Learning Based Channel Allocation and Task Offloading in Temporary UAV-Assisted Vehicular Edge Computing Networks

Chao Yang, Baichuan Liu, Haoyu Li, Bo Li, Kan Xie, and Shengli Xie, *Fellow, IEEE*

***Abstract*—High-level autonomous decision making system is one of the key technologies in intelligent transportation networks, it requires the traffic information within a certain range of vehicles in real time. When the traffic roads become congested or the roadside units (RSUs) are unaccessed beyond the communication range, the unmanned aerial vehicle (UAV)-assisted vehicular edge computing network (VECN) is considered as a potential solution. In this paper, we propose a learning based channel allocation and task offloading strategy in temporary UAV-assisted VECNs from a user perspective, in which the UAV passing temporarily can serve as the relay and edge computing node to support the decision making system. However, the limited available computation resources and time-varying communication channel states make it critical to process the received computing tasks. To address the above mentioned challenges, we design an efficient data transmission strategy combined the long-term evolution vehicle-to-everything (LTE-V2X) and time-division multiple access (TDMA) technologies firstly, then, we propose a multi-option task processing scheme, a service cost minimization problem is proposed where the integral decisions of channel allocation and task processing mode selection are jointly optimized. Under dynamic computing resources and the current data transmission conditions, the UAV selects an optimal task processing service model based on deep reinforcement learning (DRL) algorithm. Simulation results show the proposed strategy greatly improves the data transmission efficiency.**

***Index Terms*—DRL, task processing, UAV, vehicular edge computing networks.**

I. Introduction

IN RECENT years, high-level autonomous decision-making of an intelligent vehicle requires a variety of surrounding traffic information within a certain range of vehicles under strict delay constraints, to support the advanced smart vehicular services, e.g., advanced driver assistance systems and autonomous driving [1]. To this end, a requesting vehicle could assign tasks of information collection and data processing to not only the surrounding one-hop communication link vehicles, but also the vehicles in the incoming pass roads, especially in the traffic intersections. The long-term evolution vehicle-to-everything (LTE-V2X) technologies, including both the vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications are used with low service cost and delay [2], [3]. The requesting vehicle can perform data processing with the help of the appropriate proximal vehicles, which plays as volunteer vehicles to contribute their idle time and on-board resources. This scenario is facilitated with the emergence of vehicular edge computing networks (VECN), in which resources of edge servers deployed at roadside units (RSUs), parked and mobile vehicles are jointly utilized to augment the computing capacity of vehicular networks [4], [5].

However, a volunteer vehicle may be willing to gather the information but unwilling to undertake the computation task, due to the limited computing resources of a single vehicle or heavy internal workloads at that time. As an alternative, the volunteer vehicle transmits the unprocessed data to a nearest edge server. But the changeable typology of vehicular network makes V2V/V2I communication unstable. Actually, the traffic network is vulnerable, especially in the rush hour. For example, during the Chinese spring festival travel season, many suburb roads that are not busy on weekdays become extremely congested. Sudden traffic accidents will exacerbate such congestion. When the edge server is overloaded or unaccessed beyond the communication range, the offloading task cannot be received and completed within the delay requirement. To cope with the dilemma, researches have employed an unmanned aerial vehicle (UAV) as an optional candidate to receive offloading tasks

Manuscript received 8 January 2022; revised 17 March 2022; accepted 9 May 2022. Date of publication 24 May 2022; date of current version 19 September 2022. This work was supported in part by the National Natural Science Foundation of China under Grants 62003094, 61973087, and U1911401, in part by Guangdong Basic and Applied Basic Research Foundation under Grants 2019A1515011377, 2019A1515011114, and 202102020573, in part by the Research and Development Program of Key Science and Technology Fields in Guangzhou City under Grant 202206030005, and in part by the State Key Laboratory of Synthetical Automation for Process Industries under Grant 2020-KF-21-02. The review of this article was coordinated by Prof. Ju Ren. (*Corresponding author: Bo Li.*)

Chao Yang, Baichuan Liu, and Haoyu Li are with the School of Automation, Guangdong University of Technology, Guangzhou, Guangdong 510006, China, and also with the Guangdong Key Laboratory of IoT Information Technology, Guangzhou, Guangdong University of Technology, Guangzhou, Guangdong 510006, China (e-mail: yangchaoscut@aliyun.com; 2112004107@mail2.gdut.edu.cn; 3116001106@mail2.gdut.edu.cn).

Bo Li and Kan Xie are with the 111 Center for Intelligent Batch Manufacturing Based on IoT Technology, Guangdong University of Technology, Guangzhou, Guangdong 510006, China, and also with the Key Laboratory of Intelligent Detection and the Internet of Things in Manufacturing and Ministry of Education, Guangdong University of Technology, Guangzhou, Guangdong 510006, China (e-mail: libo999@gdut.edu.cn; kanxie@gdut.edu.cn).

Shengli Xie is with the School of Automation, Guangdong University of Technology, Guangzhou, Guangdong 510006, China, and with the Guangdong-HongKong-Macao Joint Laboratory for Smart Discrete Manufacturing, Guangdong University of Technology, Guangzhou, Guangdong 510006, China, and also with the Joint International Research Laboratory of Intelligent Information Processing and System Integration of IoT, Guangdong University of Technology, Guangzhou, Guangdong 510006, China (e-mail: shlxie@gdut.edu.cn).

Digital Object Identifier 10.1109/TVT.2022.3177664

from mobile vehicles and provide flexible offloading services in VECN, which leads to a new computing paradigm named by UAV-assisted VECN. The significant advantages of UAVs including lower cost, flexible deployment, and easy realization of line-of-sight (LoS) communications [6], [7], will promote the implementation of UAV-assisted VECN, particularly in the initial phase of VECN where the number of edge servers is strictly limited to avoid the tremendous economic expense.

Nowadays, the existing works on UAV-assisted VECN focus on the UAV flight trajectory optimization and the designing of task offloading scheme to tradeoff the task computation delay and the limited energy of UAVs. A UAV could perform as a relay node [8]–[10] or an edge computing node [11], [12]. For example, the UAV can be deployed above the vehicles to work as a relay to establish LoS communication links between uncovered vehicle users and the neighboring base station. In [11], the communication resource allocation and flight path planning of UAV are joint optimized to minimize the total energy consumption while satisfying quality of service of the road moving vehicles. In [12], the authors propose an edge computing model that uses the UAV swarm as relay nodes and jointly optimizes the task offloading strategy and the routing path planning to meet the user's delay requirements and computing requirements.

However, we note that the above works implicitly assume the UAVs are artificially deployed, which means that the flight trajectory, transmission power and other UAV key parameters can be optimized. In fact, in addition to those UAVs, there are many other temporary UAVs that pass around vehicles, especially in the suburban areas, such as target tracking [13], surveillance [14], etc. Normally, many onboard resources of those temporary UAVs are idle and can be further exploited and utilized without affecting UAVs' major services. Along with providing major certain specific services, the UAVs can provide offloading service in the edge computing environment [15], [16]. The dynamic behaviors of the temporary UAV make channel states between the UAV and the ground vehicles change continuously.

Motivated by above works, we consider that in a temporary UAV-assisted VECN, a set volunteer vehicles are assumed to collect the surrounding information and generate the computation tasks, which are submitted to a nearest UAV for processing. The UAV further feedbacks the output results to the requesting vehicle. However, there exist challenging issues that needs to be addressed for the success of the UAV-assisted VECN. First, both the UAV and the volunteer vehicles have dynamic mobility and the channel state information between them is time-varying. Second, a typical binary computation offloading scheme means that an offloading task of a volunteer vehicle is simply rejected by the UAV or received with the fine-grained processing where the UAV is required to undertake the total workload of the task. In some cases, the task could also accept the non-fine-grained processing where this greatly causes a moderate number of the workloads to the UAV. For example, the complete technical process of face recognition is usually consisted of four steps [17], face detection, image pre-processing, feature extraction and feature matching. When the computing tasks arrive, the UAV with different available computation resources can decide which step to process the task. Last but not least, the requesting vehicle clearly expects more volunteer vehicles to support the collaborative information collection. Whereas, with the number of the offloading tasks increasing, appropriate solution methodologies are required to effectively improve the efficiency of solving the large-scale optimization problems.

To tackle the problems, we study the joint optimization of channel allocation and task offloading in UAV-assisted VEC by using deep reinforcement learning (DRL). A suitable UAV is pre-selected to serve a requesting vehicle and a set of volunteer vehicles. We consider that the volunteer vehicles transmit the data to the UAV via the LET-V2X, and after the calculation, the UAV transmits the according output results to the requesting vehicle via the time division multiple access (TDMA). Particularly, we design a multi-option task processing scheme. When receiving an offloading task, the UAV have three modes of processing it, namely, direct data delivery, non-fine-grained data processing and fine-grained data processing. Different modes lead to different workloads to the UAV. To present a user-oriented scheme for the requesting vehicle, we formulate a service cost minimization problem where the integral decisions of channel allocation and mode selection should be jointly optimized to reduce the monetary payment for the UAV in the data transmission and processing procedure. Considering the dynamic communication environment of the UAV, real-time arrivals of the tasks, and non-convex problem with coupling constraints, we further apply the decomposition and DRL approaches to quickly seek a near-optimal solution. The whole problem is decomposed into two subproblems: channel allocation problem and mode selection problem, which are tackled by the conventional and learning based methods respectively.

- We introduce the implementation temporary UAV-assisted VECN according to an application scenario. Communication technologies are exploited to support the data transmission between the requesting vehicle, volunteer vehicles and UAV.
- We formulate the joint optimization problem to optimize the UAV-assisted offloading procedure from a user perspective. In the data transmission and processing, dynamic channel allocation and flexible mode selection are elaborately considered.
- We present the learning based solution methodology to tackle the complex problem. The combination of current optimization methods and a Deep Q-network (DQN) algorithm is proposed to find an approximate solution of the whole problem after decomposing it into two subproblems.

In addition, we conduct a set of simulations to show that the proposed task processing strategy can help reduce the task completed time and cost. The rest of the paper is organized as follows. First, we discuss the related work. Then, we describe the system model of temporary UAV-assisted VECN in the Section III. In the Section IV, we elaborate the solution algorithm based on deep reinforcement learning for the proposed problem. In the Section V, we give the setting of simulation parameters, and discuss the simulation results. Finally, in the Section VI, the conclusion of this work is proposed. In order to make this paper

TABLE I
NOTATIONS USED IN THIS PAPER

Notations	Description
K	Number of sub-channels.
r_{uav}, $r_{vehicle}$	Communication radius of UAV and vehicle respectively.
$\boldsymbol{I}$	A set of volunteer vehicles' index.
m_i	Number of sub-channels occupied by the volunteer vehicles i.
S_i	Data size of task i.
T_i^g	Time duration that the relevant data of task i has been fully prepared.
$\boldsymbol{O}_i$, $\boldsymbol{C}_i$	Data sizes of output after data processing; requirement of computing resource for data processing.
$\boldsymbol{\alpha}$	A tuple $\langle\alpha_0, \alpha_1, \alpha_2\rangle$, which denote the data processing model seleted by UAV.
$\mathcal{C}_u$	Available computing resource provided by UAV.
C_T, C_P	Prices of data transmitting and computation resources.
ξ_u, ξ_p, ξ_d	Progress degree of uploading, processing, downloading.
l_u, l_c	Location of UAV and requiring vehicle.
c_{down}, t_r, $path$	Channel capacity of downlink; real-time delay requirement of task; planned flight trajectory.
ζ	A sign, to indicates whether the data collection is successful or not.
D_0	Initial airline distance between temporary UAV and requiring vehicles.
V_{uav}	Velocity of temporary UAV.

more readable, the main parameters and descriptions are defined as shown in Table I.

II. RELATED WORK

Currently, VECN mitigates the delay sensitive application requirements of both vehicles and passengers in 5G network, via sinking computation and caching resources to the network edge nodes, which has attracted attention in both the industry and academia areas [4], [18]–[23]. In [18], the communication radio control schemes under time division duplex (TDD) in vehicular network are proposed. In [4], the computation resource load balance among the edge nodes in RSUs is analyzed in a fiber-wireless (FiWi) enhanced VECN. In [19], the author consider the RSUs are powered by solar, and a real-time task offloading strategy is proposed. The edge computing server can perform computing resource sharing with cloud computing [23] and other edge nodes [22]. In [24], a cloud-edge-terminal collaborative vehicular edge computing network is proposed, and the authors proposed an efficient task offloading scheme. Besides the edge nodes deployed in RSUs, the parked vehicles in parking lot [20], the vehicle platoon on road [21] are also enabled to support the computation resource supplement in VECN.

Moreover, with the advance of flight control technologies, UAV-assisted VECN has also be studied to improve the system performance of vehicular networks, especially when the traffic roads are lack of the RSUs or the communication links are interrupted. The UAVs can provide communication, computation and caching resources for the ground moving vehicles. Specially, the deployment, flight trajectory [9], [10], communication and computation resource allocation of UAVs [25]–[29] are mainly considered. In [9], UAVs are deployed between terrestrial base stations and vehicles to work as relays, a unified energy management framework is proposed and the end-to-end outage probability is minimized. In [10], UAVs are set up between ground mobile devices and base stations. The flight trajectory and signal transmission power of the UAVs are optimized to minimize the network interruption probability. As a type of separately added mobile device, UAVs can be integrated with different types of ground networks. In [30], the UAV-aided network with a millimeter wave (mmWave) backhaul is considered, along with the ground ad hoc network, the task completed latency minimization problem is studied. In addition, the resource scheduling of UAV-assisted edge computing system with the wireless power transmission-enabled vehicle platoon is analyzed in [28]. In [29], the authors formulate a joint multi-objective optimization problem for the UAV-assisted VECN, while the edge node selection, task offloading and resource allocation are considered.

Different from the above mentioned studies, a multi-option task processing strategy in a temporary UAV-assisted VECN is proposed in this paper. The temporary UAV is used to support the delay requirements of VECN, especially in the scenario that the UAV is moving from the data source vehicles to the objective one. After receiving the tasks from the volunteer vehicles, the UAV can select different task processing mode under the available computation resources and communication links. The combination of current optimization methods and a Deep Q-network (DQN) algorithm is proposed to optimize the channel allocation and the task processing mode selection.

III. SYSTEM MODEL AND PROBLEM FORMULATION

The temporary UAV-assisted VECN includes two main entities: vehicles and UAVs. As shown in Fig. 1, several vehicles are driving on roads, and a temporary UAV passes through the area to cover the ground vehicles. The vehicles are divided into requiring vehicles and volunteer ones, and they can convert mutually in different time slots and spaces. We consider a requiring vehicle constantly collect the surrounding traffic information within the help of a set of volunteer vehicles. The requiring vehicle will select an appropriate data transmission link based on location to connect these vehicles. If the distances between the requiring vehicle and the volunteer vehicles are within one hop communication range, the V2V communication is adopted to download data directly. Otherwise it needs to collect data through RSU or UAV as a relay node. In this paper, we focus on the latter one, the index of the volunteer vehicles is $\boldsymbol{I} = \{1, \ldots, i, \ldots, I\}$. We set that each vehicle has one task in a

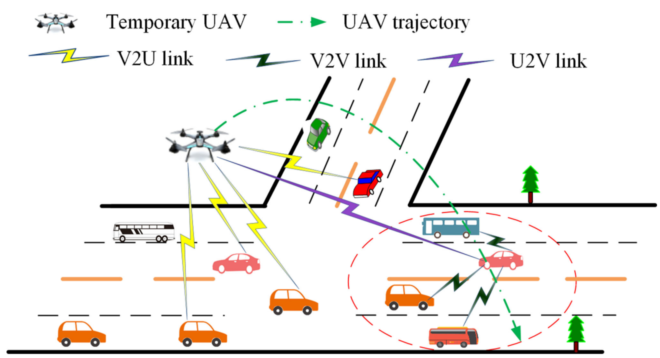


Fig. 1. Temporary UAV-assisted VECN model.

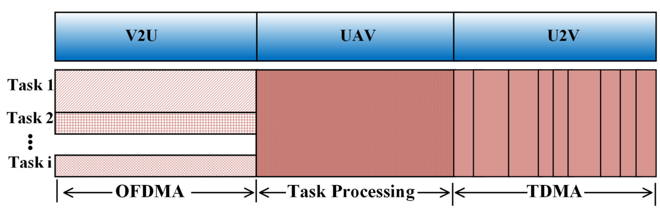


Fig. 2. Data transmission model of temporary UAV-assisted VECN.

slot, i also denotes the index of task. In the scenario considered in this paper, there is no good/available RSU service environment where the requiring vehicle is located. For example, when the roads become congested, the communication and computation loads of the RSU exceed the predetermined value. Moreover, there are obstacles affect the communication V2I links. Therefore, when the requiring vehicle needs service such as relay or data processing, the temporary UAV can be used.

Fig. 2 illustrates the data transmission model of temporary UAV-assisted VECN. The UAV performs as both the relay and edge computing node. We design a data transmission strategy combined with the LTE-V2X and TDMA. Firstly, the transmission data from the volunteer vehicles is uploaded to the UAV via the vehicle-to-UAV (V2U) link. The LTE-V2X technology is used to support the task uploading from multiple volunteer vehicles to UAV [31]. An optimal communication resource (i.e., available channels) allocation is proposed, based on the orthogonal frequency division multiple access (OFDMA). Then, the UAV performs task processing. For the limited computation resources of UAV, it processes the received tasks sequentially. The output will be downloaded to the requiring vehicle via UAV-to-vehicle (U2V) link, and TDMA technology is used here. Considering that providing a flexible computing scheme allows UAV to have more choices to balance its service cost and the requirement of adapting to the dynamic and changeable environment, we propose that UAV has three models of task processing services.

- **Model 1: Direct data delivery**. It means the UAV forward the original received data package directly without processing and waiting, but the size of the output data package remains large, same as the origin data.
- **Model 2: Non-fine-grained data processing**. It means that the UAV performs the task processing with appropriate computing resources, such as data de-noising. This process requires a moderate time and the data size of outputs is smaller than origin data.
- **Model 3: Fine-grained data processing**. It means that the origin received task data will be processed in a deep extent, after fine-grained data processing, the extracted results will be sent to the requiring vehicle. The process of this model takes a long time, but the data size of outputs is extremely small.

It should be noted that the processing time and the size of output data are related, but not a simple linear relationship.

A. Communication and Computation Delay

1) Volunteer Vehicles to UAV: Denote u and 0 as the index of UAV and requiring vehicle. The temporary UAV is moving and its flight path will not be changed. Considering the channel model of LoS link between volunteer vehicle i and UAV satisfies the free space path loss model [32], the channel gain in time t, $h_{ui}(t)$ is expressed as

$$h_{ui}(t) = \beta_0 d_{ui}^{-2}(t), \tag{1}$$

where β_0 denotes the unit channel power when reference distance $d_0 = 1\,m$, and d_{ui} denotes the airline distance between UAV and the volunteer vehicles. The available channel of V2U is divided into K subchannels, and the bandwidths of each sub-channel are equal to B_0. For the different volunteer vehicles, they should select different number of subchannels to transmit, to meet the task uploading delay requirements. We use r_{uav} and $r_{vehicle}$ to represent the communication radius of UAV and vehicle respectively. When the requiring vehicle drives out of the UAV coverage area, i.e., $d_{0,u} > r_{uav}$, the service ends. Denote m_i as the occupied number of subchannels, $\sum_{i=1}^{I} m_i = K$. According to Shannon's theorem, the signal-to-noise ratio (SNR) received by UAV at time t is shown as

$$\gamma_{u,i}(t) = \frac{p_{i,u} h_{u,i}(t)}{\sum I_{i,'u} + \sigma^2}, \tag{2}$$

where $i' \in \boldsymbol{I}$, $i' \neq i$, $p_{i,u}$ is the transmission power of volunteer vehicle. We set that the transmission powers are fixed and equally, for the energy supplement of vehicles is enough, and the vehicles can select the maximum power. $\sum I_{i,'u} = \sum p_{i,'u} h_{i,'u}$ represents the interference from other vehicles transmit task at the same channel simultaneously, σ^2 denotes the white Gaussian noise. Then, the channel capacity of uploading V2U link from volunteer vehicle i to UAV at time t can be expressed as

$$R_{u,i}(t) = m_i B_0 \log_2(1 + \gamma_{u,i}(t)), \tag{3}$$

where $m_i B_0$ is the channel bandwidth of vehicle i, which should be optimized to perform communication resource allocation. Denote S_i as the data size of task from volunteer vehicle i, set the uploading start time and end time are t_{start} and t_{end}, we have

$$S_i = \int_{t_{start}}^{t_{end}} R_{u,i}(t)\, dt. \tag{4}$$

Denote T_i^g as the time duration that the relevant data of volunteer vehicle i has been fully prepared, the task i uploading time is shown as

$$T_{upload,i} = T_i^g + t_{end} - t_{start}. \tag{5}$$

2) UAV Task Computation: Denote the received data package from volunteer vehicle i as

$$Pkg_i = \{S_i, T_{req,i}, \boldsymbol{O}_i, \boldsymbol{C}_i\}, \tag{6}$$

where S_i is the data size of input package, $T_{req,i}$ is the delay requirement. The UAV can select different task processing models, the data sizes of output and the corresponding computing resources are different. $\boldsymbol{O}_i$ represents the data sizes of output produced by data processing at UAV, $\boldsymbol{O}_i = \{O_{0,i}, O_{1,i}, O_{2,i}\} Mb$. $\boldsymbol{C}_i$ represents the requirement of computing resource for data processing, $\boldsymbol{C}_i = \{C_{0,i}, C_{1,i}, C_{2,i}\} G$. Specifically, we have

$$\{\boldsymbol{O}_i, \boldsymbol{C}_i\} = \begin{cases} \{O_{0,i}, C_{0,i}\}, & \text{Model 1}, \\ \{O_{1,i}, C_{1,i}\}, & \text{Model 2}, \\ \{O_{2,i}, C_{2,i}\}, & \text{Model 3}. \end{cases} \tag{7}$$

The data package of task processing is bound by $T_{req,i}$, if the transmission fails to complete before the delay requirement expires, the task will be discard as a invalid data. We use a tuple $\boldsymbol{\alpha} = \langle \alpha_0, \alpha_1, \alpha_2 \rangle$ to denote the data processing model selected by UAV, where $\alpha_0, \alpha_1, \alpha_2 \in \{0, 1\}$, and $\alpha_0 + \alpha_1 + \alpha_2 = 1$. Therefore, the data processing time consumption of task i can be expressed as

$$T_{process,i} = \frac{\sum_{j=1}^{2} \alpha_j C_{j,i}}{\mathcal{C}_u(t)}, \tag{8}$$

where $\mathcal{C}_u$(t) is the available computing resource provided by UAV at time t.

3) UAV to Requiring Vehicle: For the UAV makes the task processing sequentially, the outputs are transmitted to the requiring vehicle via TDMA technology. For the tasks always be interdependent, we set that the UAV begins the download while all of the received tasks had been processed. All of the available channels are used for data transmission, we set the data download time duration of the task i output as $T_{download,i}$.

Therefore, the total time consumption of task i, $T_{total,i}$ can be calculated as

$$T_{total,i} = T_{upload,i} + T_{process,i} + T_{download,i}.$$

And the total time consumption of all of the tasks T_{total} is

$$T_{total} = max\{T_{upload,i}\} + \sum_{i \in \boldsymbol{I}} T_{process,i} + \sum_{i \in \boldsymbol{I}} T_{download,i}. \tag{9}$$

When the volunteer vehicles are within the one hop transmission range of the requiring vehicle, the requiring vehicle will directly download data from the data source node by V2V communication. The V2V and V2U communications can be done at the same time. Thus, the data collection time consumption of these volunteer vehicles is not be calculated in this work. Moreover, considering that all vehicles on the road need to collect data, the relationship between the requiring vehicle and volunteer vehicles is a mutually beneficial relationship. Therefore, the data sharing among the requiring vehicle and volunteer vehicles via one-hop V2V communication will not cause cost.

B. Cost

When the temporary UAV is used as the relay and edge node to collect data, the requiring vehicle needs to pay for data transmission and computation service. We denote the price of data transmitting as C_T per Mb, and the price of computation resources of CPU cycles is C_P per G. Therefore, when the task i has been collected by the requiring vehicle successfully, the generated system cost is expressed as

$$cost_i = C_T \left(S_i + \sum_{j=0}^{2} \alpha_j O_{j,i} \right) + C_P \sum_{j=0}^{2} \alpha_j C_{j,i}. \tag{10}$$

However, it should be noted that the system cost is accumulated in the whole progress of three stages: data preparation and transmission, calculation, and downloading. If the requiring vehicle does not collect a data package from the volunteer vehicles successfully, it will also incur costs. No matter what task processing step of the data collection is in, it is necessary to ensure the task completed delay requirements. When the requirement is no longer satisfied before the data downloading step (U2V) is completed, the data packet will be abandoned, the previous process of the data package will become invalid, and the expenses incurred by the invalid process will still be recorded in the total system cost. In order to formulate a accumulative cost, ξ_u, ξ_p and ξ_d are used for denoting the completion degree of mentioned three stages respectively. Take an example to further illustrate the completion degree, if the task i is being processed in UAV, but the delay requirement expires, the ξ_u is set as 1 for the completion of the task uploading, ξ_p is set as 0.4 if 40% of the task processing completes (or 0.6 if 60% of the task processing completes), ξ_d is set as 0 because unfinished task processing produces nothing to download. Then, the cost can be re-expressed as

$$cost_i = C_T \left(\xi_u S_i + \xi_d \sum_{j=0}^{2} \alpha_j O_{j,i} \right) + C_P \xi_p \sum_{j=0}^{2} \alpha_j C_{j,i}, \tag{11}$$

where $\xi_u, \xi_p, \xi_d \in [0, 1]$. When $\xi_u = 1$, it means that the volunteer vehicle has prepared the collection sensing data. Otherwise, when $\xi_u < 1$, it means that the volunteer vehicle hasn't collect the sensing data before the delay threshold, and this task cannot be calculated in UAV, we have $\xi_p = 0$ and $\xi_d = 0$. In addition, when $\xi_p < 1$, we have $\xi_d = 0$. In summary, the relationship between ξ_u, ξ_p and ξ_d is as follows:

$$\begin{cases} \xi_p = 0, & if\ \xi_u < 1 \\ 0 < \xi_p \leqslant 1, & otherwise, \end{cases}$$

$$\begin{cases} \xi_d = 0, & if\ \xi_p < 1 \\ 0 < \xi_d \leqslant 1, & otherwise. \end{cases}$$

C. Problem Formulation

Suppose that in the proposed temporary UAV-assisted VECN, the requiring vehicle has initiated the collection of I data packages. Our goal is to collect the expected data packets on the premise of meeting the delay requirements of each data packet and minimizing the cost. Both the communication resource allocation and computation resource allocation are mainly considered. The volunteer vehicles should select suitable number of subchannels (m_i), and the UAV should select an optimal task processing model (α_j). The joint optimization problem **P1** can be expressed as

$$\min_{\{m_i,\alpha_j\}} \sum_{i\in \boldsymbol{I}} cost_i \tag{12}$$

subject to:

$$T_{total,i} \leq T_{req,i}, \quad i \in \boldsymbol{I}, \tag{13}$$

$$T_{total} \leq T_{req}^{th}, \tag{14}$$

$$\sum_{i\in \boldsymbol{I}} m_i = K, \tag{15}$$

$$m_i \in \{1, 2, \ldots, K\}, i \in \boldsymbol{I}. \tag{16}$$

where T_{req}^{th} is the delay requirement threshold of all of the tasks. In order to make the optimization problem solvable, we set $K > I$. Each vehicle can obtain one subchannel at least. For the objective function of problem **P1**, we can find that a tradeoff between the task processing and transmission. If we want to reduce the cost of data transmission (i.e., reduce the size of output $O_{j,i}$, the UAV should select the high data processing model, much more computation resources are need, and the cost of $C_{j,i}$ increases). Moreover, for the volunteer vehicles need different time duration to prepare the collection data, and the locations are changeable, they should select suitable number of subchannels to obtain the minimal data uploading delay. Problem **P1** is a complex integer programming problem, it is hard to be solved directly. To tackle this, the primal problem is decomposed into two subproblems, and be solved with Lagrange duality method and DQN-based algorithm.

IV. Multi-Option Task Processing Strategy

As shown in Fig. 1, the whole data collection includes three steps, as V2U data uploading, UAV task processing and U2V data downloading. We analyze the channel selection in step 1, and the task processing model optimization of UAV in step 2. Thus, we propose two subproblems to solve problem **P1**.

A. Communication Resource Allocation

For LTE-V2X is used to support the V2U communication, multiple volunteer vehicles upload the collection sensing data to the UAV simultaneously via OFDMA technology, the number of allocated subchannels for vehicles m_i should be optimized to minimize the task uploading delay, which also affects the system cost directly. The subproblem **P1-1** is formulated as

$$\min_{\{m_i\}} max\{T_{upload,i}\}, \tag{17}$$

subject to:

$$\sum_{i\in \boldsymbol{I}} m_i = K,$$

$$m_i \in \{1, 2, \ldots, K\},$$

$$T_{upload,i} \leq T_{req,i}. \tag{18}$$

Problem **P1-1** is a typical multivariate integer programming problem. We can rewrite the variable m_i to continuous ones, as

$$1 \leq m_i \leq K. \tag{19}$$

P1-1 becomes an convex optimization problem. set

$$T'_{upload,i} = \max\{T_{upload,i}\}.$$

Then, the Lagrange function is shown as

$$L_i\left(m_i, \zeta_i^1, \zeta_i^2, \zeta_i^3\right) = T'_{upload,i} + \zeta_i^1\left(T'_{upload,i} - T_{req,i}\right) + \zeta_i^2\left(\sum_{i\in I} m_i - K\right) - \zeta_i^3(m_i - 1),$$

where ζ_i^1, ζ_i^2 and ζ_i^3 are the Lagrange multipliers. The main KKT conditions are given as

$$\frac{\partial L_i\left(m_i, \zeta_i^1, \zeta_i^2, \zeta_i^3\right)}{\partial m_i} = 0, \qquad \forall i \in I,$$

$$\zeta_i^1\left(T'_{upload,i} - T_{req,i}\right) = 0,$$

$$\zeta_i^2\left(\sum_{i\in I} m_i - K\right) = 0,$$

$$1 \leq m \leq K.$$

Then, the Lagrange duality method and Karush-Kuhn-Tucker (KKT) conditions are used to solve this problem, more details can be found in Refs. [10], [33].

B. Task Processing Strategy in UAV

1) Problem Transformation: When the data of the volunteer vehicles is uploaded to the UAV successfully, the UAV is required to make a decision about how to process the received data, to minimize the system service cost. The problem **P1** is reduced to the subproblem **P1-2**, as

$$\min_{\{\alpha_j\}} \sum_{i\in \boldsymbol{I}} cost_i, \tag{20}$$

subject to:

$$T_{total,i} \leq T_{req,i}, \quad i \in \boldsymbol{I},$$

$$T_{total} \leq T_{req}^{th}.$$

The DQN-based algorithm is used to solve this subproblem. According to the system state s_t at the time t, an appropriate action is taken and executed, which is considered to bring about successful collection of the target data and least cost. When the data packet collection ends, whatever successful or not, the environment will feed back a reward r_t to the UAV according to the system state s_{t+1} at the current time $t+1$. Therefore, with a completed UAV-assisted data processing scheme, we can obtain a sequence $e = (s_1, a_1, r_1, \ldots, s_n, a_n, r_n, s_{n+1})$. Note that the UAV service will become unavailable when the system state is s_{n+1}, which is denoted as a terminal state. Then, in order to minimize the total cost of data collection with a set of conditions, the subproblem can be transferred to find an optimal sequence e^*, it is a finite sequence. Therefore, this decision process is a typical finite Markov decision process (MDP), deep reinforcement learning (DRL)-based algorithm is a simple, convenient and efficient solution for this problem.

2) DRL Design in Task Processing Strategy: We define a tuple $\mathcal{M} = \langle \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\mathcal{P}}, \boldsymbol{\Pi} \rangle$ to denote MDP, a continuous decision process. Specifically, $\boldsymbol{\mathcal{S}}$ is the set of states, $\boldsymbol{\mathcal{A}}$ is the set of actions, $\boldsymbol{\mathcal{R}}$ denotes the rewards, $\boldsymbol{\mathcal{P}}$ denotes the state transition probability (STP), and $\boldsymbol{\Pi}$ means the policy of action selection. The special definitions of state, action, reward, and policy are elaborated as follow.

$States$: We use a tuple $s_t = \langle l_u, l_c, c_{down}, t_r, S, path \rangle$ to denote the system state at time t, where l_u and l_c respectively express the location of UAV and requiring vehicle, c_{down} is the estimated value of channel capacity of downlink from UAV to requiring vehicle, t_r means the delay requirement of data package, i.e., T_{req}. S is the size of origin data, and $path$ is the planned flight trajectory of the temporary UAV. Actually, these parameters are affected with each other. For example, the location of UAV and the channel capacity affect the data transmission delay directly.

$Actions$: We use $\boldsymbol{\mathcal{A}} = \{a_1, a_2, a_3\}$ to denote the three model of data processing when data arrives UAV, where a_1 represents model one, a_2 represents model two and a_3 represents model three.

$Reward$: A reward r_i will be given to requiring vehicle when the collection of data package i ends, and its value is determined by three main factors, include time and cost consumption, data collection successful or not. Then, it can be calculated as

$$r_i = \zeta \left(\frac{\lambda_1 S_i}{T^i_{total}} + \frac{\lambda_2 S_i}{cost_i} \right) \tag{21}$$

where $\zeta \in \{0, 1\}$, $\zeta = 1$ indicates the successful data collection, and $\zeta = 0$ indicates failure. λ_1 and λ_2 are both constant coefficients, and be correspond to the weight of these two parts in brackets respectively.

$Policy$: We use π to denote policy, which is the mapping relationship between states and actions in the system. It indicates the probability of requiring vehicle selecting action a_t when the system state is s_t. After the tasks arrive, the UAV will take an action a_t according to the policy π and system state s_t, to process the received data. Actually, the policy can be regarded as a guidance that which action should be selected in system state s_t. A general task of reinforcement learning (RL) is proposed to find out an optimal policy π_* by loop iteration. In order to evaluate and improve the π, the Q value function [18], [34], which indicates the expected accumulated reward of the starting

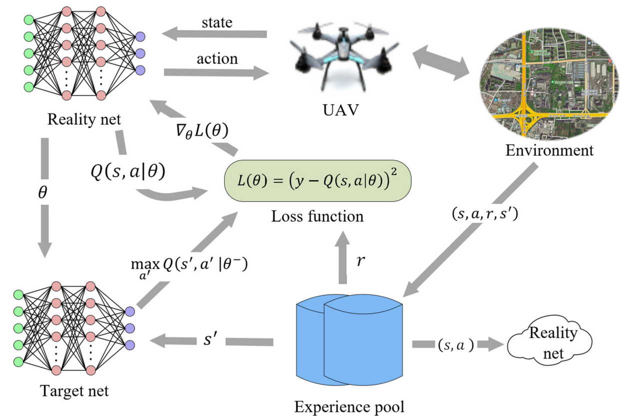


Fig. 3. DQN framework of the proposed scheme.

s_t with the following policy π and the implementation of a_t, is defined as

$$Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1} \in \boldsymbol{\mathcal{S}}, a_{t+1} \in \boldsymbol{\mathcal{A}}} \left[r_t + \eta Q_\pi(s_{t+1}, a_{t+1}) \middle| s_t, a_t \right] \tag{22}$$

where η denotes the discount factor. Following the optimal policy π_*, we can obtain the optimal Q value function. According to the Bellman equation [34], π_* can be expressed as

$$\pi_*(a|s) = \begin{cases} 1, \text{ if } a = \arg\max_{a \in \boldsymbol{\mathcal{A}}} Q_\pi(s,a) \\ 0, \text{ if } a \neq \arg\max_{a \in \boldsymbol{\mathcal{A}}} Q_\pi(s,a). \end{cases}$$

Then, the optimal Q value function can be expressed as

$$Q_*(s_t, a_t) = \mathbb{E}_{s_{t+1} \in \boldsymbol{\mathcal{S}}} \left[r_t + \eta \max_{a_{t+1} \in \boldsymbol{\mathcal{A}}} Q_*(s_{t+1}, a_{t+1}) \middle| s_t, a_t \right] \tag{23}$$

Fitting Q value function with neural network is a basic idea of DRL. DQN is a kind of DRL algorithm, which is an enhanced version of Q-learning. In Q-learning, Q-table is used to record Q value. However, two neural networks are used in DQN for fitting Q function. The reality network (RN) is used for evaluating current Q value, and the target network (TN) is used for predicting future Q value. To formulate the error between RN and TN, a loss function related to neural network parameter θ, is defined as

$$L(\theta) = (y - Q(s, a|\theta))^2, \tag{24}$$

where $y = r + \eta \max_{a' \in \boldsymbol{\mathcal{A}}} Q(s,' a'|\theta^-)$, and θ, θ^- are the RN and TN parameters respectively. Then, the gradient of $L(\theta)$ with respect to θ can be expressed as

$$\nabla_\theta L(\theta) = (y - Q(s, a|\theta)) \nabla_\theta Q(s, a|\theta) \tag{25}$$

The DQN framework is shown in Fig. 3. According to observed system state s, RN will calculate the corresponding Q value of all action, and then takes and executes the action a which corresponds to the maximum Q value, i.e., $a = \arg\max_{a \in \boldsymbol{\mathcal{A}}} Q(s, a|\theta)$. Under the influence of action a, the system state s will turn into the next state s', at the same time, a reward r is given. To obtain data for training neural network, the experience pool is set to store the produced (s, a, r, s'). When

Algorithm 1: DQN-Based Algorithm.

Initialize: ε, experience pool, θ, θ^-.
for episode $= 0$ to M
  for $l = 0$ to L **do**
    if $d_{i,0} > r_{vehicle}$ **then**
      Select a: choose an action randomly with probability ε, or take $\arg\max_{a\in\mathcal{A}} Q(s, a|\theta)$ with probability $1 - \varepsilon$.
      Perform a and wait for obtaining a reward r, then observe s'.
      Store (s, a, r, s') in experience pool.
      Randomly sample a batch of (s, a, r, s') from experience pool.
      Obtain y by $y = r + \eta \max_{a'\in\mathcal{A}} Q(s,' a'|\theta^-)$.
      Perform a gradient descend step on $L(\theta)$ with respect to θ.
      if $k/N_c = 0$ **then**
        Set $\theta^- = \theta$.
      end if
    end if
  end for
end for

the number of (s, a, r, s') record in experience pool reaches upper bound, several (s, a, r, s') record will be taken out from experience pool to train RN. Specifically, (s, a) is used for calculating $Q(s, a|\theta)$ via the operation of RN, and s' is used for calculating $\max_{a'\in\mathcal{A}} Q(s,' a'|\theta^-)$ via TN. Then, the gradient $\nabla_\theta L(\theta)$ can be calculated by Eq. (25), the gradient descend(GD) approach is used to update θ, as

$$\theta \leftarrow \theta - \alpha \nabla_\theta L(\theta), \tag{26}$$

where α is the learning rate. The value of θ will be assigned to θ^- when training times reaches an upper bound N_c. Meanwhile, with the continuous generation of (s, a, r, s'), the experience pool constantly updates. To avoid falling into local optimization, we apply the $\varepsilon - greedy$ policy. There is ε probability that the action is randomly selected, and $1 - \varepsilon$ probability that the action is $\arg\max_{a\in\mathcal{A}} Q(s, a|\theta)$. The proposed DQN-based algorithm for problem **P1-2** is shown as Algorithm 1.

In summary, the proposed efficient DQN-based Multi-option task processing Scheme in temporary UAV-assisted VECN (DMS) is shown as Algorithm 2.

Algorithm 2: DQN-Based Multi-Option Task Processing Scheme (DMS).

Step 1, the volunteer vehicles prepare the traffic information data, and sent it to the UAV via V2U communication,
The number of uploading subchannels $\{m_i\}$ is optimized via subproblem **P1-1**, the Lagrange duality method is used to solve it.
Step 2, the UAV performs task processing scheme after receiving the tasks from the volunteer vehicles,
The task processing models $\{\alpha_j\}$ is optimized via subproblem **P1-2**, a DQN-based algorithm is deigned to solve it.
Step 3, the UAV send the outputs to the requiring vehicle via TDMA technology.

TABLE II
DEFAULT PARAMETER SETUP

Parameter	Description	Value
β_0	Channel power in unit distance	$50dBm$
$P_{u,i}$	Each node's transmission power	$10dBm$
σ_2	White Gaussian noise	$-30dBm$
B_0	Bandwidth of each subchannel	$1M$
C_T	Price of data transmission	0.2
C_P	Price of data Processing	0.5
$\boldsymbol{O}_i$	Output size	$\{S_i, 0.7S_i, 0.01S_i\}$
$\boldsymbol{C}_i$	Computing resource requirements	$\{0, S_i, 2S_i\}$

V. SIMULATION

A. Parameter Setting

In this paper, we consider both the urban or suburban scenes. According to the regulations on urban traffic management, the speed limit range of urban roads is usually at the range of $30\,km/h \sim 60\,km/h$ (i.e., $8.33\,m/s \sim 16.67\,m/s$). Therefore, we commonly set the velocity of vehicles as $10 + random(0, 1) m/s$. In [35], author has studied the relationship between communication distance and communication success probability in V2V scenario, simulation results show that the success probability is near to 0.9 when communication distance is $5\,m$, or 0.1 when communication distance is $50\,m$. Therefore, we set the maximum communication distance of V2V one hop communication as $30\,m$ to ensure the basic availability and stability of V2V network. Similarly, we refer to [36] and set the maximum one hop communication distance of UAVs as $300\,m$, the flight height as $50\,m$. Referring to [15], the UAV edge server's computing capability is set from $1\,GHz \sim 3GHz(GC = 10^9 cycles, GHz = GC/s)$, the S_i is set as $random(2, 10)Mb$, and the task's computation-to-data ratios is set between $1GC/Mb \sim 2GC/Mb$. To simplify the task computing model, we set $\boldsymbol{C}_i$ as $\{0, Si, 2Si\}GC$. The number of volunteer vehicles on the road is set to $I = 30$, the locations are given randomly, and the requiring vehicle location is the beginning edge of the road. The delay requirement of data package enough $T_{req,i} \sim U(1, 5)s$, and $T_{req}^{th} = 100\,s$. In terms of software, we use python 3.8 and tensorflow 2.0 to build simulation model. The hidden layer of neural network is designed as $64 * 128 * 64$, learning rate is 0.01, and discount factor is 0.9. The initial ε is 0.9 and will constantly descend with a decline rate of 1.01. In the hardware part, the computer CPU model is AMD r3600. More details are shown in Table II.

B. Results Discussion

We evaluate the proposed DQN-based Multi-option task processing Scheme (DMS) from two aspects. Firstly, we evaluate the convergence performance and calculation time consumption of the proposed scheme, and compare it with other basic

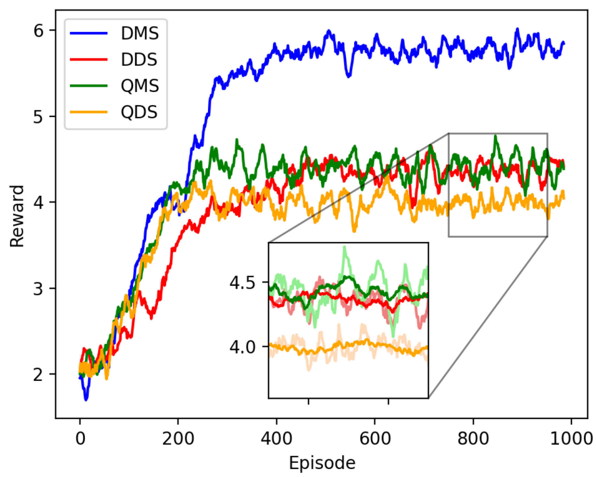


Fig. 4. Rewards of different schemes.

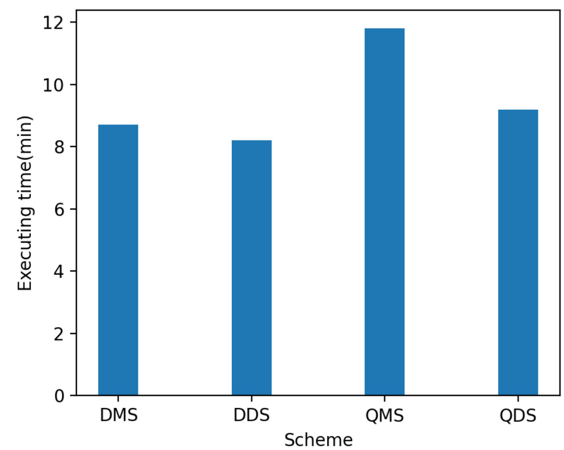


Fig. 5. Execution time of different schemes.

schemes. Secondly, we take the flight speed of UAV V_{uav}, the initial distance from UAV to requiring vehicle D_0, the computation capacity, the transmission power of UAV and the number of vehicles as the variables to evaluate the mean system service costs. The schemes used for comparison are as follows.

- **DQN-based Double-option Scheme** (DDS): It is a normal task processing scheme and corresponds to a binary decision problem [37]. Meanwhile, DQN is applied to solve the problem proposed in this paper.
- **Q-learning-based Multi-option Scheme** (QMS): The Q-learning method is applied to solve the proposed problem.
- **Q-learning-based Double-option Scheme** (QDS): The Q-learning method is applied to the normal task processing scheme.

In all of the DDS, QMS and QDS, the LET-V2X and TDMA are used to support the volunteer vehicle to UAV uploading and the UAV to requiring vehicle downloading communications. The main differences are the data processing schemes in the UAV task processing step.

Fig. 4 shows the convergence performance of the DMS and the other three comparison schemes when the V_{uav} is set as $8\,m/s$ and D_0 is $200\,m$. The blue line represents the reward of DMS, and we can find that when the number of iterations is less than about 240, the reward increases rapidly, and the growth rate gradually decreases until convergence. The green line represents the performance of QMS. We can find that in the early stage of the iteration process, the change trend of the reward is very similar to that of DMS. However, when the number of iterations is more than 200, the reward of QMS no longer increases, and it becomes converge. The red line indicates the change of DDS scheme reward. Its early growth rate is lower than that of DMS, QMS and QDS. However, when the number of iterations increases to about 500, the reward of DDS becomes converge, and the convergence value is very close to that of QMS. The orange line shows the performance of QDS. It has finally converged to about 4 and is obviously the worst one in all of schemes. From the overall comparison, DMS has higher convergence value of reward than QMS because the state space of the system is continuous. Using neural network to estimate Q value will have higher accuracy, the Q-table leads to a lower accuracy. Compared with DDS, DMS has similar convergence speed, but higher reward convergence value. The neural networks used in both DMS and DDS have the same parameter settings, so the convergence speeds are similar. Compared with DDS, in DMS scheme, the UAV has one more option in action selection, which is model two. In system states, collecting some data packets without data processing will lead to the problem of long time consumption, while deep data processing may cause the problems of not only longer time-consuming, but also higher cost. In addition, simple data processing can better balance the requirements of time consumption and cost, DMS can bring better reward than DDS.

Fig. 5 shows the time consumed by different schemes when the total number of iterations is 1000. We can find that the execution time of DMS and DDS based on DQN is about 8 minutes, while the execution time of QMS based on Q-Learning is nearly 12 minutes, which is nearly 4 minutes more than that of DMS and DDS. The biggest difference between DQN and Q-learning algorithm lies in the estimation method of Q value. DQN uses neural network to fit Q function, and the execution time mainly spends training neural network. Corresponding to the increasing number of Q-table elements, and the time required to retrieve the Q-value becomes longer and longer. While the time required for each training and calculation of Q value of neural network is almost fixed, the execution time of QMS based on Q-learning is longer than that of DMS and DDS based on DQN. Compared with QMS, QDS has a relatively lower execution time. It is because the action number in normal task offloading scheme is less one than proposed scheme. Lower action dimension brings lower Q-table dimension and query time.

Fig. 6 shows the relationship between the mean system cost and the flight velocity of UAV V_{uav} when the D_0 is fixed as $200\,m$. From the figure, we can find that the system cost of the all schemes decrease with the increasing of V_{uav} because the values of V_{uav} affect the transmission rate of communication link between UAV and requiring vehicle directly. For the temporary UAVs, the UAV is flying towards to the requiring

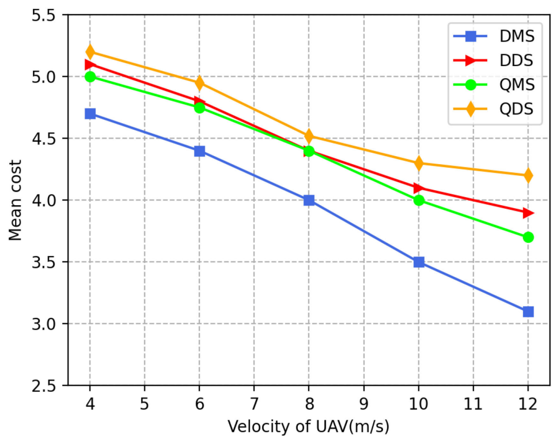


Fig. 6. System cost comparison under different flight velocities of UAV.

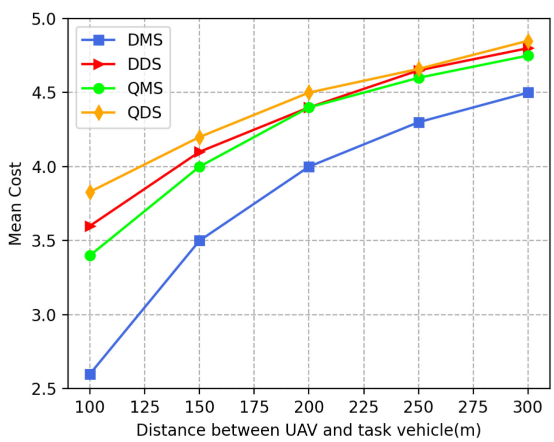


Fig. 7. System cost comparison under different distances between UAV and requiring vehicle.

vehicle, and the faster the UAV is, the UAV can approach the requiring vehicle closely, the transmission rate can also increases faster. According to the definition of $T_{total,i}$, it can be known that when the data packet size remains, the greater the growth rate of channel transmission rate, the shorter the time required to download the data packet. In this case, the UAV tends to select no data processing or simple data processing. Meanwhile, in our design, computing resources are more expensive than transmission resources, so the system cost decreases with the increasing of the V_{uav}.

Fig. 7 shows the relationship between the mean system cost and initial distance between the UAV and requiring vehicle D_0 when V_{uav} is 8 m/s. From the figure, we can find that the mean system costs of the four schemes increase with the increasing of D_0. According to the Eq. 1 and Eq. 3, the greater D_0 brings lower channel transmission rate. Thus, if the V_{uav} remains, the greater D_0 leads to the long time consumption when requiring vehicle is in the area. Meanwhile, the requiring vehicle tends to select simple data processing or deep data processing when the channel transmission rate is low because the time spent for transmitting the origin data is large, for the sum of the time spent on data processing and data transmission is calculated. Note that the requiring vehicle only needs to download the results of data

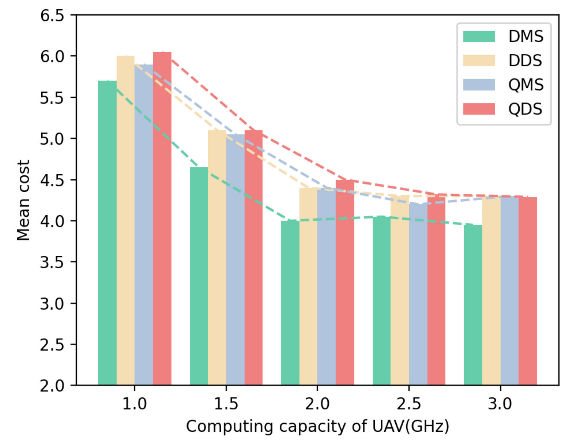


Fig. 8. System cost comparison under different computation capacities of UAV.

processing from the UAV, the data size is much smaller than the original one. Due to the mentioned above reasons, it is obvious that the system cost increases with the increasing of the distance D_0. Based on the advantages of neural network and multi-option data processing mode, we can find that the performance of DMS is better than DDS, QMS and QDS in both Figs. 6 and 7.

Fig. 8 shows the mean system cost required for the requiring vehicle to successfully collect data packets under different UAV computing capacities $\mathcal{C}_u$. We can find that when the $\mathcal{C}_u$ is less than 2 GHz, the mean cost decreases rapidly with the increasing of $\mathcal{C}_u$. However, when the $\mathcal{C}_u$ is greater than or equal to 2 GHz, the increase of $\mathcal{C}_u$ does not significantly affect the mean cost, the mean cost fluctuates in a small range. In addition, we find that the data processing time increases exponentially with the decreasing of $\mathcal{C}_u$. When $\mathcal{C}_u$ becomes small, the data processing time is long. Therefore, when the communication rate is small, it is difficult for requiring vehicle to perform effective data collecting, for some data packets which involve a grim delay requirement. Then, more invalid data packages are generated. The cost caused by invalid data packets need to be borne by the requiring vehicle. The mean cost will also be raised. With the increasing of $\mathcal{C}_u$, the data processing time decreases, the number of invalid data packets will also be reduced until it reaches the lowest. Then, the number of invalid data packets will no longer be affected by $\mathcal{C}_u$.

Fig. 9 shows the mean system cost required for the requiring vehicles to successfully collect data packets under different transmission powers of UAV. We can find that the system cost decreases with the increasing of transmission power. This is because that the increased transmission power brings the growth of communication rate of UAV to requiring vehicle links. Then, the delay requirements of data packets will become easier to be satisfied. At this time, the advantages brought by data processing will no longer be prominent, and the demand for fine-grained data processing (model 3) or non-fine-grained data processing (model 2) of requiring vehicles will decrease. In the whole progress of data collection, the mean system cost includes the

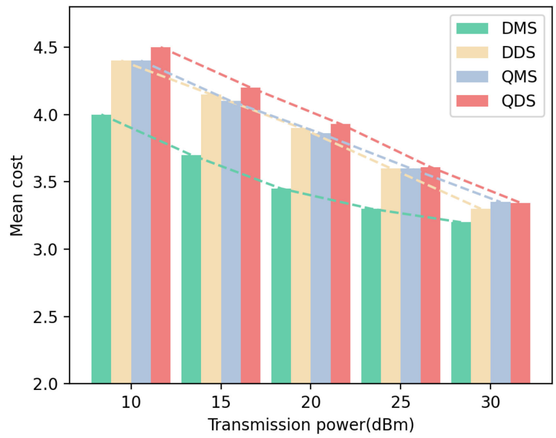


Fig. 9. System cost comparison under different transmission powers of UAV.

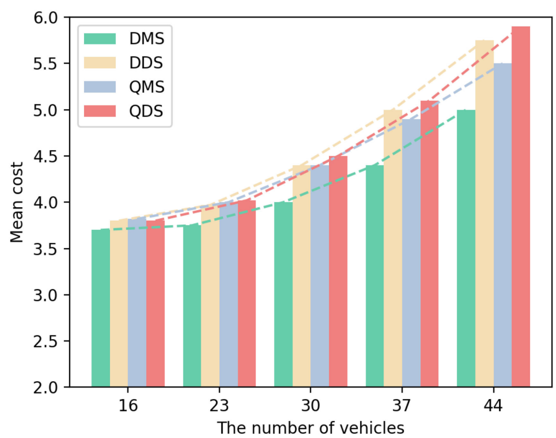


Fig. 10. System cost comparison under different number of vehicles.

cost of data processing and transmission. Therefore, under the given delay requirement, when the computation resource consumption of UAV in both the model 2 and 3 task processing decreases, the mean cost also decreases.

Fig. 10 shows the mean system cost of the requiring vehicles to successfully collect data packets under different number of volunteer vehicles. We can find that the system cost increases with the increasing of the number of volunteer vehicles, and the growth rate of system cost also increases. This is because that the number of volunteer vehicles directly affects the allocation of temporary UAV resources. Smaller the number of volunteer vehicles is, more resources the requiring vehicle can be allocated. From Fig. 8 and Fig. 9, we can find that the decline of whatever the communication rates or computing capacity can increase the system cost.

VI. Conclusion

In this paper, we propose a learning based channel allocation and task processing strategy in the temporary UAV-assisted VECN, which is applied to the situation that a requiring vehicle collects traffic data from and a set of volunteer vehicles, through the help of temporary UAV. The objective of this work is to collect enough data at a lower service cost and ensure that the data is still validated before it arrives the requiring vehicle. A combined OFDMA and TDMA transmission model is designed to support the transmission, and a DQN-based task processing mode selection is also proposed. The simulation results show that the proposed scheme has good convergence performance. Compared with the normal double-option data processing scheme and Q-learning scheme, the proposed DQN-based task processing scheme reduces the cost effectively, and can obtain a higher convergence reward.

References

[1] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, “Future intelligent and secure vehicular network toward 6G: Machine-learning approaches,” *Proc. IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.

[2] X. Jiang, F. R. Yu, T. Song, and V. C. M. Leung, “Resource allocation of video streaming over vehicular networks: A survey, some research issues and challenges,” *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2021.3065209.

[3] F. Tang, B. Mao, N. Kato, and G. Gui, “Comprehensive survey on machine learning in vehicular network: Technology, applications and challenges,” *IEEE Commun. Surv. Tut.*, vol. 23, no. 3, pp. 2027–2057, Jul.–Sep. 2021.

[4] J. Zhang, H. Guo, J. Liu, and Y. Zhang, “Task offloading in vehicular edge computing networks: A load-balancing solution,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2092–2104, Feb. 2020.

[5] X. Huang, R. Yu, S. Xie, and Y. Zhang, “Task-container matching game for computation offloading in vehicular edge computing and networks,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 10, pp. 6242–6255, Oct. 2021.

[6] J. Hu, C. Chen, L. Cai, M. R. Khosravi, Q. Pei, and S. Wan, “UAV-assisted vehicular edge computing for the 6G internet of vehicles: Architecture, intelligence, and challenges,” *IEEE Commun. Standards Mag.*, vol. 5, no. 2, pp. 12–18, Jun. 2021.

[7] W. Khawaja, I. Guvenc, D. W. Matolak, U. C. Fiebig, and N. Schneckenberger, “A survey of air-to-ground propagation channel modeling for unmanned aerial vehicles,” *IEEE Commun. Surv. Tut.*, vol. 21, no. 3, pp. 2361–2391, Jul.–Sep. 2019.

[8] X. Zhong, Y. Guo, N. Li, Y. Chen, and S. Li, “Deployment optimization of UAV relay for malfunctioning base station: Model-free approaches,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11971–11984, Dec. 2019.

[9] D. N. K. Jayakody, T. D. P. Perera, A. Ghrayeb, and M. O. Hasna, “Self-energized UAV-assisted scheme for cooperative wireless relay networks,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 578–592, Jan. 2020.

[10] S. Zhang, H. Zhang, B. Di, and L. Song, “Joint trajectory and power optimization for UAV relay networks,” *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 161–164, Jan. 2018.

[11] S. Jeong, O. Simeone, and J. Kang, “Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.

[12] B. Liu, W. Zhang, W. Chen, H. Huang, and S. Guo, “Online computation offloading and traffic routing for UAV swarms in edge-cloud computing,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8777–8791, Aug. 2020.

[13] Z. Xia *et al.*, “Multi-agent reinforcement learning aided intelligent UAV swarm for target tracking,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 931–945, Jan. 2022.

[14] S. R. Pokhrel, J. Jin, and H. L. Vu, “Mobility-aware multipath communication for unmanned aerial surveillance systems,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6088–6098, Jun. 2019.

[15] S. Zhu, L. Gui, D. Zhao, N. Cheng, Q. Zhang, and X. Lang, “Learning-based computation offloading approaches in UAVs-assisted edge computing,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 928–944, Jan. 2021.

[16] D. Liu, J. Wang, Y. Xu, Y. Zhang, and A. Anpalagan, “Opportunistic data ferrying in UAV-assisted D2D networks: A dynamic hierarchical game,” in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–6.

[17] A. K. Jain, A. Ross, and S. Prabhakar, “An introduction to biometric recognition,” *IEEE Trans. Circuits, Syst. Video Technol.*, vol. 14, no. 1, pp. 4–20, Jan. 2004.

[18] Y. Zhou, F. Tang, Y. Kawamoto, and N. Kato, “Reinforcement learning-based radio resource control in 5G vehicular network,” *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 611–614, May 2020.

[19] Y.-J. Ku, S. Baidya, and S. Dey, “Adaptive computation partitioning and offloading in real-time sustainable vehicular edge computing,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13221–13237, Dec. 2021.

[20] X. Huang, P. Li, R. Yu, Y. Wu, K. Xie, and S. Xie, "FedParking: A federated learning based parking space estimation with parked vehicle assisted edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9355–9368, Sep. 2021.

[21] C. Yang, W. Lou, Y. Liu, and S. Xie, "Resource allocation for edge computing-based vehicle platoon on freeway: A contract-optimization approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15988–16000, Dec. 2020.

[22] S. Yue *et al.*, "TODG: Distributed task offloading with delay guarantees for edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 7, pp. 1650–1665, Jul. 2022.

[23] Y. Zhang, X. Lan, J. Ren, and L. Cai, "Efficient computing resource sharing for mobile edge-cloud computing networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1227–1240, Jun. 2020.

[24] Z. Ni *et al.*, "MSCET: A multi-scenario offloading schedule for biomedical data processing and analysis in cloud-edge-terminal collaborative vehicular networks," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, to be published, doi: 10.1109/TCBB.2021.3131177.

[25] H. Peng and X. S. Shen, "DDPG-based resource management for MEC/UAV-assisted vehicular networks," in *Proc. IEEE 92nd Veh. Technol. Conf.*, 2020, pp. 1–6.

[26] L. Zhao, K. Yang, Z. Tan, X. Li, S. Sharma, and Z. Liu, "A novel cost optimization strategy for SDN-enabled UAV-assisted vehicular computation offloading," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3664–3674, Jun. 2021.

[27] X. Wang, H. Lin, H. Zhang, D. Miao, Q. Miao, and W. Liu, "Intelligent drone-assisted fault diagnosis for B5G-enabled space-air-ground-space networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 2849–2860, Oct.–Dec. 2021.

[28] Y. Liu *et al.*, "Joint communication and computation resource scheduling of a UAV-assisted mobile edge computing system for platooning vehicles," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2021.3082539.

[29] Y. He, D. Zhai, R. Zhang, J. Du, G. S. Aujla, and H. Cao, "A mobile edge computing framework for task offloading and resource allocation in UAV-assisted VANETs," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2021, pp. 1–6.

[30] Y. Yu, X. Bu, K. Yang, H. Yang, X. Gao, and Z. Han, "UAV-aided low latency multi-access edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 5, pp. 4955–4967, May 2021.

[31] T. Zhang, Y. Xu, J.K.-K. Loo, D. Yang, and L. Xiao, "Joint computation and communication design for UAV-assisted mobile edge computing in IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5505–5516, Aug. 2020.

[32] D. Yang, Q. Wu, Y. Zeng, and R. Zhang, "Energy trade-off in ground-to-UAV communication via trajectory design," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6721–6726, Jul. 2018.

[33] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[34] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

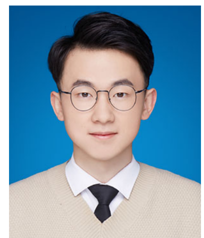
[35] T. Kimura, "Performance analysis of cellular-relay vehicle-to-vehicle communications," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3396–3411, Apr. 2021.

[36] N. Hossein Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based Internet of Things services: Comprehensive survey and future perspectives," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 899–922, Dec. 2016.

[37] L. Zhang and N. Ansari, "Optimizing the operation cost for UAV-aided mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 6085–6093, Jun. 2021.

Chao Yang received the Ph.D. degree in signal and information processing from the South China University of Technology, Guangzhou, China, in 2013. He is currently with the School of Automation, Guangdong University of Technology, Guangzhou. From July 2014 to July 2016, he was a Research Associate with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His research interests include VANETs, edge computing, and smart grid.

Baichuan Liu received the B.E. degree from the Guangdong University of Technology, Guangzhou, China, in 2020. He is currently working toward the M.S. degree with the School of Automation, Guangdong University of Technology, Guangzhou. His research interests include wireless communication networks, cooperative communications, and intelligent edge computing.



Haoyu Li received the B.E. degree from the Guangdong University of Technology, Guangzhou, China, in 2020. He is currently working toward the M.S. degree with the School of Automation, Guangdong University of Technology, Guangzhou. His research interests include wireless communication networks, smart grid, and intelligent edge computing.

Bo Li received the Ph.D. degree in computer science and technology from the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou, China, in 2021. From September 2014 to June 2016, he was a master student with the School of Engineering, Sun Yat-sen University. He is currently a Postdoctoral Research Fellow with the School of Automation, Guangdong University of Technology, Guangzhou. His research interests include intelligent transportation system, traffic information processing, and Big Data technology. Dr. Li is currently a very Active Reviewer for some international journals and a Member of program committee for many international conferences.

Kan Xie received the Ph.D. degree in control science and engineering from the Guangdong University of Technology, Guangzhou, China, in 2017. He joined the Institute of Intelligent Information Processing, Guangdong University of Technology, where he is currently an Associate Professor. His research interests include machine learning, nonnegative signal processing, blind signal processing, smart grid, and Internet of Things.

Shengli Xie (Fellow, IEEE) received the M.S. degree in mathematics from the Central China Normal University, Wuhan, China, in 1992, and the Ph.D. degree in control theory and applications from the South China University of Technology, Guangzhou, China, in 1997. He is currently a Full Professor and the Head of the Institute of Intelligent Information Processing, Guangdong University of Technology, Guangzhou, and the Director of the Laboratory for Intelligent Information Processing, the Director of the Guangdong-Hong Kong-Macao Joint Laboratory for Smart Discrete Manufacturing. He has authored or coauthored two books and more than 150 scientific papers in journals and conference proceedings. His research interests include wireless networks, automatic control, and blind signal processing.