cdots, V_q^{cav}\right), \left(V_1^{cov}, V_j^{cov}, \cdots, V_{N-q}^{cov}\right)\right\}$. The micro base stations or roadside units (RSUs) as MECNs are deployed at each road section g in urban areas and equipped with edge computing servers to provide latency-sensitive computing services [29], [30]. In addition, vehicles can utilize distributed edge computing resources for model training through edge learning, avoiding the dependence and limitation on centralized servers, and can efficiently support large-scale collaborative decision making among vehicles [31]. The CAVs with limited computing resources offload latency-sensitive and computation-intensive tasks to the MECNs. Meanwhile, the macro base station connects to the remote cloud server to provide latency-insensitive services for vehicles. Each MECN serves one road section area and multiple vehicles. Vehicles can be divided into different clusters as batch collaborative units and each vehicle is served by one MECN in a timeslot within the coverage of MECNs. The main notations in this paper are illustrated in Table I.

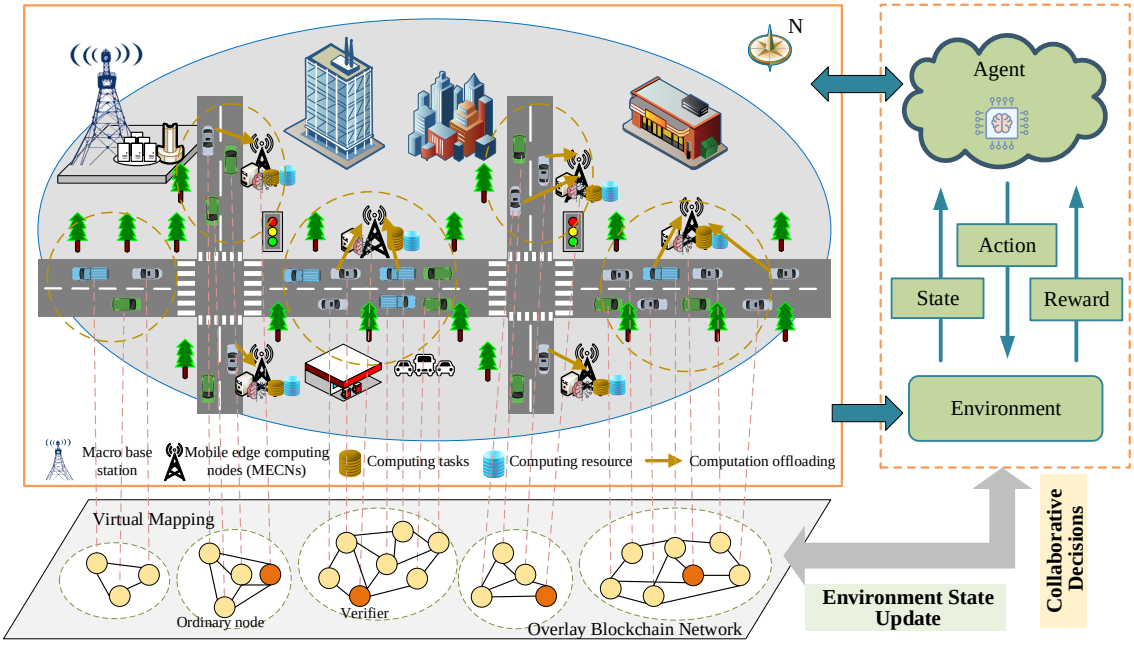


Fig. 1. System architecture of collaborative path planning and scheduling in blockchain-based CIoVs.

B. Security of Blockchain-Based Collaboration Processes

The proposed blockchain-based collaboration approach utilizes authorization and credibility evaluation mechanisms to ensure the security of the system. First, the Certificate Authority (CA) authorizes the collaborative vehicle nodes and distributes keys. The CA-authorized vehicle n receives an asymmetric public-private key $\{K_n^{pu}, K_n^{pr}\}$ and an anonymous identity for identity verification and authorization management. The K_n^{pu} and K_n^{pr} are the public key and private key of collaborative vehicle n respectively. To ensure the security and reliability of vehicle collaboration information, the sender performs a hash operation on the decision message and uses K_n^{pr} encryption to generate a digital signature. The receiver exploits the sender's public key K_n^{pu} to decrypt the digital signature and verify whether the sender's identity is legitimate. This process compares the hash value of the shared decision data through a hash operation to ensure the integrity of the received message and verify whether it has been tampered with, destroyed or forged. In addition, to ensure security and resist malicious attacks in the network environment, the proposed blockchain system evaluates node credibility through our proposed credit-based delegated byzantine fault tolerance (CDBFT) algorithm, thus eliminating security issues caused by malicious node interference [13]. The credit of consensus node MECNs changes based on the normal and abnormal behavior of the consensus process. Abnormal or malicious attacks by consensus nodes will be punished and their credit value will be reduced. Therefore, the credibility of consensus node i can be obtained is $R_i = \eta_1 R_i^O - \eta_2 R_i^U$. Among them, η_1 and η_2 are weight coefficients, R_i^O and R_i^U respectively represent normal and abnormal behaviors that affect the credibility of consensus node i. By adjusting the coefficient η_2, a more stringent penalty strategy can be obtained [13]. Unlike other existing consensus protocols of DBFT [32] and practical byzantine fault tolerance (PBFT) [33], during our proposed consensus process, candidate consensus nodes with higher credibility ranking priority are selected to participate in the consensus process to resist malicious attacks. Assuming that the number of the selected consensus node set is N_c, the maximum number of fault-tolerant nodes is $f = (N_c - 1)/3$. When the voting confirmation message is not less than $N_c - f$, the proposed new block completes the consensus, thereby ensuring the security and efficiency of the blockchain system [13], [32].

1) Distributed Decision Process: As shown in Fig. 2, we design a collaboration framework based on our previous work [13] in blockchain-based CIoVs to support an auditable, traceable, and trusted collaborative path planning and scheduling. The consortium blockchain is exploited and the consensus process utilizes our proposed CDBFT algorithm in [13] taking into account the energy consumption and time efficiency. We introduce credibility evaluation and penalty mechanisms to stimulate participants to behave honestly and make them comply with the consensus rules. In addition, the proposed distributed collaborative decision making and the consensus process of the blocks are carried out simultaneously, which ensures the efficiency and data security of collaborative path planning of vehicles. Furthermore, asymmetric encryption technology and hash function are utilized to verify the authenticity of decisions and ensure the system security [12], [33]. During the collaboration process, the MECNs on the roadside update the environment state and perform collaborative decision consensus simultaneously. The processes of collaborative path planning and scheduling are as follows:

- Firstly, the vehicles perform path planning using Q-learning algorithm based on the current state of the environment and share the signed decisions with the

nearby cognitive engines deployed at MECNs.
- Then, if the decision information passes validation, the cognitive engines update the environment states concerning traffic situations and computation loads based on the decisions. The other collaborative vehicles dynamically adjust the reward function according to the updated environment state.
- Finally, all collaborative vehicles obtain the global optimal decision and MECNs package them into new blocks for consensus to achieve consistency. The distributed decision making and consensus process on blocks are are executed in parallel.

2) Consensus Process: The consensus process of the proposed CDBFT consensus mechanism is shown in Fig. 3.

a) Pre-Prepare: The leader node packages the validated decision transactions into a new block proposal and broadcasts the unconsensus new block to other consensus nodes for verification and conformation.

b) Prepare: the selected consensus node decrypts the new block signature via public key to verify the integrity and the legitimacy of the block. If the verification passes, the selected consensus node votes for the new block signature and sends a verification message to other consensus nodes.

c) Commit: All consensus nodes participating in the verification send a vote confirmation message $conf$ to the leader node. Once it confirms that no less than $N_c - f$ voting information is satisfied, it sends a confirmation completion message to the consensus node.

d) Reply: After receiving the confirmation completion message, the consensus nodes reply to the leader node to complete the confirmation and enter the block generation stage.

e) Block Generation: Once the consensus node receives no less than $N_c - f$ confirmation completion message, it replies to the leader node to complete the confirmation and enter the phase to reach consensus, and the new block will be permanently recorded on the blockchain [13], [32].

Specifically, take vehicle n as an example, vehicle n uses private key K_n^{pr} to sign the decision digest to guarantee the authenticity and legitimacy of the uploaded decisions. Then, the receiving cognitive engine verifies the identity of the sender with the public key K_n^{pu} and validate the hash to ensure the integrity of the received message which cannot be tampered with. For each consensus epoch, the authorized MECNs form a group of verifiers based on their credibility.

The verifier group selects the packer based on the credibility and resource status of the candidates, which is selected as leader node MECN for block production. In the current epoch, the leader node packs the shared decisions into a new block with a common structure as shown in Fig. 2 [13]. The leader node signs the new block with the private key K_m^{pr} and broadcast to other verifiers for consensus. If more than two-thirds of the verifiers endorse the block, a consensus is reached and it is added to the end of the chain. Once consensus is reached, the collaborative decisions are permanently and securely stored on the blockchain, which facilitates the traceability of traffic events [34]. During the consensus process, the vehicle does not acquire the decision information but adjusts its strategy according to the expected environment state updates of the cognitive engine. The participants reach consensus on collaborative path planning and scheduling strategies, resulting in credible and traceable online proof of collaborative decision results, which is beneficial for improving traffic management and accident determination in CIoVs. If a traffic accident or incident requires liability determination and investigation, the service requester will pay the appropriate tokens to access the information on the blockchain.

IV. System Model and Problem Formulation

In this section, we construct the communication model, traffic situation and computation load model, and task processing model in CIoVs. Then, the factors affecting the collaborative consensus latency are analyzed, and a joint optimization problem for proactive load balancing of both road infrastructures and MECNs is formulated.

A. Communication Model

The environment state needs to be dynamically updated based on the vehicle's decisions to support vehicle collaboration, which requires the transmission of decision information. Therefore, the proposed collaboration framework utilizes the PC5 and the Uu interface of 5G V2X [35] for vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-network (V2N) communication. We consider the wireless channel gain between vehicle n and RSU m is $h_{n,m}$ which follows an exponential distribution due to Rayleigh fading, and the transmission power from the vehicle to RSU is $p_{n,m}$ [36], [37]. We assume that the total bandwidth is $B_m(t)$, $N_{n,m}(t)$ denotes the number of vehicles served by RSU m, and $B_m(t)$ is dynamically assigned to the serving vehicles in a time slot t [38]. In addition, to avoid inter-cell interference, neighboring RSUs use different frequency bands. The communication rate of the n-th vehicle served by m-th RSU can be written as

$$\Gamma_{nm}(t) = \frac{B_m(t)}{N_{n,m}(t)} \log_2\left(1 + \frac{p_{n,m}|h_{n,m}|^2(t)}{\delta^2 + \sum_{k\in\mathcal{B},k\neq m} p_{n,k}|h_{n,k}|^2(t)}\right), \tag{1}$$

where δ^2 denotes the noise power of the additive Gaussian white noise (AGWN) with zero mean and variance δ^2 [39]. The fraction in parentheses represents the signal-to-interference-plus-noise ratio (SINR) of n-th vehicle served by m-th RSU. The $\sum_{k\in\mathcal{B},k\neq m} p_{n,k}\,|h_{n,k}|^2$ represents the interference from other micro base stations $\mathcal{B}$ in vehicular networks [18]. Also considering the traffic service requirements of the vehicle, we performed the traffic situation analysis below.

B. Traffic Situation and Computation Load Model

1) Mobility Model: We build a traffic situation model based on the perception of the traffic environment. Assume the number of vehicles on road section g at timeslot t is $N_g(t)$, the vehicle density of road section g can be derived as

$$\mathfrak{D}_g(t) = \frac{\sum_{n=1}^{N_g(t)} l_n}{L_g u_g}, \tag{2}$$

where l_n are the length of vehicle n, L_g is the length of the road section g, and u_g is the number of lanes. The estimated

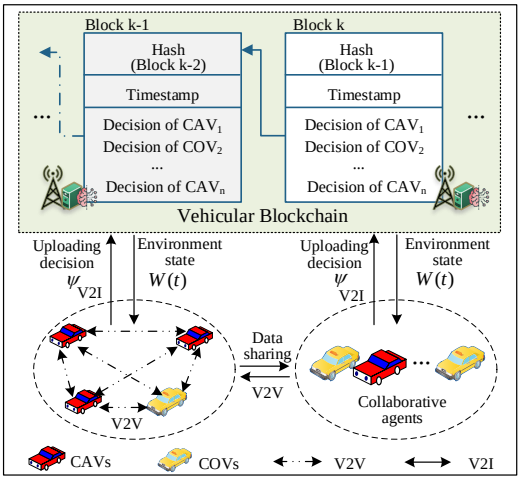


Fig. 2. Blockchain-based distributed collaborative decision process.

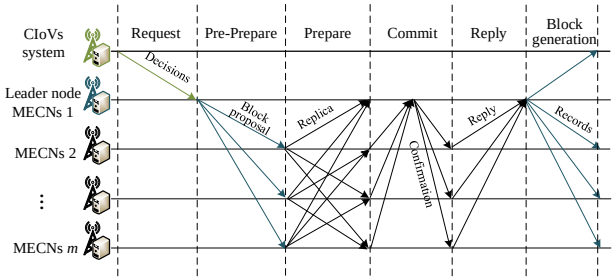


Fig. 3. The consensus process.

velocity V_g is derived from the vehicle density of the road and can be written as [15]

$$V_g(t) = v_g\left(1 - \frac{\mathfrak{D}_g(t)}{\mathfrak{D}_{jam}(t)}\right), \tag{3}$$

where v_g is the maximum velocity limit of road section g, $\mathfrak{D}_{jam}$ is the vehicle density when the road g is congested. Since the road with a fixed total length, we can derive $\frac{\mathfrak{D}_g(t)}{\mathfrak{D}_{jam}(t)} = \frac{\sum_{n=1}^{N_g(t)} l_n}{\sum_{n=1}^{N_{jam}} l_n} = \frac{N_g(t)}{N_{jam}}$, N_{jam} is the maximum number of vehicles when road is congested. We derive the travel time of vehicle n on the road section g as

$$T_{g,n} = \frac{L_g}{V_g} = \frac{L_g}{v_g\left(1 - \frac{N_g(t)}{N_{jam}}\right)} = \frac{L_g \times N_{jam}}{v_g\left(N_{jam} - N_g\left(t\right)\right)}. \tag{4}$$

2) Traffic Flow Model: We build a traffic flow model to adapt to dynamic traffic situations. The volume of inflow and outflow directly influence the congestion variation on the road, which can be crowd sensed by induced loops, vehicular sensor data, and cameras [4]. The traffic inflow and outflow on road section g at timeslot t are $f_{in,g}\left(t\right)$ and $f_{out,g}\left(t\right)$, respectively. We derive the traffic volume on road section g at timeslot t as

$$N_g(t) = N_g\left(t-1\right) + f_{in,g}\left(t\right) - f_{out,g}\left(t\right), \; N_g(t) \geq 0. \tag{5}$$

If $f_{in,g}(t) < f_{out,g}(t)$, the number of vehicles on the road is decreasing, which can reduce the level of road congestion and load on MECNs. It keeps decreasing or even $N_g(t) = 0$ will cause the underutilization of computing resources and road infrastructures, especially during rush hour. Conversely, if $f_{in,g}(t) > f_{out,g}(t)$, it increases the load on the MECNs and road infrastructures. We define the change in traffic flow as $F_g(t) = f_{in,g}(t) - f_{out,g}(t)$, where $F_g(t) > 0$ denotes that the inflow is greater than the outflow, $|F_g(t)|$ indicates the number of vehicles added to the road section g and vice versa. For better computing performance and driving experience, the ational scheduling of COVs and CAVs should be performed to fully utilize the computing resources of MECNs without causing congestion.

3) Computation Load Distribution: The computation load of MECNs is tightly related to the vehicle density and the proportion of CAVs which have various computing tasks such as image recognition, object detection, decision making, etc. Based on the perception of the traffic situation, we derive the computation load distribution of the MECN m,

$$Load_m(t) = \sum\nolimits_{n=1}^{\chi(t)N_g(t)} J_n(t), \tag{6}$$

where $J_n(t)$ is the amount of computing tasks for CAV n, and $\chi(t)$ is the proportion of CAVs. There will be a large number of computing tasks offloaded to connected MECNs as the vehicle density and the proportion of CAVs increase. Since a large number of computational tasks of CAV require the assistance of MECNs, the edge computing task processing model is analyzed below.

C. Computing Task Processing Model

The latency of computing tasks offloaded to MECNs for processing includes transmission latency, queuing latency, and computing latency. It depends on the number of tasks, bandwidth, and available computation resources of MECNs, i.e., the central processing unit (CPU) cycles per second [31], [40]. Assume CAV n has J different computing tasks, each task has a data volume of $D_{n,m}^j$, the computation resources required to complete task j is $C_{n,j}^{req}$. The computing tasks j need to be completed within the specified maximum latency $T_n^{j,max}$. The transmission latency of CAV n for offloading the computing tasks to MECN m can be written as

$$T_{nm}^{tj} = \sum\nolimits_{j=1}^{J} a_n^j b_{nm} \frac{D_{n,m}^j}{\rho \Gamma_{nm}}, \tag{7}$$

where ρ denotes communication disturbance factor, a_n^j denotes the selection factor of CAV n offloading the computing task j to MECNs, $a_n^j = 1$ represents CAV n offloads the task j, otherwise $a_n^j = 0$. Similarly, $b_{nm} = 1$ represents CAV n offloading computing tasks to MECN m, otherwise $b_{nm} = 0$.

Task j has to queue if the available computation resources of MECN m are less than $C_{\min}^j$ that the minimum computation resource for task j. The MECNs take the First Input First Output (FIFO) queue to process the arriving tasks. We assume that the MECN m has a computing capacity of C_{cap} and Φ computing tasks before task j. The queuing options ϖ_{nj} for task j can be derived as

$$\varpi_{nj} = \begin{cases} 1 & \left(C_{cap} - \sum_{i=1}^{\Phi} C_{nm}^i\right) < C_{\min}^j \\ 0 & \left(C_{cap} - \sum_{i=1}^{\Phi} C_{nm}^i\right) \geq C_{\min}^j \end{cases}, \tag{8}$$

where C_{nm}^{i} is the computing resources occupied by previous task i, $C_{occ} = \sum_{i=1}^{\Phi} C_{nm}^{i}$ represents occupied computing resources of MECN m. The queuing latency T_{nm}^{wj} for the task j from CAV n offloaded to MECN m can be expressed as

$$T_{nm}^{\varpi j} = \varpi_{nj} \sum_{i=1}^{\Phi} \frac{C_n^i}{C_{nm}^i}, \forall i \in \Phi, C_{nm}^i \geq C_{\min}^j, \quad (9)$$

where C_n^i is the computation resources required for previous task i. Then, we can derive the computation latency of CAV n as

$$\begin{aligned} T_{nm}^{cj} &= \sum_{j=1}^{J} a_n^j b_{nm} \frac{C_{n,j}^{req}}{C_{nm}^j}, \; C_{nm}^j \geq C_{\min}^j \\ C_{nm}^j &= \frac{C_{cap} - C_{occ}}{\sum_{n=1}^{q-N_\Phi} J_n}, \end{aligned} \quad (10)$$

where C_{nm}^{j} is the computation resource allocated to task j, $C_{n,j}^{req}$ is the computation resource required to complete task j from CAV n, and N_Φ denotes the number of service vehicles corresponding to the previous Φ tasks. The total latency of CAV n offloading the computing tasks to MECN m process is achieved as

$$\begin{aligned} T_{nm} &= T_{nm}^{tj} + T_{nm}^{\varpi j} + T_{nm}^{cj} \\ &= \sum_{j=1}^{J} a_n^j b_{nm} \frac{D_{n,m}^j}{\rho \Gamma_{nm}} + \varpi_{nj} \sum_{i=1}^{\Phi} \frac{C_n^i}{C_{nm}^i} \\ &+ \sum_{j=1}^{J} a_n^j b_{nm} \frac{C_{n,j}^{req}}{C_{nm}^j}. \end{aligned} \quad (11)$$

The total computation latency of tasks offloaded by all CAVs is an indicator to evaluate the computing performance of CIoVs, which can be derived as

$$T_m = \sum_{n=1}^{q} \sum_{m=1}^{M} T_{nm}, \quad (12)$$

where q is the total number of CAVs and M is the total number of MECNs.

D. Collaborative Consensus Latency Analysis

The Collaborative consensus latency of blockchain-based CIoVs system includes decision completion latency T_c, decision transmission latency T_t, and block consensus latency T_v which are associated with the number of collaborative vehicles, communication, and computation resources. Assume that the node selection factor is a_n, i.e., participating in the collaboration, $a_n = 1$, otherwise $a_n = 0$. The decision completion latency for path planning and scheduling can be expressed as

$$T_c = \sum_{n=1}^{N} \frac{a_n C_n^r(\xi, \mathbf{\Omega})}{C_n^{pro}}, \quad (13)$$

where C_n^r represents the computation resources required by the path planning and scheduling algorithm, which is associated with the algorithm complexity ξ and the state and action space dimensions $\mathbf{\Omega}$ owned by the vehicle n, C_n^{pro} is the computation resources provided by collaborative vehicle n. The state and action space dimensions $\mathbf{\Omega}$ will affect the complexity of the decision algorithm. The data volume of decision information for each vehicle is D_n, we can derive the transmission latency as follows,

$$T_t = \sum_{n=1}^{N} a_n \frac{D_n}{\rho \Gamma_{nm}}. \quad (14)$$

As shown in Fig. 3, in the consensus process, the block consensus latency for collaborative decisions mainly includes the block production latency, verification latency, and the latency of verification results broadcasting and judgment among verifiers. The leader node MECN 1 gets the privilege to pack new blocks for proposal in a consensus epoch, the block consensus latency can be written as

$$T_v(a_n, D_b, C_v^{ava}) = \frac{D_b(a_n) C_b^{req}}{C_m^{ava}} + \frac{C_v^{req}}{C_v^{ava}} + \vartheta D_b \left| \mathbb{M} \right|, \quad (15)$$

where C_b^{req} and C_m^{ava} represent the volume of computing resources required for block production and available computing resources of packager, respectively, and D_b is the packaged block size. The amount of computation required for verification is C_v^{req}, and the available computation resources of verifier v is C_v^{ava}. Similar to that in [37], the latency of verified block broadcast and verification results judgment among verifiers is a function of the block size D_b, the number of verifiers $|\mathbb{M}|$, and average verification speed of each verifier, which denoted as $\vartheta D_b |\mathbb{M}|$. Specifically, ϑ is a predefined parameter for the process of verification result broadcast and judgment, which is related to the behavior of consensus nodes. Attacks and interference from malicious consensus nodes will seriously slow down the consensus process and are related to the abnormal attack behavior of consensus nodes. Two typical attack models are considered here, namely Blackhole Miners attacks and Colluding Attacks. Blackhole miners are when selected consensus nodes refuse to broadcast blocks or perform verification during the consensus process. Collusion attacks refer to malicious consensus nodes colluding together to perform biased and erroneous consensus guidance and add malicious blocks on the blockchain [41]. From the above analysis, the collaborative consensus latency of blockchain-based CIoVs systems can be written as

$$T_{con} = T_c + T_t + T_v. \quad (16)$$

It can be seen that T_{con} is mainly associated with the node selection factor a_n that participates in collaboration, decision making algorithm complexity, and the available computation resources $C_r = \{C_n^{pro}, C_m^{ava}, C_v^{ava}\}$. Therefore, the proposed solution will focus on optimizing the above parameters, i.e., reducing the complexity of the algorithm and perform load balancing of the MECNs. To meet the low latency demand of CIoVs, the proposed consensus mechanism prioritize nodes with abundant computing resources and high credit to complete block generation and consensus [13]. Moreover, we reduce the state and action space dimensions $\mathbf{\Omega}$ of MARL that affect the decision latency T_c by distributed collaboration and control the node selection factor a_n that involved in collaboration considering a batch iterative update strategy. Next, based on the developed model and analysis, we performed the problem formulation.

E. Problem Formulation

The travel time and computing latency are associated with the density of vehicles and the volume of computing tasks offloaded on MECNs which can be optimized by proactive load balancing of MECNs and road infrastructures. The

large number of congested and idle road sections and the MECNs indicate a high index of load disparity which will degrade the system performance. Therefore, a reasonable path planning and scheduling strategy is needed to complete the load balancing of road and MECNs. We divide the map area into different road sections $\mathcal{G} = \{1, 2, \ldots, g, \ldots, G\}$, the vehicle density set of road sections and computing load set of MECNs are $\mathcal{D} = \{\mathfrak{D}_1, \mathfrak{D}_2, \ldots, \mathfrak{D}_g, \ldots, \mathfrak{D}_G\}$ and $\mathcal{L} = \{Load_1, Load_2, \ldots, Load_m, \ldots, Load_M\}$, respectively. To evaluate the system performance, we define the road load balance index $\mathbb{B}$ and MECNs load balance index $\mathbb{L}$ [42], which can be expressed as

$$\mathbb{B}(t) = \sqrt{\sum_{g=1}^{G} \frac{\left|\mathfrak{D}_g(t) - \bar{\mathfrak{D}}(t)\right|^2}{G}}, \tag{17}$$

$$\mathbb{L}(t) = \sqrt{\sum_{m=1}^{M} \frac{\left|Load_m(t) - \overline{Load}(t)\right|^2}{M}}, \tag{18}$$

where $\mathfrak{D}_g$ and $\bar{\mathfrak{D}}$ are the vehicle density of the road section g and the average vehicle density of all road sections; $Load_m$ and $\overline{Load}$ are the load of the MECN m and the average load of all MECNs, respectively. The uniformity of load distribution is inversely proportional to the index value $\mathbb{B}$ and $\mathbb{L}$, $\mathbb{B} \geq 0$, $\mathbb{L} \geq 0$, the smaller the value, the better the load balancing performance. Travel time and computational latency are closely related to vehicle density and computational load of MECNs. Active load balancing of MECNs and road infrastructure enables efficient and rational utilization of physical and cyberspace multidimensional resources. The collaborative path planning and scheduling guides vehicles to lighter load and non-congested road sections and minimize $\{\mathbb{B}(t), \mathbb{L}(t)\}$.

Since MECNs load balance index $\mathbb{L}$ directly affects the total computing latency T_m, we formulate the joint optimization problem to obtain the optimal collaborative path planning and scheduling strategies $\mathbf{\Psi} = \{\psi_1, \psi_2, ..., \psi_q, \psi_{q+1}, ..., \psi_N\}$ as follows:

$$\begin{aligned}
\mathcal{P}: \underset{\mathbf{\Psi}}{\arg\,\mathrm{Min}} & \left\{ \underbrace{\lambda_1 \mathbb{B}(t) + \lambda_2 T_m}_{joint\ optimization} + \underbrace{\sum_{\Delta} \sum_{\Delta'} \eta \left|\mathbb{N}_{\mathrm{g}\to\mathrm{g}'}(t)\right|}_{Penalty: L_1 regularization} \right\} \\
s.t.\ C\ a): & \max\left\{C_{occ}, C^{req}_{n,j}, C^{i}_{nm}, C^{j}_{nm}\right\} \leq C_{cap} \\
b): & T_{nm} \leq \sum_{j=1}^{J} T_n^{j,max}, \forall n \in \mathcal{N}, m \in \mathcal{M} \\
c): & \mathbb{N}_{\mathrm{g}\to\mathrm{g}'} \geq F_{\mathrm{g}}(t) \geq 0; F_{\mathrm{g}}(t) > F_{\mathrm{g}'}(t), \mathrm{g}' \in \mathcal{R}(\mathrm{g}) \\
& \mathbb{N}_{\mathrm{g}\to\mathrm{g}'} \leq \left|F_{\mathrm{g}'}(t)\right|; F_{\mathrm{g}'}(t) < 0, \mathrm{g}' \in \mathcal{R}(\mathrm{g}),
\end{aligned} \tag{19}$$

where λ_1, λ_2 are optimization weights, $\mathbb{N}_{\mathrm{g}\to\mathrm{g}'}(t)$ is the number of CAVs dispatched from section g to adjacent sections g′ at timeslot t, $\mathrm{g}' \in \mathcal{R}(\mathrm{g})$. $\mathcal{R}(\mathrm{g})$ denotes the adjacent road set of road section g. The objective is to minimize the road load imbalance index $\mathbb{B}$ and total computation latency T_m.

Meanwhile, to avoid over-scheduling, the number of vehicles scheduled to a neighbor region needs to be minimized while satisfying load balancing conditions. Specifically, if the objective is only to consider minimizing the $\lambda_1 \mathbb{B}(t) + \lambda_2 T_m$, there may exist multiple optimal scheduling strategies resulting in redundancy and unnecessary expense while does not significantly improve performance. For instance, two adjacent regions g and neighbor g′ , the computation capacity is all 50, and the load of g is 70, g′ is 20. In this case, there exist multiple optimal policies for g to schedule the load to g′, i.e. $(\mathbb{N}_{\mathrm{g}\to\mathrm{g}'}(t) = 20, \mathbb{N}_{\mathrm{g}'\to\mathrm{g}}(t) = 0)$, $(\mathbb{N}_{\mathrm{g}\to\mathrm{g}'}(t) = \{21, 22, ..., 29\}, \mathbb{N}_{\mathrm{g}'\to g}(t) = 0)$, $(\mathbb{N}_{\mathrm{g}\to\mathrm{g}'}(t) = 30, \mathbb{N}_{\mathrm{g}'\to\mathrm{g}}(t) = 0)$. However, all solutions except the first one are unnecessary. Hence, a complexity penalty term $\sum_{\mathrm{g}} \sum_{\mathrm{g}'} \eta \left|\mathbb{N}_{\mathrm{s}\to\mathrm{g}'}(t)\right|$ is added to the formula, which is $\mathcal{L}_1$ regularization with parameter η, avoiding unnecessary scheduling of vehicles to obtain the unique solution [4]. The formulated joint optimization problem is a complex non-linear programming (NP) problem. In the following, we present our proposed solution of DMARL.

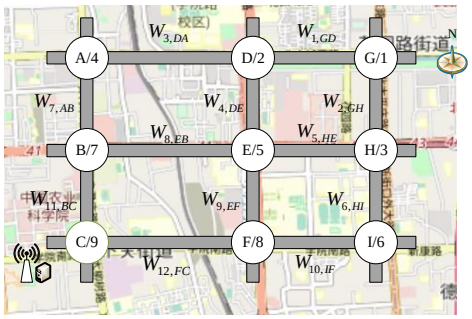


Fig. 4. Road networks.

V. Proposed Solution

This section introduces the proposed Q-learning-based DMARL algorithm for collaborative path planning and scheduling to solve the formulated joint optimization problem in detail.

A. DMARL Modeling

As shown in Fig. 4, we adopt a road map from the OpenStreetMap (OSM[1]) to construct the road networks, which is formulated as a grid with vertices and edges. Since the state of the traffic environment changes dynamically over time and space with a strong short-term correlation, one-off planning on a large area lacks effective foresight of the future environment. Thus, we select a 3-row, 3-column road map in a limited area and model it as a graph,

$$\boldsymbol{G}(t) = \langle \boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W}(t) \rangle, \tag{20}$$

where $\boldsymbol{V}$ is the vertex, $\boldsymbol{E}$ is the edge, and $\boldsymbol{W}(t)$ is the weight matrix of the edges which represents the computation load distribution and traffic situation. Each edge represents a road section, and the vertex is the intersection. The weight matrix $\boldsymbol{W}(t) = \{V_g(t), Load_m(t)\}, g \in \mathcal{G}, m \in \mathcal{M}$ characterizes the travel velocity and computation load of the MECNs, respectively. As shown in Fig. 5, we formulate the path planning and scheduling for different types of vehicles as two MDPs with

[1]https://www.openstreetmap.org/

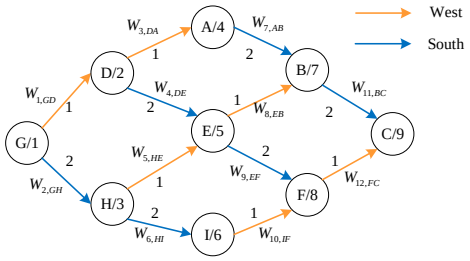


Fig. 5. Markov decision process for CAVs and COVs.

different reward functions, the numbering of the intersections corresponds to {G=1,D=2,H=3,A=4,E=5,I=6,B=7,F=8,C=9}, where node G stands for departure point and node C for destination. The theoretical driving trajectories from origin to destination corresponds to the road section serial number {1-3-7-11,1-4-8-11,1-4-9-12,2-5-8-11,2-5-9-12,2-6-10-12}. Based on the established MDPs, the collaborative agents, states, actions, and rewards of the DMARL are introduced as follows:

1) Agents: The collaborative agents are the objects that learn and explore from the environment, i.e., CAVs and COVs.

2) States: The states denote the location, category, and environment information of the agent at each step. The states of the collaborative agent n can be written as

$$\mathcal{S}_n(t) = \{position_i(t), class_i, \boldsymbol{W}_i(t)\}_{i=1,\cdots,N_a}, \tag{21}$$

where *position* represents the location of vehicles and MECNs that process the offloading tasks, *class* is the category of the vehicle, N_a is the number of collaborative agents, and $\boldsymbol{W}_i(t)$ is the state of environment which is changing dynamically with the actions of the agents.

3) Actions: The MDPs has two discrete southward and westward actions at each intersection, i.e. $(''south \downarrow'';'' west \leftarrow'')$. The set of actions from the origin to the destination of the agent n is $\mathcal{A}_n = \{a_{n1}, a_{n2}, \cdots, a_{nH}\}_{n=1,\cdots,N_a}$. The collaborative actions of multi-agent can be obtained as

$$\mathcal{A}_{N_a} = \begin{pmatrix} a_{11} & \cdots & a_{1H} \\ \vdots & \ddots & \vdots \\ a_{N_a 1} & \cdots & a_{N_a H} \end{pmatrix}_{K_{N_a}}, \tag{22}$$

where H is the action steps of each agent, and K_{N_a} is the signature of all collaborative agents.

4) Environment State Update: As shown in Fig. 6, the optimal Q-table of each agent is obtained by collaborative decisions and environment state updates. Since the decisions of agents affect the environment state and the decisions of other agents. Therefore, the environment state needs to be dynamically updated according to the action strategies. The cognitive engine needs to update the environment state $\boldsymbol{W}(t)$ based on agent action decisions $\mathcal{A}_{N_a}$. We consider two strategies, iterative update, which updates $\boldsymbol{W}(t)$ according to the action of each agent, and batch update is based on the group actions. The environmental state update according to Equations (3) and (6) can be expressed as

$$\boldsymbol{W}(t) = \begin{cases} [V_g(t), Load_m(t)] \Leftarrow \{act(\mathcal{A}_n)\} \text{ Iterative} \\ [V_g(t), Load_m(t)] \Leftarrow \{acts[\mathcal{A}_{N_a}]\} \text{ Batch} \end{cases}. \tag{23}$$

Then, the collaborative agents learn from the updated environment states and find the optimal decision strategies with the maximum rewards, maximizing global cumulative rewards.

5) Rewards: In the hybrid CIoVs scenarios, the rewards for different types of vehicles associated with the velocity V_g and the computation load $Load_m$ of the MECNs m. Normalizing the vectors V_g and $Load_m$, the rewards for different types of agents i and j are achieved as

$$R^{cav}_{i\in q}(\tau) = \sum\nolimits_{g\in\mathcal{G}} \left(\lambda e^{V_g} + \beta e^{2-Load_m}\right) e^{-\frac{F_g(\tau)}{\varphi}}, \tag{24}$$

$$R^{cov}_{j\in N-q}(\tau) = \sum\nolimits_{g\in\mathcal{G}} \left(\lambda e^{V_g}\right) e^{-\frac{F_g(\tau)}{\varphi}}, \tag{25}$$

where λ and β represent the weight coefficients, respectively, and φ is the penalty factor, the total cumulative rewards of the collaborating agents at step τ is

$$\begin{aligned} R_{Mix}(\tau) &= \sum\nolimits_{i=1}^{q} R^{cav}_{i\in q}(\tau) + \sum\nolimits_{j=1}^{N-q} R^{cov}_{j\in N-q}(\tau) \\ &= \sum\nolimits_{i=1}^{q} \left(\sum\nolimits_{g\in\mathcal{G}} \left(\lambda e^{V_g} + \beta e^{2-Load_m}\right) e^{-\frac{F_g(\tau)}{\varphi}}\right) \\ &+ \sum\nolimits_{j=1}^{N-q} \sum\nolimits_{g\in\mathcal{G}} \left(\lambda e^{V_g}\right) e^{-\frac{F_g(\tau)}{\varphi}}. \end{aligned} \tag{26}$$

Based on the above analysis, to solve the joint optimization problem in Section IV-E, the optimization objective is converted to maximize the total cumulative rewards R_{Mix} of all collaborating agents. That is, the global optimal collaborative path planning and scheduling strategy can be obtained by maximizing the cumulative reward R_{Mix}. Therefore, the joint optimization problem can be solved by maximizing R_{Mix},

$$\underset{s_i(\tau)\in\ \mathcal{S},\ a_i(\tau)\in\ \mathcal{A}}{\text{Max}}\ R_{Mix}(\tau)\ \ \mathcal{S} \subseteq \boldsymbol{G}(t) \tag{27}$$

B. Q-Learning-Based DMARL Algorithm

As shown in Algorithm 1, to maximize the total cumulative rewards R_{Mix} of collaboration, we propose Q-Learning-based DMARL algorithm for collaborative path planning and scheduling. We design the multi-agent reinforcement learning model as a tuple $\{N_a, \mathcal{S}, (a_1, \cdots, a_{N_a}), (r_1, \cdots, r_N), \gamma, Tp(s'|s,a)\}$, $\mathcal{S}$ is the environment states based on the established graph network $\boldsymbol{G}$(t) [43]. Where $(a_1, \cdots, a_{N_a})$, $(r_1, \cdots, r_{N_a})$ are the set of actions and rewards of the agents, respectively, $Tp(s'|s,a)$ is the state transfer probability function representing the probability that the agent performs action a in the current state s to transfer to the next state s'; $\gamma \in [0,1)$ is discount factor representing the degree of decay for future rewards, $R_n(s'|s,\ a)$ is the reward function, representing the reward for agent n taking action a in state s [44], [45].

The state, action, and reward update trajectory from departure to the destination of each agent is $(s,a,r) = \{s_\tau, a_\tau, r_\tau; s_{\tau+1}, a_{\tau+1}, r_{\tau+1}; \cdots; s_{\tau+H}, a_{\tau+H}, r_{\tau+H}\}$. The

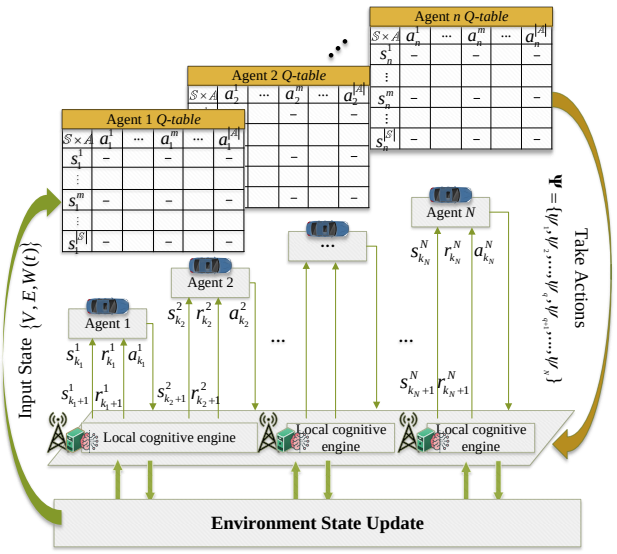


Fig. 6. The Q-learning-based DMARL model.

reward r further away from the current state, the more severe the reward decay. $Q(s,a)$ is the state-action value function, representing the reward for acting a in a particular state s. The goal of each agent is to maximize the cumulative discounted reward expectation,

$$Q^*(s_\tau, a_\tau) = \max_{\pi} \mathbb{E}\left[R_\tau \,|s_\tau \in \mathcal{S}, a_\tau \in \mathcal{A}\right], \tag{28}$$

where $\mathbb{E}$ represents expectation, which evaluates the merit of acting a_τ in the state s_τ.

To achieve the maximum cumulative reward and the optimal strategy $\pi^*(s) = \arg\max_{a\in\mathcal{A}} Q^*(s,a)$, we propose the $\varepsilon - greedy$ decay algorithm to explore and exploit the action space efficiently. The agent randomly selects the actions with probability ε and chooses the behavior corresponding to the largest value Q^* in the Q-table with probability $1-\varepsilon$, $Q^*(s,a) = \max Q(s,a), a \in \mathcal{A}$. The action selection of the agent can be expressed as

$$a = \begin{cases} \arg\max\limits_{a\in\mathcal{A}} Q(s,a), & \text{probability 1-}\varepsilon \\ random, & \text{probability } \varepsilon \end{cases}. \tag{29}$$

The agent requires more exploration when it is unfamiliar with the environment, i.e., ε decays as learning proceeds. Since the agents start learning unaware of their environment, thus require a large proportion of random actions to gain experience. We design the dynamic decay strategy to adjust the ratio of exploitation to exploration for $\varepsilon - greedy$ algorithm. As the number of episodes increases, the ε is gradually reduced until the end of training or the minimum value is reached. The dynamic decay update function of the $\varepsilon - greedy$ can be written as

$$\varepsilon(\tau+1) = \varepsilon(\tau) \times (1 - \mathcal{E}_{De}), \tag{30}$$

where $\mathcal{E}_{De}$ is the decay factor of ε. The ε value is updated iteratively for each episode.

Based on the reward of environmental feedback, DMARL algorithm employ time-difference to update the Q-value critic

Algorithm 1 The Q-Learning-Based DMARL Algorithm

Initialization: hyperparameters, $\mathcal{N}$, $\mathcal{M}$, and parameters $\varepsilon(\cdot)$; Initialize traffic flow $F_g(t)$, state matrix $\boldsymbol{W}(t)$, Q-function $Q(\cdot)$.
for each agent **do**
  Get current environment status updates $\boldsymbol{W}(t)$;
  Establish the MDPs of CAVs and COVs;
  for $episode = 1 : max\_episode$ **do**
    if $\varepsilon > 0.01$ **then**
      $\varepsilon = \varepsilon * (1 - \mathcal{E}_{De})$;
    else
      $\varepsilon = \varepsilon$;
    end if
    $S = S_0$;
    $is\_terminated = False$;
    while $is\_terminated$! **do**
      $a(\tau) = \begin{cases} \arg\max\limits_{a\in\mathbb{A}} Q(s,a), & \text{probability 1-}\varepsilon \\ random\ selection, & \text{probability } \varepsilon \end{cases}$
      Move every agent to the next state $s(\tau+1)$ by taking the action $a(\tau)$ and get rewarded R;
      if $s(\tau+1)$!= 'terminal' **then**
        Calculate $Q\_target = r + \gamma \max\limits_{a'\in\mathbb{A}} Q(s'_\tau, a'_\tau)$;
      else
        $Q\_target = R$;
        $is\_terminated = True$;
      end if
      Update $Q'(s_\tau, a_\tau)$ using Eq. (31);
      $s' \leftarrow s(\tau+1)$;
      $episode = episode + 1$
    end while
  end for
  if $agent \in V^{cav}$ **then**
    Calculate $R^{cav}_{i\in q}$ for CAV_i using Eq. (24);
  else
    Calculate $R^{cov}_{j\in N-q}$ for COV_j using Eq. (25);
  end if
  Update environment state $\boldsymbol{W}(t)$ using Eq. (23);
  Calculate total rewards $R_{Mix}(\tau)$ using Eq. (26);
end for

$Q(s_\tau, a_\tau)$. We derive the update strategy for the Q-value function as follows:

$$Q'(s_\tau, a_\tau) = Q(s_\tau, a_\tau) + \alpha \left\{ \underbrace{\overbrace{r + \gamma \max_{a'\in\mathcal{A}} Q(s'_\tau, a'_\tau)}^{critic\ loss\ function} - Q(s_\tau, a_\tau)}_{estimation} \right\}, \tag{31}$$

where the part in brackets is the loss function and the learning rate is $0 < \alpha < 1$.

C. Computation Complexity Analysis

For the centralized MARL approach, the dimensionality of the Q-table grows exponentially as the number of collaborative agents increases will result in slow convergence that cannot satisfy the low latency requirements [38]. Each agent with the sizes of states and actions $|\mathbb{S}|$ and $|\mathbb{A}|$, respectively. Since the agents need to explore all possible joint action cases, the

TABLE II
THE SIMULATION PARAMETERS

Simulation parameters	Value
Maximum velocity limit v_g	60 Km/h
Learning rate α	0.9 [38]
Discount factor γ	0.9
Penalty factor φ	5
Epsilon greedy ε	0.9
Epsilon decay factor $\mathcal{E}_{De}$	0.01
Reward weight λ, β	$[1, 1]$
Maximum episode	120
Batch size	50
Section length	1Km
Number of lanes	3
Number of collaborative vehicles	200
CAV ratio χ	0.5
Vehicles of congested road N_{jam}	300
Minimum computation resources $C^j_{\min}$	0.1 G CPU cycles/s [46]
Computing capability of MECNs $C^{\max}_m$	100 GHz [47]
Computing workload for one-bit data	500 CPU cycle/bit [48]
SINR	20dB [49]
Data volume of task j $D^j_{n,m}$	0.5Mbits [48]
Bandwidth B_m	20MHz [31]

state and action space of Q-table for centralized multi-agent Q-learning is $\mathbf{\Omega}^c_{multi} = |\mathbb{S}|^{N_a} \times |\mathbb{A}|^{N_a}$, which is exponential growth with the number of collaborative agents. Therefore, the MARL with centralized learning algorithm is hard to achieve online real-time collaborative decisions. In addition, due to the dynamic nature of traffic demand in CIoVs, the centralized solution does not match the temporal correlation of vehicle decisions.

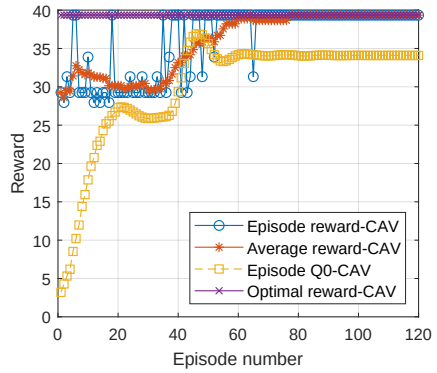
To deal with the dimensional explosion issue and reduce the algorithm complexity, the proposed Q-learning-based DMARL algorithm and iterative updating policy to obtain the optimal solution quickly with low complexity. As shown in Fig. 6, each agent corresponds to a Q-table of decision strategies with size $|\mathbb{S}| \times |\mathbb{A}|$, and the total distributed Q-tables dimension space of all agents is $\mathbf{\Omega}^d_{multi} = N_a \times |\mathbb{S}| \times |\mathbb{A}|$, much smaller than $\mathbf{\Omega}^c_{multi} = |\mathbb{S}|^{N_a} \times |\mathbb{A}|^{N_a}$ of centralized MARL.

VI. EXPERIMENT RESULTS AND DISCUSSIONS

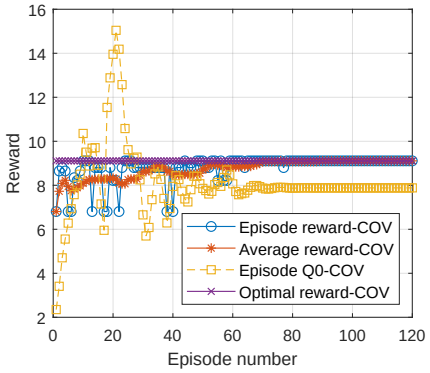
In this section, we conduct experiments to evaluate the effectiveness of the proposed approach based on the established road networks as shown in Fig. 4.

A. Simulation Settings

As shown in Fig. 4, we select a map with $2km \times 3km$ square area, which is pulled from OSM and mapped as a 3×3 road network. Each node represents an intersection and each edge is a road segment. In order to achieve different vehicle velocities of road sections and evaluate the performance of the proposed algorithm under different parameters, we set different traffic volume in the selected road section of the map to be $N_{g1} = [80, 89, 110, 66, 70, 95, 30, 60, 26, 161, 88, 72]$ and $N_{g2} = [75, 84, 105, 61, 65, 90, 25, 55, 21, 156, 83, 67]$ respectively. Moreover, the traffic flow F_g is set to be $F_g = [-4, -3, 6, -5, -5, -4, -3, 1, 4, 5, 1, 1]$, $N_{jam} = 300$. We consider the number of MECNs M is 12 which are deployed on each road segment to execute offloading computing tasks



(a) The training process of CAVs



(b) The training process of COVs

Fig. 7. The convergence of DMARL.

[46]. The total number of collaborative vehicles is N_a = 200 and each CAV has 5 offloading tasks, $J = 5$ [50]. We perform the simulations on a computer with Intel(R) Core(TM) i5-8250U CPU @ 1.60 GHz, 8 GB 1600 MHz DDR4 RAM. Table II summarizes the other main simulation parameters. For the fairness of comparison in simulation, each scheme adopts the same simulation parameters as Table II. To verify the effectiveness of the proposed algorithm, we compare the performances with the other schemes as follows:

- Non-collaborative scheme: Each vehicle makes decisions individually without environmental state updates, as labelled as "Noncollaboration scheme" [23].
- Non-joint optimization scheme: The vehicles optimize only the travel time without considering the computation latency, as labelled as "RoadLoadoptimal scheme" [6].
- Random selection scheme: The vehicle selects the path in a random way, rather than according to the traffic situation and updated state of the environment, as labelled as "Random scheme" [14].
- Proposed batch collaboration scheme: Collaborative vehicles perform distributed collaborative decision making in a batch update fashion where the environmental state is updated in a batch-based manner, as labelled as "Batch-Collaboration scheme".
- Proposed collaborative scheme: Each vehicle makes collaborative decision based on a real-time iterative update of the environmental state employing iterative update strategy, as labelled as "Collaboration scheme".

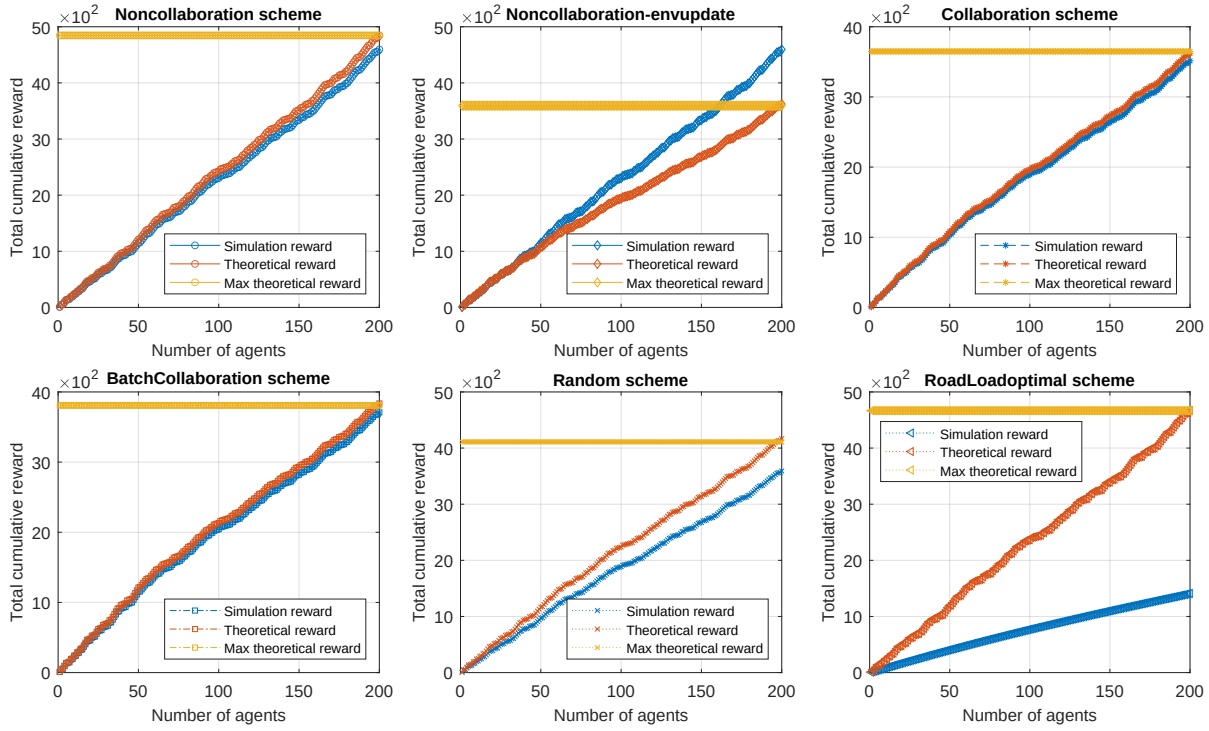


Fig. 8. The global convergence of different schemes.

B. The Convergence of Proposed Algorithm

This subsection shows the training process and analyzes the convergence of DMARL for different types of vehicles in a hybrid driving scenarios. As shown in Fig. 7, both agents CAVs and COVs can quickly converge at around 70 episodes. Since they have different service demands and reward functions, the optimal rewards for convergence are different. Fig. 7 (a) shows the training process of the agent CAVs, we can find that the optimal cumulative rewards $R^{cav}_{i\in q}(\tau)$ = 39.3441 equals to the theoretical optimal value. The estimated long-term discounted reward $Q0$ gradually converges to 34.0894 due to $\gamma < 1$. Similarly, Fig. 7 (b) shows the training process of COVs. The optimal reward $R^{cov}_{j\in N-q}(\tau)$ = 9.1096, $Q0$ = 7.8817 which are all achieve the theoretical optimal value for COVs. The above results show that different types of vehicles are able to find the optimal decision strategy to meet their service requirements, which provides a practical and feasible basis for distributed collaborative decision making. Fig. 8 shows the global convergence of different optimization approaches. As shown in the Fig. 8, the "Noncollaboration scheme" cannot converge to the theoretical value for the actual updated environment state since it cannot obtain the state updates caused by the decisions of other agents. The rewards R_{Mix} obtained by the proposed "Collaboration scheme" and "BatchCollaboration scheme" are closest to the theoretical optimal value, which illustrates the rapid convergence of the proposed distributed collaborative approach. The reason is that the collaborative solutions can update the environment state based on the agent's decision to obtain the global optimal strategy. The "RoadLoadoptimal scheme" has the worst convergence due to its poor performance in terms of computational latency and MECNs load balancing, as it only considers the optimization of road load. The simulation results demonstrate that the proposed approach is able to accurately find the globally optimal path and scheduling strategy that maximizes the cumulative reward R_{Mix}. In the following, we will further verify the performance of the proposed approach on each performance metric.

C. Consensus Latency and Throughput of Blockchain System

This subsection shows the consensus latency and throughput of blockchain system. Since attacks by malicious nodes will slow down the consensus process, the consensus latency under untrusted conditions reflects the security of blockchain systems. To verify the security of blockchain system, we conduct experiment to evaluate the consensus latency under different number of consensus nodes compared with different protocols. We select two consensus protocols for comparison, including DBFT [32] and PBFT [33] consensus protocols. As shown in Fig. 9, the proposed CDBFT consensus algorithm has achieved the lowest consensus latency. Moreover, it can be seen that with the increase of the credit node ratio, the consensus latency gradually decreases. In addition, we conduct experiments and verify the throughput of the blockchain system, i.e., the number of decision transactions which successfully reaching consensus per second [51], under the condition that the consensus node number CNs is 30. It can be seen from the Fig. 10 that our proposed CDBFT consensus protocol obtains the highest throughput, which can process tens of thousands of decision transactions per second. The simulation results show that our proposed CDBFT consensus algorithm has the lowest consensus latency compared with other consensus protocols,

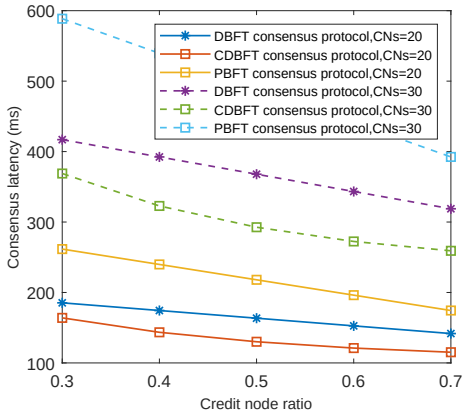


Fig. 9. Consensus latency of blockchain system.

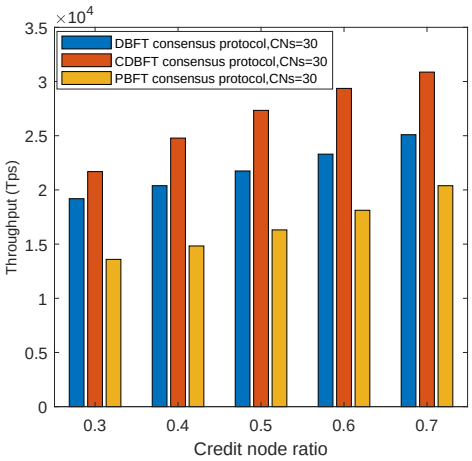


Fig. 10. Throughput of blockchain system.

indicating that our proposed consensus protocol can resist attacks by malicious nodes. Furthermore, the results also show that the proposed consensus protocol can efficiently support the application of the proposed DMARL algorithm in large-scale vehicle collaborative decision making scenarios.

D. The Effectiveness of Proposed Approach

Fig. 11 and Fig. 13 show the road load balance index $\mathbb{B}$ under different parameters of N_{g1} and N_{g2}, respectively. As can be seen from the figure, our proposed collaboration approach achieves a smaller value compared to the "Noncollaboration scheme" and "Random scheme". Moreover, all curves fall and then rise as the number of vehicles increases. To explain, the "Noncollaboration scheme" does not consider the impact of vehicle decisions on the environment, and the selfish behavior leads to the aggregation of vehicles. Meanwhile, the "Road-Loadoptimal scheme" approach considers only the load of the road networks, which is why it achieves the lowest $\mathbb{B}$. The proposed collaborative approach can perceive environmental state updates caused by vehicle decisions, thus achieving a better load balancing performance than other schemes. Fig. 12 and Fig. 14 show the MECNs load balance index $\mathbb{L}$ under different parameters. The results prove that the proposed algorithm achieved better load balancing performance compare with

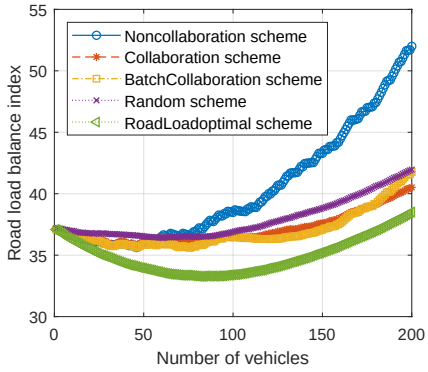


Fig. 11. The road load balance index $\mathbb{B}$ ($N_g = N_{g1}$).

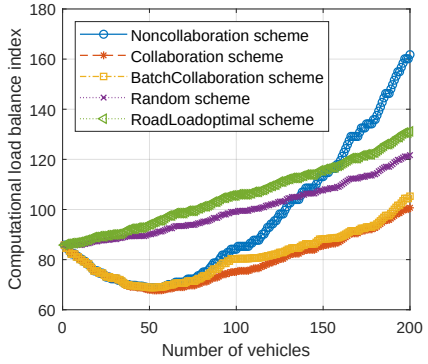


Fig. 12. The MECNs load balance index $\mathbb{L}$ ($N_g = N_{g1}$).

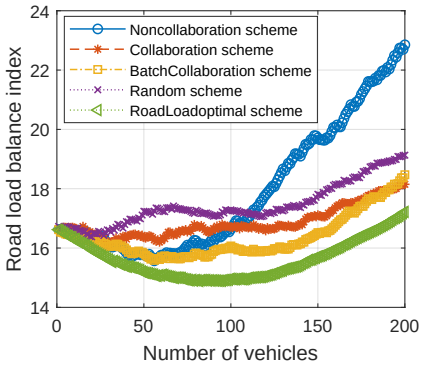


Fig. 13. The road load balance index $\mathbb{B}$ ($N_g = N_{g2}$).

other algorithms under different parameters of N_{g1} and N_{g2}. In contrast to Fig. 11 and and Fig. 13, the "RoadLoadoptimal scheme" has the largest value indicating a severe imbalance load distribution of MECNs, which will seriously affect the computation latency of CAVs. The reason is that it ignores the optimization of the computation load of MECNs in hybrid driving scenarios. Our proposed collaborative approaches also achieve the optimal results. To explain, the CAVs agents can be scheduled to light-loaded MECNs based on load distribution and vehicle decisions, avoiding vehicle aggregation and MECNs overload.

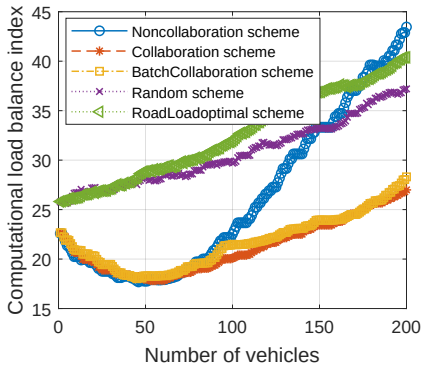


Fig. 14. The MECNs load balance index $\mathbb{L}$ ($N_g = N_{g2}$).

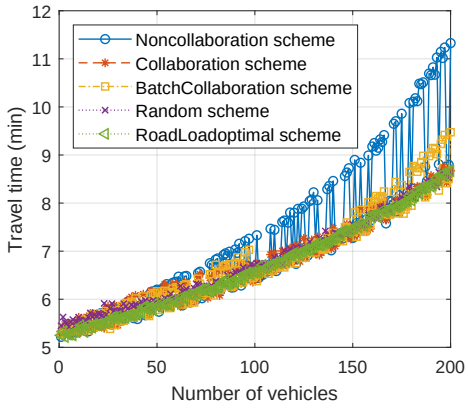


Fig. 15. The travel time of each vehicle.

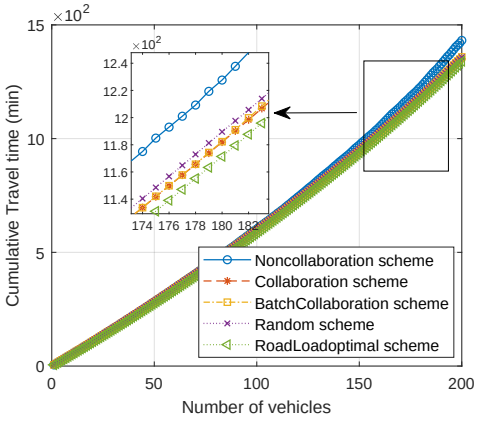


Fig. 16. The cumulative travel time of collaborative vehicles.

E. The Travel Time and Computation Latency

In this subsection, we evaluate the performance of the proposed algorithm on vehicle travel time and computation latency compared with different optimization approaches. Fig. 15 shows the travel time for each vehicle. The proposed approaches and the "RoadLoadoptimal scheme" are shorter than other schemes. We also find that all curves increase gradually with the increase of vehicles. The reason is that the vehicle density of the road affects the vehicle velocity. The roads will become congested as the number of vehicles increases. The results show that the proposed collaborative approaches can effectively carry out the load balancing of road and avoid congestion caused by vehicle aggregation, which promotes the reduction of travel time.

To further validate the global collaboration performance, we compared the total cumulative travel time of the collaborating agents. As shown in Fig. 16, the cumulative travel time of "Noncollaboration scheme" has the longest travel time. The proposed "Collaboration scheme" and "BatchCollaboration scheme" are close to the "RoadLoadoptimal scheme" which are shorter than the other schemes. This is because the aggregation effect of the "Noncollaboration scheme" tends to lead to secondary congestion on the roadway, increasing travel time. Since "RoadLoadoptimal scheme" focuses only on optimizing the road load, the shortest travel time is achieved. The results show that our proposed approach can obtain globally optimal paths and scheduling strategies to uniformly distribute vehicles on the road network, thus efficiently utilizing road infrastructure and reducing traffic congestion.

We evaluate the computation latency of CAV n regarding T_{nm} and T_m of completing the offloaded tasks. Fig. 17 shows the computation latency T_{nm} of each vehicle. It can be seen that the proposed "Collaboration scheme" has the lowest computation latency while the "Noncollaboration scheme" grows exponentially with the increase of vehicles. The "Random scheme" randomly selects roads without considering traffic situations, which is why it has a longer computation latency close to that of the "RoadLoadoptimal scheme". To explain, as the load on MECNs increases, the computation resources allocated to individual tasks decrease.

Fig. 18 shows the total cumulative computation time versus the number of collaborative vehicles. It can be seen that the "RoadLoadoptimal scheme" has the longest cumulative computation time since it only focuses on optimizing the road load balance index $\mathbb{B}$ and ignores the MECNs load balance index $\mathbb{L}$. The "Random scheme" also has a long cumulative computation time since its random nature resulting in the inability to obtain the optimal paths with a light computation load. The proposed "Collaboration scheme" achieve the shortest latency since it can perform globally optimal collaborative scheduling and avoid overload of MECNs. The results also shows that all curves rise dramatically with the increase of vehicles while our approaches are still significantly lower than other schemes. The above results demonstrate the effectiveness of our proposed approach, which achieve better load balancing performance to meet the traffic and computational demands of different types of vehicles in hybrid driving scenario.

VII. Conclusions and Future Works

In this paper, we consider a hybrid driving scenario and propose a DMARL algorithm for collaborative path planning and scheduling in blockchain-based CIoVs to satisfy the diverse service requirements of different types of vehicles. We design a blockchain-based collaborative framework to support distributed decision making for global collaborative optimization, effectively avoiding the aggregation effect caused by individual decisions. By modeling the communication, traffic

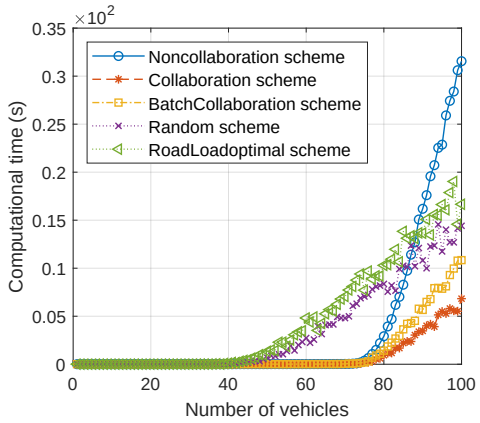


Fig. 17. The computation latency of each vehicle.

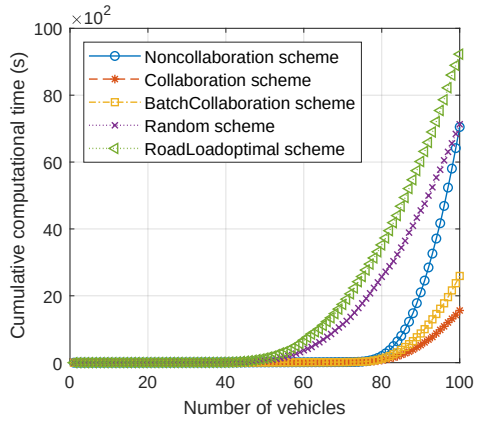


Fig. 18. The cumulative computation time of collaborative vehicles.

situation and computational models, we analyze the distributed collaborative consensus latency and formulate a joint optimization problem to minimize both travel time and computation latency. Finally, to solve the problem, we reformulate the scheduling of different types of vehicles as MDPs and propose a Q-learning-based DMARL algorithm to obtain the globally optimal collaborative path planning and scheduling strategy and achieve proactive load balancing of both road infrastructure and MECNs. Simulation results show that the proposed approach achieves better performance compared with the benchmark schemes in terms of load balancing indexes, travel time, and computation latency. In future work, we will consider an incentive scheme and explore the social properties of vehicles to facilitate intelligent collaborations among vehicles and safeguard the service requirements of high-priority vehicles in CIoVs.

REFERENCES

[1] H. Lu, Q. Liu, D. Tian, Y. Li, H. Kim, and S. Serikawa, "The cognitive Internet of vehicles for autonomous driving," *IEEE Network*, vol. 33, no. 3, pp. 65–73, 2019.

[2] L. Liu, J. Feng, X. Mu, Q. Pei, D. Lan, and M. Xiao, "Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2023.

[3] Y. Song, Y. Fu, F. R. Yu, and L. Zhou, "Blockchain-enabled internet of vehicles with cooperative positioning: A deep neural network approach," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3485–3498, 2020.

[4] J. Li, G. Luo, N. Cheng, Q. Yuan, Z. Wu, S. Gao, and Z. Liu, "An end-to-end load balancer based on deep learning for vehicular network traffic control," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 953–966, 2019.

[5] Y. Xu, H. Zhou, T. Ma, J. Zhao, B. Qian, and S. Shen, "Leveraging multi-agent learning for automated vehicles scheduling at non-signalized intersections," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 427–11 439, 2021.

[6] N. Sun, H. Shi, G. Han, B. Wang, and L. Shu, "Dynamic path planning algorithms with load balancing based on data prediction for smart transportation systems," *IEEE Access*, vol. 8, pp. 15 907–15 922, 2020.

[7] G. Sun, K. Liu, G. O. Boateng, G. Liu, and W. Jiang, "Intelligent cruise guidance and vehicle resource management with deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3574–3585, 2022.

[8] T. Wu, P. Zhou, K. Liu, Y. Yuan, X. Wang, H. Huang, and D. O. Wu, "Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8243–8256, 2020.

[9] C. Li, Y. Fu, F. R. Yu, T. H. Luan, and Y. Zhang, "Vehicle position correction: A vehicular blockchain networks-based GPS error sharing framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 898–912, 2020.

[10] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. Leung, "Blockchain and machine learning for communications and networking systems," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1392–1431, 2020.

[11] M. B. Mollah, J. Zhao, D. Niyato, Y. L. Guan, C. Yuen, S. Sun, K.-Y. Lam, and L. H. Koh, "Blockchain for the Internet of vehicles towards intelligent transportation systems: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4157–4185, 2021.

[12] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and Y. Zhang, "An autonomous lane-changing system with knowledge accumulation and transfer assisted by vehicular blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 11 123–11 136, 2020.

[13] H. Chang, Y. Liu, and Z. Sheng, "Blockchain-enabled online traffic congestion duration prediction in cognitive internet of vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25 612–25 625, 2022.

[14] K. Lin, C. Li, G. Fortino, and J. J. Rodrigues, "Vehicle route selection based on game evolution in social Internet of vehicles," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2423–2430, 2018.

[15] J. Pan, I. S. Popa, and C. Borcea, "Divert: A distributed vehicular traffic re-routing system for congestion avoidance," *IEEE Transactions on Mobile Computing*, vol. 16, no. 1, pp. 58–72, 2017.

[16] J. Xie, S. Xiao, Y.-C. Liang, L. Wang, and J. Fang, "A throughput-aware joint vehicle route and access network selection approach based on smdp," *China Communications*, vol. 17, no. 5, pp. 243–265, 2020.

[17] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4377–4387, 2019.

[18] Z. Li, C. Wang, and C.-J. Jiang, "User association for load balancing in vehicular networks: An online reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2217–2228, 2017.

[19] J. Cui, F. Ouyang, Z. Ying, L. Wei, and H. Zhong, "Secure and efficient data sharing among vehicles based on consortium blockchain," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2021.

[20] H. Chai, S. Leng, Y. Chen, and K. Zhang, "A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in Internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 3975–3986, 2021.

[21] Y. He, K. Huang, G. Zhang, F. R. Yu, J. Chen, and J. Li, "Bift: A blockchain-based federated learning system for connected and autonomous vehicles," *IEEE Internet of Things Journal*, pp. 1–12, 2021.

[22] X. Jiang, F. R. Yu, T. Song, Z. Ma, Y. Song, and D. Zhu, "Blockchain-enabled cross-domain object detection for autonomous driving: A model sharing approach," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3681–3692, 2020.

[23] K. Lin, C. Li, Y. Li, C. Savaglio, and G. Fortino, "Distributed learning for vehicle routing decision in software defined internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3730–3741, 2021.

[24] A. Guillen-Perez and M.-D. Cano, "Multi-agent deep reinforcement learning to manage connected autonomous vehicles at tomorrows intersections," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7033–7043, 2022.

[25] L. Ren, X. Fan, J. Cui, Z. Shen, Y. Lv, and G. Xiong, "A multi-agent reinforcement learning method with route recorders for vehicle routing in supply chain management," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2022.

[26] W. Qin, Y.-N. Sun, Z.-L. Zhuang, Z.-Y. Lu, and Y.-M. Zhou, "Multi-agent reinforcement learning-based dynamic task assignment for vehicles in urban transportation system," *International Journal of Production Economics*, vol. 240, p. 108251, 2021.

[27] D. Kwon and J. Kim, "Multi-agent deep reinforcement learning for cooperative connected vehicles," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[28] S. Islam, S. Badsha, S. Sengupta, H. La, I. Khalil, and M. Atiquzzaman, "Blockchain-enabled intelligent vehicular edge computing," *IEEE Network*, vol. 35, no. 3, pp. 125–131, 2021.

[29] W. Xu, Z. Yang, D. W. K. Ng, M. Levorato, Y. C. Eldar, and M. Debbah, "Edge learning for b5g networks with distributed signal processing: Semantic communication, edge computing, and wireless sensing," *IEEE journal of selected topics in signal processing*, vol. 17, no. 1, pp. 9–39, 2023.

[30] Y. Cang, M. Chen, Z. Yang, Y. Hu, Y. Wang, C. Huang, and Z. Zhang, "Online resource allocation for semantic-aware edge computing systems," *IEEE Internet of Things Journal*, 2023.

[31] Z. Ning, K. Zhang, X. Wang, M. S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, B. Sadoun, and R. Y. Kwok, "Joint computing and caching in 5G-envisioned internet of vehicles: A deep reinforcement learning-based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5201–5212, 2021.

[32] J. Zhang, Y. Rong, J. Cao, C. Rong, J. Bian, and W. Wu, "DBFT: A byzantine fault tolerance protocol with graceful performance degradation," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 5, pp. 3387–3400, 2021.

[33] G. Sun, M. Dai, J. Sun, and H. Yu, "Voting-based decentralized consensus design for improving the efficiency and security of consortium blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6257–6272, 2021.

[34] U. Javaid, M. N. Aman, and B. Sikdar, "A scalable protocol for driving trust management in Internet of vehicles with blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 815–11 829, 2020.

[35] 3GPP, "Digital cellular telecommunications system; Universal Mobile Telecommunications System (UMTS); LTE; 5G," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 21.916, 01 2022, version 16.1.0. [Online]. Available: https://www.3gpp.org/release-16

[36] J. Zhou, D. Tian, Y. Wang, Z. Sheng, X. Duan, and V. C. Leung, "Reliability-optimal cooperative communication and computing in connected vehicle systems," *IEEE Transactions on Mobile Computing*, vol. 19, no. 5, pp. 1216–1232, 2020.

[37] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled Internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[38] S. Lee, H. Yu, and H. Lee, "Multi-agent q-learning based multi-uav wireless networks for maximizing energy efficiency: Deployment and power control strategy design," *IEEE Internet of Things Journal*, vol. 9, no. 9, pp. 6434–6442, 2022.

[39] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. Leung, "Decentralized resource allocation for video transcoding and delivery in blockchain-based system with mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11 169–11 185, 2019.

[40] X. Lin, J. Wu, S. Mumtaz, S. Garg, J. Li, and M. Guizani, "Blockchain-based on-demand computing resource trading in iov-assisted smart city," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1373–1385, 2021.

[41] F. Kandah, B. Huber, A. Skjellum, and A. Altarawneh, "A blockchain-based trust management approach for connected autonomous vehicles in smart cities," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2019, pp. 0544–0549.

[42] L. Yang, H. Yao, J. Wang, C. Jiang, A. Benslimane, and Y. Liu, "Multi-uav-enabled load-balance mobile-edge computing for iot networks," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6898–6908, 2020.

[43] I. Althamary, C.-W. Huang, and P. Lin, "A survey on multi-agent reinforcement learning methods for vehicular networks," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 1154–1159.

[44] J. Yang, J. Zhang, and H. Wang, "Urban traffic control in software defined Internet of Things via a multi-agent deep reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3742–3754, 2020.

[45] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "A survey and critique of multiagent deep reinforcement learning," *Autonomous Agents and Multi-Agent Systems*, vol. 33, no. 6, pp. 750–797, 2019.

[46] X. Ma, A. Zhou, S. Zhang, and S. Wang, "Cooperative service caching and workload scheduling in mobile edge computing," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 2076–2085.

[47] Z. Song, R. Ma, and Y. Xie, "A collaborative task offloading strategy for mobile edge computing in Internet of vehicles," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5. IEEE, 2021, pp. 1379–1384.

[48] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031–5044, 2019.

[49] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, "The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car," *IEEE vehicular technology magazine*, vol. 14, no. 2, pp. 103–111, 2019.

[50] J. Feng, W. Feng, and S. Lin, "Reverse computing offloading for enhanced computing capacity in cooperative vehicle infrastructure system," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 1011–1016.

[51] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, "Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability," in *2019 IEEE international conference on blockchain (Blockchain)*. IEEE, 2019, pp. 536–540.

Huigang Chang received his M.S. degree from the School of Computer Science and Technology, Henan Polytechnic University in 2018, and the Ph.D. degree in Information and Communication Engineering from Beijing University of Posts and Telecommunications (BUPT) in 2023. He is currently an Researcher at China Intelligent and Connected Vehicles (Beijing) Research Institute Co.,Ltd, Beijing 100176, China. His research interests are in the area of wireless communication, Internet of Vehicles, and blockchain. Email: changhuigang@bupt.edu.cn

Yiming Liu received the B.E. degree in Communication Engineering from Shanghai University in 2014, and the Ph.D. degree in Information and Communication Engineering from Beijing University of Posts and Telecommunications (BUPT) in 2019. In 2017 and 2018, she was a visiting Ph.D. student at the University of British Columbia. She is currently an Associate Researcher at the School of Information and Communication Engineering, BUPT. Her current research interests include next generation wireless networks, edge intelligence, blockchain, and distributed ledger technology. Email: liuyiming@bupt.edu.cn

Zhengguo Sheng (Senior Member, IEEE) received the B.Sc. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2006, and the M.S. and Ph.D. degrees from Imperial College London, London, U.K., in 2007 and 2011, respectively. He is currently a Reader with the University of Sussex, Brighton, U.K. Previously, he was with UBC, Vancouver, BC, Canada, as a Research Associate and with Orange Labs, Santa Monica, CA, USA, as a Senior Researcher. He has more than 120 publications. His research interests cover IoT, vehicular communications, and cloud/edge computing.