Joint Computing and Caching in 5G-Envisioned Internet of Vehicles: A Deep Reinforcement Learning-Based Traffic Control System

Zhaolong Ning, *Senior Member, IEEE*, Kaiyuan Zhang, Xiaojie Wang, Mohammad S. Obaidat, *Life Fellow, IEEE*, Lei Guo, Xiping Hu, Bin Hu, *Senior Member, IEEE*, Yi Guo, Balqies Sadoun, and Ricky Y. K. Kwok, *Fellow, IEEE*

***Abstract*— Recent developments of edge computing and content caching in wireless networks enable the Intelligent Transportation System (ITS) to provide high-quality services for vehicles. However, a variety of vehicular applications and time-varying network status make it challenging for ITS to allocate resources efficiently. Artificial intelligence algorithms, owning the cognitive capability for diverse and time-varying features of Internet of Connected Vehicles (IoCVs), enable an intent-based networking for ITS to tackle the above-mentioned challenges. In this paper, we develop an intent-based traffic control system by investigating Deep Reinforcement Learning (DRL) for 5G-envisioned IoCVs, which can dynamically orchestrate edge computing and content caching to improve the profits of Mobile Network Operator (MNO). By jointly analyzing MNO's revenue and users' quality of experience, we define a profit function to calculate the MNO's profits. After that, we formulate a joint optimization problem to maximize MNO's profits, and develop an intelligent traffic control scheme by investigating DRL, which can improve system profits of the MNO and allocate network resources effectively. Experimental results based on real traffic data demonstrate our designed system is efficient and well-performed.**

***Index Terms*— Internet of connected vehicles, 5G, deep reinforcement learning, traffic control system, edge computing, content caching.**

Manuscript received July 27, 2019; revised November 29, 2019; accepted January 24, 2020. Date of publication February 5, 2020; date of current version August 9, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61971084, Grant 61771120, and Grant 61671092, in part by the Fundamental Research Funds for the Central Universities under Grant DUT19JC18, in part by the National Natural Science Foundation of Chongqing under Grant cstc2019jcyj-msxmX0208, in part by the open research fund of National Mobile Communications Research Laboratory, Southeast University, under Grant 2020D05, in part by the China Postdoctoral Science Foundation under Grant 2018T110210, and in part by the Shenzhen Science and Technology planning project under Grant JCYJ20170818111012390. The Associate Editor for this article was R. Venkatesha Prasad. *(Corresponding authors: Xiaojie Wang; Lei Guo; Xiping Hu; Bin Hu; Yi Guo.)*

Zhaolong Ning is with the Chongqing Key Laboratory of Mobile Communications Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China, also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China, and also with the School of Software, Dalian University of Technology, Dalian 116024, China (e-mail: z.ning@ieee.org).

Kaiyuan Zhang is with the School of Software, Dalian University of Technology, Dalian 116024, China (e-mail: zky123123@live.com).

Xiaojie Wang is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: xiaojie.wang@polyu.edu.hk).

Mohammad S. Obaidat is with the College of Computing and Informatics, The University of Sharjah, Sharjah 27272, UAE, also with the King Abdullah II School of Information Technology, The University of Jordan, Amman 24600, Jordan, and also with the University of Science and Technology Beijing, Beijing 100083, China (e-mail: m.s.obaidat@ieee.org).

Lei Guo is with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: guolei@cqupt.edu.cn).

Xiping Hu is with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: xp.hu@siat.ac.cn).

Bin Hu is with the School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China (e-mail: bh@lzu.edu.cn).

Yi Guo is with The Second Clinical Medical College (Shenzhen People's Hospital), Jinan University, Shenzhen 518053, China (e-mail: xuanyi_guo@163.com).

Balqies Sadoun is with the College of Engineering, Al-Balqa' Applied University, Al-Salt 19117, Jordan (e-mail: sadounbalqies@gmail.com).

Ricky Y. K. Kwok is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: ricky.kwok@hku.hk).

Digital Object Identifier 10.1109/TITS.2020.2970276

I. Introduction

With the emergence of the Fifth Generation (5G) wireless networks, various devices in communication networks are able to connect and exchange information more efficiently than ever before [1]. As an important application of Internet of Things (IoTs), the Internet of Connected Vehicles (IoCVs) gains tremendous capability to construct an efficient transportation system to serve vehicles. Intent-based networking along with the IoCV enables the development of Intelligent Transportation System (ITS), which plays an important role to improve the quality of lives for inhabitants in smart cities. With the increasing scale of IoCVs and diverse requirements of vehicular applications, Mobile Network Operators (MNOs) are urgent to design systems with high Quality-of-Service (QoS) and Quality of Experience (QoE), which can provide satisfactory services for their users.

Among various applications, traffic controlling among IoCVs has been a key issue in ITS, which schedules traffic signals dynamically to relieve the traffic congestion and improve the driving experience [2]. In IoCV-based traffic control systems, a vehicle can communicate with vehicles (V2V) or infrastructures (V2I) via wireless links supported by 5G communication technologies [3]. Compared with detector-based ones, IoCV-based traffic control systems can integrate the states of vehicles and transport information of facilities into traffic signal scheduling. However, due to the increasing traffic volumes in smart cities, it is challenging for the MNO to schedule limited resources to collect and process real-time traffic conditions to guarantee the flexible and efficient traffic controlling. Therefore, in order to promote productivity, improve efficiency and ensure the efficient handling of continuous growing data, an intent-based traffic control system, that can process the data with minimum delays, is necessary [4].

Mobile Edge Computing (MEC) is a promising paradigm, which can not only satisfy the strict deadline constraints of users, but also fully utilize network resources [5], [6]. As more vehicles are involved by 5G-envisioned networks, the ITS places part of storage, communication and computation resources at MEC servers equipped at Road Side Units (RSUs), so that they can execute tasks in a distributed manner to save backhaul bandwidths. Compared with the central Base Station (BS), MEC servers equipped at RSUs are lightweight with limited resources. Therefore, a feasible resource allocation scheme is urgent for ITS to perform delay-sensitive and computation-intensive tasks at edge servers. On one hand, although 5G is promising to guarantee the QoS of networks and QoE of users, how to take full advantages of channel resources and bandwidths is still challenging [7]. Non-Orthogonal Multiple Access (NOMA) is a promising solution, which enables multiple users to reuse some unoccupied subchannels to gain multiplexing benefits [8]. On the other hand, it is vital to develop the intent-based networking to integrate automation with intelligence, which brings the network strategy for ITS to a higher level to handle explosive data effectively.

There are a number of researches focusing on resource allocation issues in IoCVs. Binary decision for resource allocation is considered in [9], indicating that users' tasks are arbitrarily executed by vehicles or offloaded to edge servers. Optimization problems are formulated to maximize the execution rate and minimize the energy consumption. The authors in [10] propose a partial decision strategy, which allows tasks to be executed both locally and remotely on MEC servers. Based on the large converge area of the BS/cloud server, some studies focus on cooperative task execution paradigms to shorten the task execution time [11]. Although cooperative partial decisions for tasks are relatively reasonable, they are limited in time-varying systems for dynamic resource allocation. During the movement of vehicles, the network status and available resources of MEC servers change dynamically in practice. Thus, it is indispensable to propose a time-varying resource allocation scheme for ITS by considering dynamic status and task requests of vehicles.

Owning the cognitive capability for dynamic environments, Artificial Intelligence (AI) is proved to be promising to address traffic control and resource allocation problems [12]. Integrating with the AI technique, intent-based networking has the capability to efficiently simplify network operations and provide scalable services. Deep Reinforcement Learning (DRL) is an important branch of AI methods, which has gained a great success in many fields [13]. Some studies have applied DRL methods, including Q-learning and Deep Q Network (DQN), to schedule resources for users, since it is effective in joint resource management [14]. However, these algorithms may be unfeasible for the practical ITS due to their continuous action spaces. Moreover, most of them ignore the influence of vehicles' mobilities on computation and caching resource allocation.

Naturally, promoting users' QoE is an important goal for ITS. The authors in [15] propose a resource allocation scheme for ITS to minimize the execution time and guarantee users' QoE. The authors in [9] regard the maximal computation rate as the best QoE performance for users. Fulfilling the requirement of users' QoE brings MNOs with good reputations. However, the main purpose of MNOs is to gain great benefits of offering task computing and content downloading services. Thus, a trade-off should be considered between the QoE of system users and the operating revenue of MNOs to benefit both users and MNOs.

In this paper, we investigate edge computing and caching problem in IoCVs, where the MNO employs the intent-based networking paradigm on ITS to handle users' requests and allocate resources in IoCVs. In order to orchestrate network traffic and allocate limited resources dynamically, we integrate the DRL method with hierarchical IoCVs to maximize the profits function of the MNO. First, we construct an integrated architecture for communication, computing and caching, and fully consider the mobility and time-varying status in IoCVs. Then we formulate a joint task scheduling and resource allocation problem to maximize the profit of the MNO, which makes a trade-off between the QoE of users and benefits of MNOs. After that, we solve the formulated optimization problem by exploiting an advanced DRL method. The main contributions of this paper can be summarized as follows:

- We construct a hierarchical edge computing and caching model for time-varying IoCVs under 5G communication environment, in which the BS together with RSUs can deliver contents and execute tasks for vehicles cooperatively.
- We design a novel profit function to measure the performance of the MNO, which jointly considers MNO's revenue and users' QoE level. Then we formulate a joint optimization problem to maximize the benefit for the MNO by an intent-based traffic control scheme, which is proposed to assign tasks and allocate resources in IoCVs.
- By considering the dynamic communication conditions between vehicles and MEC servers, we design a mobility-aware edge computing and caching scheme. It can jointly optimize task assignment and resource allocation in a continuous space. Based on the Deep Deterministic Policy Gradient (DDPG) based model, the formulated problem can be solved effectively.
- We conduct experiments based on real traffic data in Hangzhou, China, to evaluate the system performance. Numerical results demonstrate that our proposed scheme allocates resource efficiently and achieves a better performance on improving the profits of MNOs compared with representative algorithms.

The rest of this paper is organized as follows. We review the related work in Section II, and present the system model in Section III. In Section IV, we formulate the edge computing and caching scheduling as an optimization problem and design a DRL-based system to solve the problem. Numerical results are provided in Section V, and Section VI concludes this paper.

II. Related Work

A. Resource Allocation in IoCVs

Appropriate resource allocation can promote the data transmission rate during vehicular communications, which can satisfy the requirements of latency-sensitive service deadline constraints [16], [17]. Relevant studies for resource allocation in IoCVs mainly focus on improving the performance of ITS in three areas, i.e., networking, caching, and computing. The authors in [18] consider different types of vehicular communication modes to design an optimal data transmission scheme with the purpose of minimizing transmission costs. A multi-tier caching scheme is proposed in [19], aiming to make content prefetching decisions with the help of edge caching for both users and service providers. The authors in [20] propose a joint computing offloading and content caching strategy to design an asymmetric search tree, which generates resource allocation strategies to minimize the total latency of the MEC system. However, these studies mainly focus on resource allocation schemes to benefit users without considering the profits for MNOs, who play important roles in constructing smart cities.

B. Energy Consumption in IoCVs

There exists some works that address the energy efficiency issue in the ITS by exploring communication, computation and caching resource allocation schemes [21], [22]. Hu *et al.* [23] put forward a model predictive fuel-optimal control scheme to solve energy, which utilized V2V/V2I communications to control state trajectories of the leading vehicle. Khan *et al.* [24] studied file placement on caching of servers in cooperative scenarios to maximize the energy efficiency in wireless networks. Zhou *et al.* [25] explored a cooperative two-hop D2D-V2V transmission in vehicular networks, which optimized the energy efficiency of the BS. However, these studies only considered one aspect for energy consumption, which can not provide a comprehensive analysis of the ITS among IoCVs.

C. AI Methods for IoCVs

AI has attracted great interests in addressing edge computing and caching issues. Recent developments in deep learning and especially the Deep Neural Network (DNN) have superiorities in accuracy and efficiency compared with traditional methods, such as convex optimization, game theory and machine learning [26]. In addition, DRL combining DNN and Reinforcement Learning (RL) has been a promising tool to tackle optimization problems for decision making and resource allocation [14]. According to DRL methods, the agents can learn knowledge comprehensively to solve problems by interacting with the environment. The authors in [27] proposed a heuristic Q-learning model to obtain an efficient caching strategy to minimize the total system latency. The Double DQN model is illustrated in [28] to perform offloading and channel allocation to maximize users' QoE. However, the mobility of vehicles is often ignored, and those optimization problems are solved by the value-based DRL method, which is difficult to handle the large-scale data and continuous action spaces.

Different from existing studies, we design a novel intent-based networking traffic control system, which constructs a hierarchical architecture in IoCVs to orchestrate network traffic and allocate limited resources dynamically. Meanwhile, we formulate the optimization problem to maximize the profits for MNOs. Since the formulated optimization problem contains plenty of continuous variables and frequent handovers between servers and vehicles, we design a novel Actor-Critic based DRL model to solve the problem effectively.

III. System Model

The hierarchical edge computing and caching system, including moving vehicles, one base station and several RSUs equipped with intelligent edge servers, can provide communication, computing and caching capabilities for vehicles. In the constructed system, the MNO deploys an AI-based offloading and caching scheme, and allocates resources in a centralized manner when vehicles intend to perform various applications. We consider applications from vehicles consist of two components, i.e., computing tasks and content requesting tasks [29]. These kinds of applications are widespread in traffic control among 5G-envisioned IoCVs. For instance, applications from self-driving vehicles upload the observed road conditions to RSUs and then RSUs send driving directions as computing results to vehicles to avoid collision. Meanwhile, vehicles download local road-maps and real-time traffic conditions to determine optimal driving routes, which can avoid traffic congestion in smart cities. Notice that applications from vehicles are computing-insensitive and delay-sensitive. We assume that the BS, whose wireless communication range is large enough to cover all vehicles, owns a large amount of computing and storage capabilities. Therefore, the BS and RSUs can deal with users' requests together cooperatively. Based on the advanced wireless communication technologies in the 5G era, vehicles can simultaneously download contents from BS/RSUs and upload computing tasks to them by employing the full-duplex radio. The back-haul links between RSUs and the BS are wired lines, which guarantee robust data transmissions. Moreover, we assume tasks generated from vehicles can be divided into arbitrary parts, so that they support partial offloading and parallel computing at different servers. The illustration of main notations in this paper is in Table I.

Vehicles move along the road in the coverage of BS and RSUs with a certain velocity, and time is divided into discrete time slots. MNO decides offloading ratio and content caching components, and intelligently allocates resources for each user to complete tasks in time. According to the current states of vehicles and task requiring information, traffic control center in the ITS designs corresponding polices for vehicles, and dynamically allocates resources to servers for task execution.

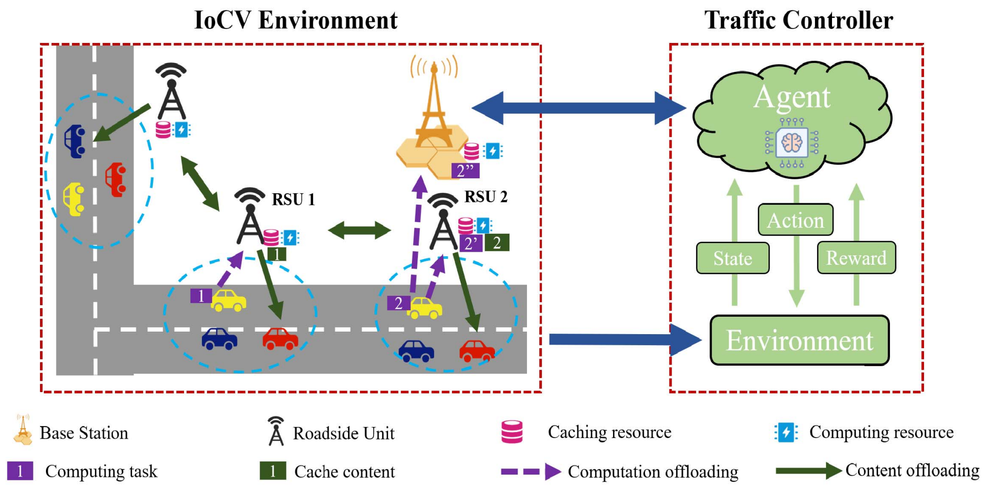


Fig. 1. Illustration of AI-based traffic control system with hierarchical architecture.

TABLE I
MAIN NOTATIONS

Notation	Definition
V	A set of users (vehicles)
M	A set of MEC servers
v_i	The velocity of user i
T_i	The time to cross the system of user i
s_i^{cp}	The size of computing tasks for user i
l_i^{cp}	The CPU cycles of computing tasks for user i
c_i	The size of requested contents for user i
p_i	The popularity of requested contents for user i
F_j	The total computational capability of MEC server j
G_j	The total caching capability of MEC server j
B_j	The total bandwidth of MEC server j
$C_i^{cp}(t)$	The size of tasks that user i needs to compute in timeslot t
$f_{i,j}(t)$	The computational resources of MEC server j allocated to user i in timeslot t
$D_i^{ca}(t)$	The contents of user i downloading in timeslot t
$g_{i,j}(t)$	The caching resources of MEC server j allocated to user i in timeslot t
$b_{i,j}(t)$	The total bandwidth of MEC server j allocated to user i in timeslot t
$SNR_{i,j}^t$	The SNR value for the link between server j and user i in timeslot t

The constructed AI-based traffic control system is illustrated in Fig. 1. The left part of the figure is the IoCV, and the right part is the AI-based traffic controller deployed on the BS. In addition, the traffic controller consists of environment information collected from IoCVs and AI-based agent. The former is the real-time information of IoCVs, such as vehicle speeds and users' demands. The latter is the AI-based processor, which designs sophisticated actions based on current states and rewards, including offloading and caching decisions for vehicles and resource allocation schemes for servers. For example, the yellow car in Fig. 1 moves from RSU1 to RSU2, content downloading and computation offloading change from RSU1 to RSU2 during the movement. When the available resources of the nearest RSU for the vehicle cannot satisfy the needs of users' requests, it will offload the task to the BS for cooperative execution. The resources allocated to RSUs and the BS are scheduled by traffic controller.

A. Communication Model

In our model, we consider each RSU has its own coverage area, indicating that a vehicle can establish data transmission links with only one RSU. Besides, let T_i^j be the duration time of vehicle i within the coverage of RSU j. Notice that $\sum_{j=1}^{M} T_i^j = T_i$, in which T_i represents the time that vehicle i crosses the system area. We consider that environment factors remain unchanged within T_i^j, and users associated with the MEC server equipped at the RSU can concurrently download the requested content and upload the computing task with no interference by adopting the Orthogonal Frequency Division Multiplexing (OFDM) Access technology. The bandwidth $b_{i,j}(t)$ that MEC server j allocates to vehicle i in timeslot t consists of two components, i.e.,

$$b_{i,j}(t) = b_{i,j}^{cp}(t) + b_{i,j}^{ca}(t), \tag{1}$$

where $b_{i,j}^{cp}(t)$ is the bandwidth used for compution task offloading, and $b_{i,j}^{ca}(t)$ is the consumed bandwidth for requested content downloading. When user i needs to upload or download data, it sends request signals to MEC server j, then the server assigns OFDM based links between them. Following the Shannon theory, the data transmission rate for the link from user i to server j is denoted by:

$$R_{i,j}^{comm}(t) = b_{i,j}^{comm}(t) log_2(1 + SNR_{i,j}^t), \tag{2}$$

where $b_{i,j}^{comm}(t)$ is the allocated bandwidth for the link from user i to server j. Symbol $SNR_{i,j}(t)$ is the Signal Noise Ratio (SNR) value of the link between server j and user i in timeslot t.

B. Computation Model

If the computing task generated from user i needs to be handled on the MEC server j via computation offloading, which has computational capability F_j (CPU cycles per second), user i uploads information about its tasks to the traffic control center in ITS, and then the ITS decides offloading radio, requested content caching of vehicles and resource allocation from MEC servers to user i. Due to the large coverage of BS, user i can both offload computing tasks to its nearest RSU and the BS for task computation in a cooperative manner. We denote the size of computing tasks of user i in timeslot t as $C_i^{cp}(t)$, where a_i^t is the percentage of computing tasks executed at the nearest RSU, and $(1 - a_i^t)$ is the counterpart executed at the BS. Let l_i^{cp} denote the required CPU cycles of the computing task generated from user i, and s_i^{cp} represents the size of the task. The allocated computational resources from RSU j to user i is $f_{i,j}$ (the BS to user i is $f_{i,0}$), which can be expressed as:

$$f_{i,j}(t) = a_i^t \cdot \frac{C_i^{cp}(t) l_i^{cp}}{s_i^{cp}|t|},$$
$$f_{i,0}(t) = (1 - a_i^t) \cdot \frac{C_i^{cp}(t) l_i^{cp}}{s_i^{cp}|t|}. \tag{3}$$

The transmission time of results from MEC servers to users can be ignored because their sizes are relatively small. Moreover, multiple vehicles can concurrently communicate with the same server and share its computational resources by OFDM based links. In order to guarantee computation offloading, ITS needs to allocate enough bandwidths for transmitting computing tasks, which can be expressed as:

$$b_{i,j}^{cp}(t) = a_i^t \cdot \frac{C_i^{cp}(t)}{log_2(1 + SNR_{i,j}^{cp,t})|t|},$$
$$b_{i,0}^{cp}(t) = (1 - a_i^t) \cdot \frac{C_i^{cp}(t)}{log_2(1 + SNR_{i,0}^{cp,t})|t|}. \tag{4}$$

C. Caching Model

We consider the requested content of vehicle i consists of two components, which can be expressed by $\{c_i, p_i\}$. Here, c_i is the size of requested content and p_i is the popularity of the content. Moreover, we denote $D_i^{ca}(t)$ as the size of content that user i downloads in timeslot t. b_i^t is the ratio of content downloading from the nearest RSU, while $(1 - b_i^t)$ is that downloading from the BS. After receiving the request for content from vehicle i, the control system decides the ratio of content downloading from MEC server j equipped with the RSU associated with the vehicle (i.e., b_i^t) and the counterpart from the BS (i.e., $(1 - b_i^t)$). The caching storage of MEC server allocated to user i in timeslot t for downloading content by RSU j (i.e., $g_{i,j}(t)$) and the BS (i.e., $g_{i,0}(t)$) can be expressed as:

$$g_{i,j}(t) = b_i^t \cdot D_i^{ca}(t),$$
$$g_{i,0}(t) = (1 - b_i^t) \cdot D_i^{ca}(t). \tag{5}$$

Then, the bandwidths for downloading requested contents by RSUs and BS can be calculated by:

$$b_{i,j}^{ca}(t) = \frac{g_{i,j}(t)}{log_2(1 + SNR_{i,j}^{ca,t})|t|},$$
$$b_{i,0}^{ca}(t) = \frac{g_{i,0}(t)}{log_2(1 + SNR_{i,0}^{ca,t})|t|}. \tag{6}$$

We assume that the requested content can be recovered if the vehicle receives all segments from MEC servers. Due to the finite caching capability of MEC servers located at RSUs, it is not possible to cache all requested contents in a single RSU.

IV. Problem Formulation

In this section, we define a profit function to calculate the total profits gained by the MNO, in which we jointly consider the benefits of MNO's service together with the QoE performance of vehicles. Then we formulate an optimization problem, which jointly considers task assignments and resource allocation to improve system performance (i.e., profits of the MNO) in IoCVs.

A. Profit Function

By considering the resource consumption on computing and caching, we formulate the total profits of the MNO consisting of four components, i.e., revenues, computation costs, caching costs and the QoE performance penalty. Specifically, the revenue for MNO in timeslot t can be denoted by:

$$R_{rev}(t) = \sum_{i=1}^{V} (\sum_{j=0}^{M} \alpha \frac{f_{i,j}(t) s_i^{cp}}{l_i^{cp}} |t| + \beta p_i g_{i,j}(t)), \tag{7}$$

where α and β are the charges for computing tasks and requested contents, respectively. In addition, the caching cost in timeslot t can be expressed as:

$$C_{ca}(t) = \sum_{i=1}^{V} [\sum_{j=1}^{M} (\delta_R b_{i,j}^{ca}(t) + \nu_R SNR_{i,j}^{ca,t} + \varphi_R g_{i,j}(t))$$
$$+ \delta_0 b_{i,0}^{ca}(t) + \nu_0 SNR_{i,0}^{ca,t} + \varphi_0 g_{i,0}(t)], \tag{8}$$

where δ_R and δ_0 are the communication costs consumed by vehicle downloading data from the RSU and the BS, respectively. In addition, ν_R and ν_0 are the channel costs for vehicles to access the virtual network via RSUs and the BS, respectively. Symbols φ_R and φ_0 are the costs for caching content at RSUs and the BS, respectively.

The total computation cost consists of two components, i.e., communication and CPU computing costs. The former is similar to the process of requested contents, and the latter is $\eta_R l_i^{cp} \omega_R + \eta_0 l_i^{cp} \omega_0$ (if the system completes all the computing tasks of user i). Symbol η is the cost for consuming energy, and ω is the energy consumption of running one CPU cycle. Then, the total computation cost can be calculated by:

$$C_{cp}(t) = \sum_{i=1}^{V} [\sum_{j=1}^{M} (\delta_R b_{i,j}^{cp}(t) + \nu_R SNR_{i,j}^{cp,t} + \eta_R f_{i,j}(t) \omega_R)$$
$$+ \delta_0 b_{i,0}^{cp}(t) + \nu_0 SNR_{i,0}^{cp,t} + \eta_0 f_{i,0}(t) \omega_0]. \tag{9}$$

In addition, we also take the QoE performance of users into consideration. We consider that if the control system along with ITS completes users' applications before they leave the overlaid area of ITS, the QoE performance of vehicular users is good enough without penalty. Otherwise, the system will cause certain losses to vehicular users as a result of poor QoE performance with penalty. Thus, we define QoE performance penalty function $\sigma(i, T_i)$ for vehicle i, which achieves a trade-off between the QoE performance of vehicular users and the profit of MNO. The sum of supplied caching resources from MEC servers to vehicle i can be expressed as $c_i^{T_i} = \sum_{t=1}^{T_i} D_i^{ca}(t)$. The total size of computing tasks from vehicle i computed by MEC servers is $s_i^{T_i} = \sum_{t=1}^{T_i} C_i^{cp}(t)$. Then we can express penalty function $\sigma(i, T_i)$ as:

$$\sigma(i, T_i) = \sigma^{ca}(c_i - c_i^{T_i}) + \sigma^{cp}(s_i^{cp} - s_i^{T_i}), \tag{10}$$

where σ^{ca} and σ^{cp} are penalty coefficients for content downloading and task computing, respectively. If the current scheme has not accomplished tasks from vehicle i until it moves away, i.e., $c_i^{T_i} < c_i$ or $s_i^{T_i} < s_i^{cp}$, the value of penalty function $\sigma(i, T_i)$ for user i is negative. Otherwise, it equals to 0. Hence, the total profits for the MNO is:

$$P_{MNO} = \sum_{t=1}^{T}(R_{rev}(t) - C_{ca}(t) - C_{cp}(t)) + \sum_{i=1}^{V}\sigma(i, T_i). \tag{11}$$

The above profit function is comprehensive, because the MNO charges customers (vehicular users) according to their different requested content sizes and computing resources. Meanwhile, the MNO needs to pay for utilized resources, such as network, electric power and channel resources. Moreover, the QoE performance is an important metric to attract vehicular users to produce benefits for the MNO.

B. Optimization Objective

By jointly considering task assignment and resource allocation in IoCVs, we formulate an optimization problem aiming to maximize the total profits of MNO, i.e.,

$$\max_{f_{i,j}(t), g_{i,j}(t), b_{i,j}(t)} P_{MNO} \tag{12a}$$

$$\text{s.t.} \quad \sum_{i=1}^{V} b_{i,j} \leqslant B_j, \quad \forall j \in M, \tag{12b}$$

$$\sum_{i=1}^{V} f_{i,j} \leqslant F_j, \quad \forall j \in M, \tag{12c}$$

$$\sum_{i=1}^{V} g_{i,j} \leqslant G_j, \quad \forall j \in M, \tag{12d}$$

$$g_{i,j}, b_{i,j}, f_{i,j} \geqslant 0, \quad \forall i \in V, \ \forall j \in M, \tag{12e}$$

$$c_i^{T_i} \leqslant c_i, \quad \forall i \in V, \tag{12f}$$

$$s_i^{T_i} \leqslant s_i^{cp}, \quad \forall i \in V. \tag{12g}$$

Constraints (12b), (12c) and (12d) guarantee the sum of allocated bandwidths, computing sources and caching storage for tasks not exceed the total capability of MEC servers, respectively. Constraint (12e) ensures that bandwidth, computing and caching resources allocated from servers to vehicular users are all equal or greater than 0. Constraint (12f) ensures the sum of supplied caching resources from MEC servers cannot exceed the users' requirements. Similarly, constraint (12g) means sums of computing resources from MEC servers are in the range of users' requirements.

V. DRL-Based Traffic Control System for Edge Computing and Caching

Convex optimization [5], [9] and game theory [30], [31] are two widely used approaches for traffic control systems to solve edge computing and caching problems. However, there are some deficiencies about these methods:

- Most of these methods are assumed to know key factors in IoCVs, such as channel conditions and content popularity. However, they are time-varying and unavailable in reality.
- The mobility of vehicles results in complicated and dynamic topology of IoCVs. Thus, it is challenging for these methods to guarantee reliable and efficient data transmissions.
- Most of these methods can merely achieve optimal or near optimal results for one snapshot of IoCVs. However, they ignore how the current decision exerts long-term influence on resource allocation.

Compared with these methods, DRL methods utilize deep neural network to interact with environment and allocate resources efficiently. The traffic control system faces plenty of continuous variables and frequent handovers between servers and vehicles, however, traditional value-based DRL methods (e.g., Q-learning and DQN), are not strong enough in processing the problem with large state space and continuous actions. Therefore, we design a novel Actor-Critic based DRL model to solve the problem in this section.

A. Traffic Control System Model

In our proposed traffic control scheme, the control center in ITS first collects environmental information, including the capabilities of MEC servers (i.e., RSUs and the BS), the task information (i.e., parameters for computing tasks and requested content), and vehicular mobility information (i.e., location and velocity). Specifically, vehicle's location is an important feature to determine which RSU in the IoCV to execute the corresponding tasks. The velocity is employed to calculate the duration time of constructed links between vehicles and RSUs. After collecting the status from MEC servers and vehicles, the system utilizes the DDPG-based algorithm to learn these information for scheme designing, which guides vehicles to offload computing tasks to MEC servers and download caching contents from them. At the same time, the agent determines actions to allocate communication, computing and caching resources for diverse requirements. Finally, MEC servers and vehicles execute the designed scheme in different time slots to accomplish tasks. In our system, three key elements of DRL can be expressed as follows:

1) States: The state in our system reflects the situation of IoCVs, including the status of each vehicle i ($i \in V$) and available resources of each MEC server j ($j \in M$). The state in timeslot t is denoted as:

$$s_t = \left\{D_i(t), F_j(t), G_j(t), B_j(t)\right\}. \tag{13}$$

The vehicular status set $D_i(t)$ includes vehicle's location, velocity, the total size of computing tasks and requested contents, the popularity of requested contents, the size of residual contents, remained computing tasks and required computation resources, and required CPU cycles for the task computing. In addition, $F_j(t)$, $G_j(t)$ and $B_j(t)$ denote the available resources of computation, caching and bandwidth of each MEC server, respectively.

2) Actions: In the constructed system, the agent decides how many computing tasks and requested contents are executed at different MEC servers and how many resources should be allocated to vehicles. The system receives different requests from vehicles and dispatches resources of MEC servers, so that vehicles can download contents from and upload computing tasks to different MEC servers. Denote the action in timeslot t by:

$$a_t = \left\{f_{i,j}(t), g_{i,j}(t), b_{i,j}(t)\right\}, \tag{14}$$

where $f_{i,j}(t)$, $g_{i,j}(t)$ and $b_{i,j}(t)$ represent the amount of computational resource, caching resource and bandwidth that MEC server j allocates to vehicle i, respectively. By considering the mobility of vehicles and diverse requirements of various applications, $f_{i,j}(t), g_{i,j}(t)$ and $b_{i,j}(t)$ are all continuous values, which guarantees accurate resource allocations by the time-varying control system.

3) Rewards: Based on the state and action in timeslot t, the agent receives reward R. Meanwhile, the reward should correspond with the objective function. Hence, we regard profit function $P(t) = R_{rev}(t) - C_{ca}(t) - C_{cp}(t)$ as the reward function for DRL in timeslot t. Particularly, the agent checks task accomplishments in the timeslot when vehicles leave the controlling area of ITS to decide QoE performance penalty for vehicles.

B. DDPG-Based Traffic Control Scheme

Since the state space is formulated by much dynamic environmental information and the action space contains many continuous values, we propose a DDPG-based method to maximize the reward function. In contrast to the traditional DRL, the DDPG-based method combines DNN and the Actor-Critic structure to evaluate and select actions, which can accelerate the convergence rate and achieve accurate estimations. The framework of DDPG-based method is shown in Fig. 2, consisting of an actor network, a critic network and a replay memory. Besides, each network is constructed by two DNNs, i.e., an online network to select actions and a target network to evaluate actions.

In the DDPG-based method, the environment includes a BS, some RSUs and several vehicles. The agent is placed at the BS with sufficient computing capabilities, which is responsible for designing optimal actions and sending them to servers and vehicles. After receiving designed actions, MEC servers and vehicles perform corresponding computing and caching schemes to allocate network resources. Finally, the agent calculate the MNO profit as the reward of designed scheme. The detailed processes of DDPG-based edge computing and caching scheduling are introduced as follows.

First, the agent collects environmental information, including the status of MEC servers and vehicles to formulate state tuples. The actor network selects an action a_t by substituting

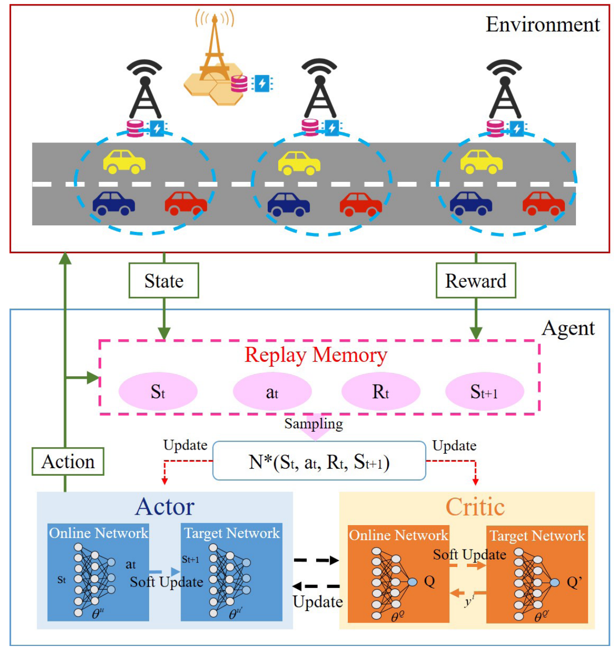


Fig. 2. The framework of the DDPG-based method for vehicular edge caching and computing.

current state s_t into the behaviour policy β, i.e.,

$$a_t = \beta(s_t) = \mu(s_t, \theta^{\mu}) + N_t, \tag{15}$$

where μ is the current online policy simulated by a Convolutional Neural Network (CNN) and N_t is the stochastic noise. For vehicular environment, vehicles carry out the computing offloading and edge caching scheme based on action a_t, and then MEC servers allocate corresponding resources to vehicles. The status of each entity will turn into s_{t+1} and return an immediate reward R for the agent.

After that, the actor network stores transition tuples (s_t, a_t, R_t, s_{t+1}) into the replay memory as training data sets for the online network. The critic network calculates Q-value to give a score for the online policy μ. According to the Bellman equation, the Q-value can be calculated by:

$$Q^{\mu}(s_t, a_t) = E[r(s_t, a_t) + \gamma\, Q^{\mu}(s_{t+1}, \mu(s_{t+1}, \theta^{Q}))]. \tag{16}$$

Specifically, the critic-online network constructs a CNN network to simulate Bellman equation to solve the recursive Q function and randomly samples a mini-batch transition tuples from the replay memory to update the parameter of the critic-online network. The critic-target network is responsible to calculate the following target Q-value for training the critic-online network:

$$y_t = r_t + \gamma\, Q'(s_{t+1}, \mu'(s_{t+1}, \theta^{\mu'}), \theta^{Q'}), \tag{17}$$

which combines the actor-target network with the critic-target network to stabilize the learning process and accelerate the convergence rate. After obtaining y_i, the critic-target network transmits it to the critic-online network with the objective of minimizing its loss function $L(\theta^{Q})$, i.e.,

$$L(\theta^{Q}) = \frac{1}{N}\sum_{t}^{N}(y_t - Q(s_t, a_t, \theta^{Q}))^2. \tag{18}$$

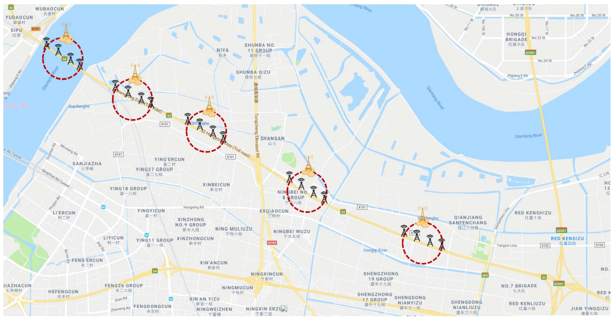


Fig. 3. Illustration of data selection on the S2 Hangyong Expy in Hangzhou.

Based on result θ^Q and the sampling transition tuples from relay memory, the actor-online network updates the behavior policy using the following policy gradient:

$$\begin{aligned}\nabla_{\theta^\mu} J(\mu) &= \int_S \rho^\beta(s) \nabla_a Q(s,a,\theta^Q) \cdot \nabla_{\theta^\mu} \mu(s,\theta^\mu) ds \\ &= E_{s\sim\rho^\beta}[\nabla_a Q(s,a,\theta^Q)|_{a=\mu(s)} \cdot \nabla_{\theta^\mu} \mu(s,\theta^\mu)],\end{aligned} \tag{19}$$

where $\rho^\beta(s)$ is the probability distribution function of states s based on behaviour policy β. Based on the Monte-Carlo method, we input a mini-batch size of data into Eq. (19) to estimate the policy gradient as follows:

$$\begin{aligned}\nabla_{\theta^\mu} J(\mu) = \frac{1}{N} \sum_t^N (\nabla_a Q(s,a,\theta^Q)|_{s=s_t,a=\mu(s_t)} \\ \times \nabla_{\theta^\mu} \mu(s,\theta^\mu)|_{s=s_t}).\end{aligned} \tag{20}$$

Finally, we utilize the soft updating method to partially update parameters of target networks by online networks, which can be formulated as follows:

$$\begin{aligned}\theta^{Q'} &\leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} &\leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}\end{aligned} \tag{21}$$

where τ is the updating coefficient. The objective function can be denoted as a long-term reward in our DDPG-based method. When the long-term reward converges to a stable value, we consider the optimal solution of the problem is trained. The pseudo-code of our DDPG-based algorithm is illustrated in Algorithm 1.

Algorithm 1 DDPG-Based Edge Computing and Caching Algorithm

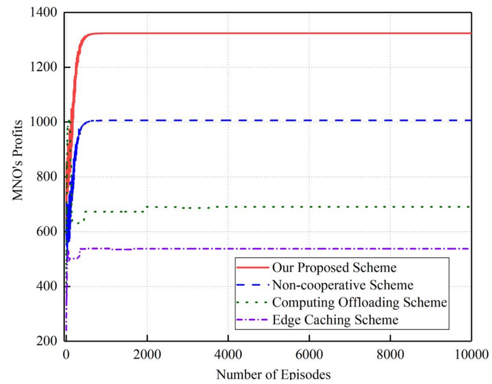
```
Initialize the weights of critic-online network and actor-online network as θ^Q and θ^μ
Initialize critic-target network Q' and actor-target network μ' with weights θ^{Q'} ← θ^Q, θ^{μ'} ← θ^μ
Initialize the replay memory K = ∅
for episode = 1, 2, ..., M do
  Initial observation state s_0, and P_MNO = 0
  Random choose N_t for action exploration
  for t = 1, T do
    The controlling system receives observation state s_t
    Select action a_t = μ(s_t, θ^μ) + N_t
    Calculate immediate reward R_t by s_t and a_t, and obtain the new state s_t --a_t--> s_{t+1}
    P_MNO ← P_MNO + R_t
    Store transition tuples (s_t, a_t, R_t, s_{t+1}) in K
    Randomly sample a mini-batch of N transition tuples from K
    Set target y_t = r_t + γ Q'(s_{t+1}, μ'(s_{t+1}, θ^{μ'}), θ^{Q'})
    Update the critic network by minimizing the loss according to (18)
    Update the behavior policy according to (20)
    Update target networks according to (21)
  end for
  Obtain executed task size s_i^T and downloaded content size c_i^T for each user from s_T
  P_MNO ← P_MNO + Σ_{i=1}^V σ(i, T_i)
end for
```

VI. Performance Evaluation

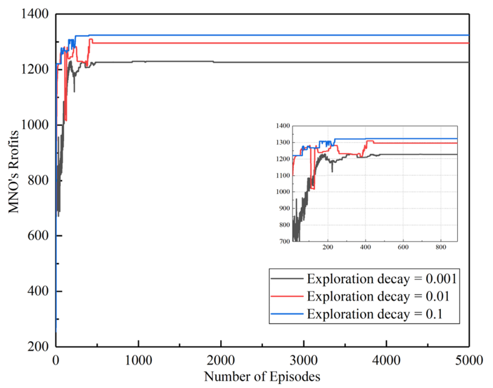
In our constructed DDPG-based control system for edge computing and caching, we consider a BS covers the whole ITS operating area and four RSUs cover non-overlapping areas with 250×250 m^2, respectively. We select 5 areas on the S2 Hangyong Expy in Hangzhou, China, to analyze the number and velocity of vehicles, which is shown in Fig. 3. The centers of these areas are (30.2701,120.2382), (30.2626,120.2527), (30.2545,120.2727), (30.2433,120.2951) and (30.2340,120.3183), respectively. According to the traffic data in September, 2017, the average speed of vehicles is about 30km/h. Therefore, the sojourn time of vehicles in ITS is about 2 min (calculated by system controlling area, i.e., 250m $\times$ 4 = 1km, and the average speed of vehicles, i.e., 30km/h). The traffic flow on the road is about 20 in every 2 minutes during the evening peak (from 5p.m. to 7p.m.). We test the system performance in the condition that 20 vehicles drive on the road at the average speed of 30km/h.

The sizes of computing tasks and requested contents are randomly distributed in $[0.15, 0.25, 0.3, 0.4, 0.45, 0.6]$ GB. The required CPU cycles of computing tasks are randomly selected from $[0.5, 0.6, 0.7, 0.8, 0.9, 1.2]$ Gcycles. The popularity of required content is distributed in $[1, 6]$ by a uniform manner. The computational capability, caching storage and total bandwidth for the BS are set as 10 GHz, 10 GB and 20 MHz, respectively. The counterparts for each RSU are set as 5 GHz, 5 GHz and 10 MH, respectively. The MNO's charges for task computing and content downloading are $\alpha = 200$ units/GB and $\alpha = 100$ units/GB, respectively. The costs for different resources are: $\delta_0 = 10$, $\delta_R = 20$ units/MHz, $\nu_0 = 4$, $\nu_R = 20$ units/dB, $\varphi_0 = 2$, $\varphi_R = 10$ units/GB, $\eta_0 = 3$, $\eta_R = 30$ units/W, and $\omega_0 = 2$, $\omega_R = 2$ W/GHz, respectively. The penalty coefficients for two types of tasks are $\sigma_{cp} = -0.5$, $\alpha = -100$ units/GB and $\sigma_{ca} = -0.5$, $\beta = -50$ units/GB, respectively. Our experiments are implemented by Python 3.5 and TensorFlow 1.12.0. Three schemes are compared with our proposed algorithm:

- Non-cooperative scheme: Cooperative computing and caching is not considered in this scheme, i.e., each vehicle



(a) Convergence performance for different schemes.



(b) Convergence performance with different exploration decay values.

Fig. 4. The convergence performance of our proposed scheme.

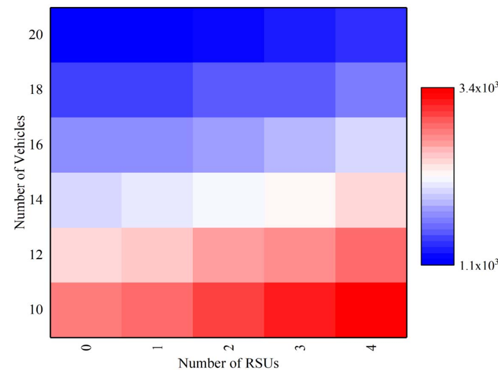


Fig. 5. MNO's profits with different numbers of vehicles and RSUs.

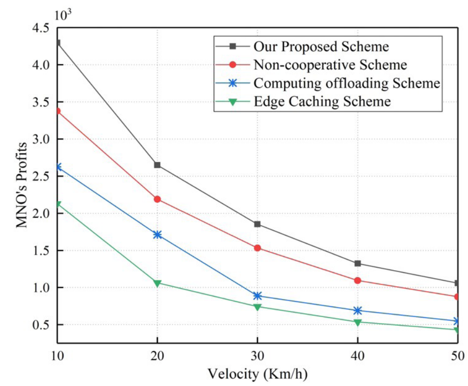


Fig. 6. Effects of vehicular velocity.

can only execute their tasks or download contents from one MEC server (either BS or RSU).

- Computing offloading scheme: This scheme serves vehicles without edge caching, i.e., vehicles download requested contents from the BS.
- Edge caching scheme: Offloading is not considered, i.e., vehicles only execute their computing tasks at the BS.

The convergence performance of our proposed scheme is illustrated in Fig. 4. We can see from Fig. 4(a) that our proposed scheme outperforms other representative methods in terms of the MNO's profits and convergent speed. We observe that the MNO's profits of our proposed method are obviously higher than those of other methods. This is because our DDPG-based method considers the traffic control scheme of the BS and RSUs cooperatively, by which the utilization efficiency of network resources can be increased and the benefits for the MNO can be largely promoted. Moreover, Fig. 4(a) shows that values of MNO profits of different methods are low at the beginning of the training process. As the number of episodes increases, the MNO profits of different methods increase rapidly, and reach stable values after about 1,000 episodes. The convergence of the MNO profits implies that the agent in our traffic control system can learn better resource allocation policies than others.

Fig. 4(b) shows the convergence performance with different exploration decays of our proposed DDPG-based algorithm. Exploration decay represents the chance to decay the action randomness, which increases the probability to perform the action produced by the actor network. The initial exploration probability is set as a large value, and then decays with different rates to balance the exploration and exploitation in the selected actions. It can be observed that a larger exploration decay results in good convergence performance in terms of MNO's profits. Meanwhile, a larger exploration decay can accelerate the convergence of our algorithm. Thus, we set the exploration decay value as 0.1 in the rest of simulations.

Fig. 5 illustrates the MNO's profits with different numbers of vehicles and RSUs. We increase the number of RSUs from $M = 0$ to $M = 4$ and increase the number of running vehicles from $V = 10$ to $V = 20$. Specifically, $M = 0$ represents there is only a BS to serve vehicles in the system. The MNO gets less profits with the increase of vehicles. Since more vehicles tend to share the limited resources, it leads to the MNO can hardly satisfy the strict deadline of vehicles' requests. Thus, the QoE performance penalty gets larger, inducing the decline of MNO's profits. Meanwhile, the MNO's profits will increase as more RSUs are deployed to provide more bandwidth, computing and caching resources for vehicles.

We illustrate the effects of velocity on MNO's profits for different schemes in Fig. 6. The vehicle with high speed implies that its duration time in the system is short, so that the system needs to allocate more resources to satisfy its requests. Thus, we can observe from that the MNO's profits of different

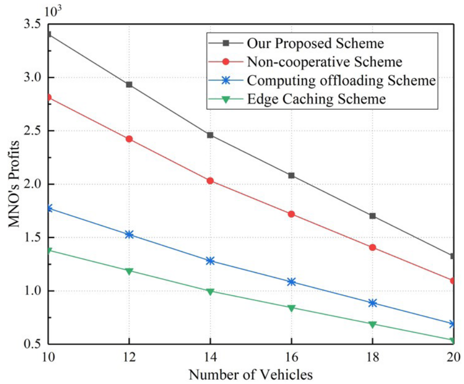


Fig. 7. Effects of vehicle number.

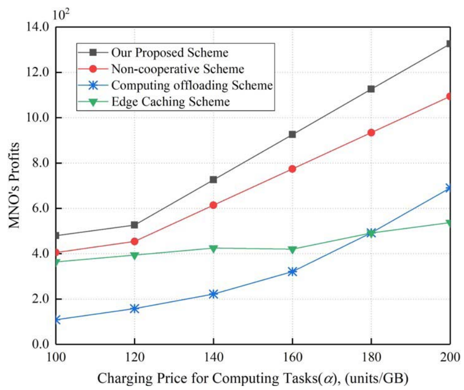


Fig. 8. Effects of computing charging price.

schemes decrease with the increase of velocity. When vehicles move at a relatively low speed (i.e., 10 or 20 km/h), the system has abundant resources to execute tasks, leading to a high value of the MNO's profits. However, with the increase of velocity, the system has limited resources to meet the deadline of tasks, so that MNO's profits become relatively low.

The effect of vehicle number passing by on the MNO's profits is shown in Fig. 7. The decline of MNO's profits is approximately linear with the increasing number of vehicles. This is because more vehicles share the computing, caching and bandwidth resources with the increasing number of vehicles. Especially, when we do not consider computing offloading in the edge caching scheme, the single server cannot handle the computing tasks timely, leading to low QoE performance of users.

Fig. 8 illustrates the MNO's profits under different computing charge prices for different schemes. We can see that our proposed scheme can obtain high profits with the increase of charging prices for computing tasks. This is because the incoming profits of computing offloading increase when the charging price rises, and agent prefers to perform computing offloading to execute users' computing-intensive tasks. Meanwhile, we can see that edge caching scheme without computing offloading almost keeps unchanged with different charging prices.

Fig. 9 shows the MNO's profits under various caching charging prices. It can be seen that MNO's profit of our

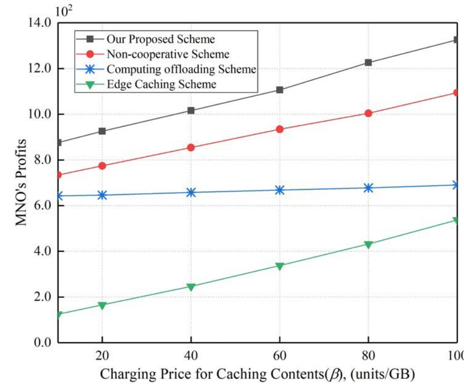


Fig. 9. Effects of caching charging price.

scheme outperforms others as the increase of charging prices for caching. The trends of non-cooperative and edge caching schemes grow parallelly because they are similarly affected by the increase of caching prices. Besides, they pay much attention to edge caching placement issues for content requiring tasks, which brings much benefits for the agent. The performance of computing offloading scheme remains unchanged with the increase of charging prices for caching contents.

VII. Conclusion

We construct an AI-based traffic control system for 5G-envisioned IoCVs, and put forward a hierarchical edge computing and caching model, which can process various vehicular applications with limited resources to guarantee efficient traffic controlling. After that, we formulate the traffic control scheme and resource allocation strategy as a joint optimization problem to maximize profits of the MNO. The defined profit function of the MNO not only analyzes its revenue, but also takes users' QoE performance into consideration. Regarding the time-varying task assignments and resource allocation strategies are continuous values, we design the DDPG-based scheme to solve the optimization problem. Numerical results based on real traffic data in Hangzhou, China, demonstrate that our proposed scheme achieves satisfied performance in IoCVs, and brings great benefits for the MNO. We intend to utilize AI-based algorithms to make proactive caching of the requested contents and pre-allocate network bandwidth in the future work.

References

[1] Z. Ning, X. Wang, J. J. P. C. Rodrigues, and F. Xia, "Joint computation offloading, power allocation, and channel assignment for 5G-enabled traffic management systems," *IEEE Trans. Ind. Inf.*, vol. 15, no. 5, pp. 3058–3067, May 2019.

[2] Z. Ning, J. Huang, and X. Wang, "Vehicular fog computing: Enabling real-time traffic management for smart cities," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 87–93, Feb. 2019.

[3] S. Garg, K. Kaur, N. Kumar, and J. J. P. C. Rodrigues, "Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 566–578, Mar. 2019.

[4] E. E. Ugwuanyi, S. Ghosh, M. Iqbal, T. Dagiuklas, S. Mumtaz, and A. Al-Dulaimi, "Co-operative and hybrid replacement caching for multi-access mobile edge computing," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2019, pp. 394–399.

[5] Z. Ning, X. Wang, and J. Huang, "Mobile edge computing-enabled 5g vehicular networks: Toward the integration of communication and computing," *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 54–61, Mar. 2019.

[6] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. P. C. Rodrigues, and M. Guizani, "Edge computing in the industrial Internet of Things environment: Software-defined-networks-based edge-cloud interplay," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 44–51, Feb. 2018.

[7] X. Wang *et al.*, "Optimizing content dissemination for real-time traffic management in large-scale Internet of Vehicle systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1093–1105, Feb. 2019.

[8] Z. Ning *et al.*, "When deep reinforcement learning meets 5G-enabled vehicular networks: A distributed offloading framework for traffic big data," *IEEE Trans. Ind. Inf.*, vol. 16, no. 2, pp. 1352–1361, Feb. 2020.

[9] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 4177–4190, Jun. 2018.

[10] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[11] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4804–4814, Jun. 2019.

[12] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2923–2960, Jun. 2018.

[13] D. Silver *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, Dec. 2018.

[14] Z. Li, C. Wang, and C.-J. Jiang, "User association for load balancing in vehicular networks: An online reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 2217–2228, Aug. 2017.

[15] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[16] Z. Ning *et al.*, "Deep learning in edge of vehicles: Exploring trirelationship for data transmission," *IEEE Trans. Ind. Inf.*, vol. 15, no. 10, pp. 5737–5746, Oct. 2019.

[17] S. Garg *et al.*, "Edge computing-based security framework for big data analytics in VANETs," *IEEE Netw.*, vol. 33, no. 2, pp. 72–81, Mar. 2019.

[18] Z. Ning *et al.*, "Deep reinforcement learning for intelligent Internet of Vehicles: An energy-efficient computational offloading scheme," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1060–1072, Dec. 2019.

[19] Z. Zhao, L. Guardalben, M. Karimzadeh, J. Silva, T. Braun, and S. Sargento, "Mobility prediction-assisted over-the-top edge prefetching for hierarchical VANETs," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1786–1801, Aug. 2018.

[20] J. Zhang *et al.*, "Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4283–4294, Jun. 2019.

[21] Z. Ning, J. Huang, X. Wang, J. J. P. C. Rodrigues, and L. Guo, "Mobile edge computing-enabled Internet of Vehicles: Toward energy-efficient scheduling," *IEEE Netw.*, vol. 33, no. 5, pp. 198–205, Sep. 2019.

[22] F. Jameel, M. A. Javed, and D. T. Ngo, "Performance analysis of cooperative V2V and V2I communications under correlated fading," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[23] X. Hu, H. Wang, and X. Tang, "Cyber-physical control for energy-saving vehicle following with connectivity," *IEEE Trans. Ind. Electron.*, vol. 64, no. 11, pp. 8578–8587, Nov. 2017.

[24] B. S. Khan, S. Jangsher, H. K. Qureshi, and S. Mumtaz, "Energy efficient caching in cooperative small cell network," in *Proc. 16th IEEE CCNC*, Jan. 2019, pp. 1–6.

[25] Z. Zhou, F. Xiong, C. Xu, Y. He, and S. Mumtaz, "Energy-efficient vehicular heterogeneous networks for green cities," *IEEE Trans. Ind. Inf.*, vol. 14, no. 4, pp. 1522–1531, Apr. 2018.

[26] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, Jun. 2018.

[27] L. Hou, L. Lei, K. Zheng, and X. Wang, "A *Q*-learning-based proactive caching strategy for non-safety related services in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4512–4520, Nov. 2019.

[28] Z. Ning, P. Dong, X. Wang, J. Rodrigues, and F. Xia, "Deep reinforcement learning for vehicular edge computing: An intelligent offloading system," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 6, pp. 1–24, 2019.

[29] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.

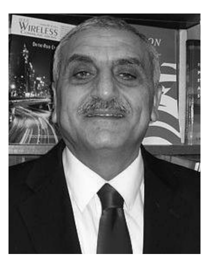
[30] G. S. Aujla, R. Chaudhary, N. Kumar, J. J. P. C. Rodrigues, and A. Vinel, "Data offloading in 5G-enabled software-defined vehicular networks: A Stackelberg-game-based approach," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 100–108, Aug. 2017.

[31] S. Josilo and G. Dan, "Selfish decentralized computation offloading for mobile cloud computing in dense wireless networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 207–220, Jan. 2019.

Zhaolong Ning (Senior Member, IEEE) received the M.S. and Ph.D. degrees from Northeastern University, Shenyang, China. He was a Research Fellow with Kyushu University, Japan. He is currently an Associate Professor with the Dalian University of Technology, China, an Adjunct Professor with the Chongqing University of Posts and Telecommunications, China, and a Visiting Professor with the National Mobile Communications Research Laboratory, Southeast University, China. He has published over 100 scientific papers in international journals and conferences. His research interests include the Internet of vehicles, mobile edge computing, and artificial intelligence.

Kaiyuan Zhang received the B.S. degree from the Dalian University of Technology, Dalian, China, in 2017, where he is currently pursuing the M.S. degree. His research interests include mobile edge computing, artificial intelligence, and network caching.

Xiaojie Wang received the M.S. degree from Northeastern University, China, in 2011, and the Ph.D. degree from the Dalian University of Technology, Dalian, China, in 2019. From 2011 to 2015, she was a Software Engineer with NeuSoft Corporation, China. She holds a postdoctoral position at The Hong Kong Polytechnic University. Her research interests are wireless networks, mobile edge computing, and machine learning.



Mohammad S. Obaidat (Life Fellow, IEEE) received the Ph.D. degree in computer engineering with a minor in computer science from The Ohio State University, Columbus, USA. He is currently the Founding Dean of the College in Computing and Informatics, The University of Sharjah, UAE, and an Adjunct Professor with The University of Jordan, Jordan, and the University of Science and Technology Beijing. He has published, to date (2020), about 1,000 refereed technical articles; about half of them are journal articles, over 70 books, and about 70 book chapters. He is a fellow of SCS. He received many best paper awards for his articles including ones from IEEE ICC, IEEE Globecom, AICSA, CITS, SPECTS, and DCNET International conferences. He also received best paper awards from the IEEE SYSTEMS JOURNAL in 2018 and in 2019 (two Best Paper Awards). He also received many other worldwide awards for his technical contributions. He also received the SCS Outstanding Service Award and the IEEE CITS Hall of Fame Distinguished and Eminent Award. He has chaired numerous (over 160) international conferences and has given numerous (over 160) keynote speeches worldwide. He founded or co-founded four international conferences. He has served as ABET/CSAB evaluator and on IEEE CS Fellow Evaluation Committee. He has served as an IEEE CS Distinguished Speaker/Lecturer and an ACM Distinguished Lecturer. Since 2004, he has been serving as a SCS Distinguished Lecturer. He is the Editor-in-Chief of three scholarly journals and an editor of many other international journals. He is the founding Editor-in Chief of Wiley *Security and Privacy* journal.

Lei Guo received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2006. He is currently a Full Professor with the Chongqing University of Posts and Telecommunications, Chongqing, China. He has authored or coauthored more than 200 technical papers in international journals and conferences. His current research interests include communication networks, optical communications, and wireless communications. He is an editor for several international journals.

Xiping Hu received the Ph.D. degree in electrical and computer engineering from The University of British Colombia, Vancouver, Canada. He is currently a Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China. He was the Co-Founder and CTO of Bravolol Limited, Hong Kong, a leading language learning mobile application company with over 100 million users, and listed as top 2 language education platform globally. He has around 50 articles published and presented in prestigious conferences and journals, such as the IEEE TETC/TVT/TII/IoT journal, ACM TOMM, IEEE COMST, *IEEE Communications Magazine, IEEE Network*, HICSS, ACM MobiCom, and ACM WWW. His research areas consist of distributed intelligent systems, crowdsensing, social networks, and cloud computing.

Bin Hu (Senior Member, IEEE) is currently a Professor with Lanzhou University and a Guest Professor with ETH Zurich, Switzerland. He has published more than 100 articles in peer-reviewed journals, conferences, and book chapters, including *Science, Journal of Alzheimer's Disease, PLoS Computational Biology*, IEEE Transactions, the IEEE INTELLIGENT SYSTEMS, and AAAI. He is an IET Fellow, Co-Chair of IEEE SMC TC on Cognitive Computing, a Member at Large of ACM China, and the Vice President of International Society for Social Neuroscience (China committee). He has served as the Chair/Co-Chair in many IEEE international conferences/workshops. He has served as an Associate Editor in peer-reviewed journals on cognitive science and pervasive computing, such as the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, *Brain Informatics*, and *IET Communications*.

Yi Guo received the Medical degree and M.Med. degree from the Tongji Medical College, Huazhong University of Science and Technology, Wuhan, China, in 1985 and 1991, respectively, and the Ph.D. degree from the University of Greifswald, Greifswald, Germany, in 1997. He is currently the Chief of neurology with the Second Clinical Medical College, Jinan University, a member of the Cerebrovascular Disease Group, Chinese Medical Association neurology branch, and the Chairman of the Shenzhen Medical Association of neurology and the Shenzhen Medical Association of psychosomatic medicine. His major research areas are cerebrovascular diseases, dementia, movement disorder diseases, sleep disorder, and depression and anxiety.

Balqies Sadoun received the M.S. degree in civil engineering from the Ecole Nationale Des Travaux Publique de LEtat, Lyon, France, and the M.S. and Ph.D. degrees from The Ohio State University, Columbus, Ohio, USA, where show was on a scholarship by JUST. She served as a Department Chair more than once. She worked at Yarmouk University and Jordan University of Science and Technology, Jordan. She is currently a Professor with the Department of Surveying Engineering and Geomatics and Civil Engineering, Faculty of Engineering, Al-Balqa Applied University, Jordan. She has published good number of refereed journals and conference papers. Her research interests include modeling and simulation, geographical information systems (GIS), wireless navigation systems, data analytics, transportation systems, and smart homes and cities, among others.

Ricky Y. K. Kwok (Fellow, IEEE) received the B.Sc. degree in computer engineering from The University of Hong Kong in 1991 and the M.Phil. and Ph.D. degrees in computer science from The Hong Kong University of Science and Technology (HKUST) in 1994 and 1997, respectively. His research focus has been on designing efficient communication protocols and robust resources management algorithms toward enabling large scale distributed mobile computing. In these research areas, he has authored one textbook, coauthored another two textbooks, and published more than 200 technical papers in various leading journals, research books, and refereed international conference proceedings. He is a fellow of the HKIE and the IET. From March 2006 to December 2011, he served on the Editorial Board of *Journal of Parallel and Distributed Computing* as a Subject Area Editor in peer-to-peer computing. He also served as an Associate Editor for the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS from January 2013 to December 2016.