Received 20 November 2023; accepted 5 December 2023. Date of publication 11 December 2023; date of current version 29 December 2023.

Digital Object Identifier 10.1109/OJCOMS.2023.3341251

Task-Oriented Satellite-UAV Networks With Mobile-Edge Computing

PENG WEI [1] (Member, IEEE), WEI FENG [1] (Senior Member, IEEE), YUNFEI CHEN [2] (Senior Member, IEEE), NING GE [1] (Member, IEEE), WEI XIANG [3] (Senior Member, IEEE), AND SHIWEN MAO [4] (Fellow, IEEE)

[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]Department of Engineering, University of Durham, DH1 3LE Durham, U.K.
[3]School of Computing, Engineering and Mathematical Sciences, La Trobe University, Melbourne, VIC 3086, Australia
[4]Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849, USA

CORRESPONDING AUTHOR: W. FENG (e-mail: fengwei@tsinghua.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFA0711301; in part by the National Natural Science Foundation of China under Grant 62341110 and Grant U22A2002; and in part by the Suzhou Science and Technology Project.

ABSTRACT Networked robots have become crucial for unmanned applications since they can collaborate to complete complex tasks in remote/hazardous/depopulated areas. Due to the cost inefficiency of deploying cellular network infrastructure in these areas, hybrid satellite-UAV networks emerge as a promising solution. These networks provide seamless and on-demand connectivity for multiple robots with various task requirements, and support computation-intensive and latency-sensitive services through mobile edge computing (MEC)-based offloading. However, to complete tasks in limited times, the rapid collective movement of mobile robots may cause frequent service migration, and a large number of gathered robots may compete for limited bandwidth resources in satellite and UAV communications. As a result, offloading latency may increase significantly. To address this issue, the average completion time of multi-robot offloading in task-oriented satellite-UAV networks with MEC is formulated as an optimization problem. Unlike conventional mobility-aware MEC-based offloading schemes, joint optimization of mobility control, data offloading, and resource allocation is proposed using velocity control of multiple robots. According to Lyapunov optimization, the original optimization problem is simplified into minimizing the average completion time of offloading for all robots within UAV and satellite coverage. A multi-agent Q-learning algorithm, including multi-group dual-agent Q-learning, is proposed based on local state observation and global reward calculation. In each dual-agent Q-learning, one agent is responsible for velocity control and communication resource allocation, while the other is responsible for data offloading and computational resource allocation. The convergence of the proposed multi-agent Q-learning algorithm is also theoretically analyzed. Simulation results show that the proposed scheme can effectively reduce the offloading latency by up to 35% in the multi-robot environment over its conventional counterparts.

INDEX TERMS Offloading, reinforcement learning, resource allocation, satellite-UAV network, velocity control.

I. INTRODUCTION

NETWORKED robots have received increasing attention since they use a communication network to connect and coordinate with each other/humans, sensors, and computers to complete complex tasks [1], [2], such as nuclear decommissioning to handle radiation and contamination hazards [1], cave exploration for search and rescue [3], and maintenance of offshore wind turbines [4]. To enable teams of robots for autonomous task completion in remote/hazardous/depopulated areas, satellite networks provide a cost-effective deployment strategy for wide-area coverage and information exchange. However, their high propagation latency and severe path loss challenge the diverse and stringent requirements of robotic tasks in theses areas. To address these challenges, satellite-unmanned aerial vehicle (UAV) integrated networks are a promising solution to provide on-demand services [5], [6], [7]. Furthermore, for latency-sensitive and computation-intensive robotic applications, deploying edge computing servers on UAVs/satellites enables robots to send data to the network edge for processing, such as mobile edge computing (MEC) [8]. However, to accomplish tasks in a limited time, the collective movement of mobile robots may lead to a large number of gathered robots competing for limited network resources and frequent service migration due to the rapid movement of robots. As a result, the service latency for these robots may increase significantly.

An increasing number of studies have been conducted to improve the quality of service (QoS) of data offloading in satellite-UAV networks by deploying edge servers on UAVs [9], [10], [11], [12] or transferring data to edge servers through UAVs [13], [14], [15]. In [9], considering the energy dynamics in wireless-powered Internet of Things (IoT) devices and time-varying channel conditions, a deep learning-based offloading strategy was given to maximize the task success rate in satellite-UAV-served IoT systems. Furthermore, by offloading data from a UAV to its neighboring UAVs, the authors of [12] proposed a collaborative computation offloading scheme in the centralized and distributed UAV-enabled MEC networks. For IoT users in remote areas, the authors of [10] formulated a model of computational resource allocation and task scheduling for edge servers in UAVs and proposed an reinforcement learning (RL)-based offloading algorithm to minimize the total cost of server usage, energy consumption of the IoT devices, and offloading delay. In [11], a satellite-UAV-MEC collaborative architecture for offloading in vehicular applications was proposed, and a joint optimization of UAV deployment and resource allocation was employed to maximize the long-term profit. In [13], to efficiently process the data collected by a UAV, by offloading the data to the cloud server via multiple satellites and edge servers in cellular base stations (BSs), a distributionally robust optimization problem was formulated to minimize the system latency. Using unmanned surface vehicles in maritime communication networks, [14] presented a collaborative offloading scheme to reduce the task execution time. To guarantee the safety of traffic offloading in satellite-UAV networks, the authors of [15] provided a blockchain-based federated learning architecture, and then proposed a node security evaluation mechanism and an improved practical Byzantine fault tolerance algorithm. However, such schemes require frequent service migration between multiple MEC servers when terminals, UAVs, or satellites move fast [16], [17]. The above works do not consider service migration in satellite-UAV networks. Moreover, compared to static BSs in terrestrial networks, the mobility of satellites and UAVs may trigger more service migrations [18], [19]. As a result, frequent service migration will significantly increase the service latency.

Many studies have been conducted to address the problem of frequent service migration in satellite- and UAV-related networks by detecting the mobility of terminals [20], satellites [21], [22], [23], and UAVs [23], [24]. In [20], a distributed two-layer decomposition model was proposed to minimize the migration cost according to the mobility of users in satellite networks. Considering satellite motion and the relationship between space-based services, the authors of [21] developed a migration strategy to reduce the migration delay and packet loss rate. The difference between the profit and energy consumption during migration was optimized to balance between migration delay and energy consumption. In [22], a backhaul migration policy based on satellite mobility was presented to enable data offloading for invisible satellites. Based on this, an RL-based privacy-preserving offloading scheme was proposed to jointly optimize the task completion time, energy consumption, communication reliability, and user privacy leakage. In [23], based on the mobility of UAVs and satellites, the migration cost and additional delay in live migration and reinstantiation of virtual network function (VNF) remapping were modeled. Then, the joint VNF mapping and scheduling in the UAV-supported satellite-terrestrial networks were efficiently optimized. In [24], considering the migration of UAV, a multi-path transmission control protocol was proposed to dynamically allocate bandwidth resources for static and mobile users. These existing works focus on the mobility-aware optimization of data offloading and service migration for mobile devices/users. Nevertheless, in areas with no or weak network coverage, these optimization methods are ineffective since a robot's mobility cannot be detected by the network. An alternative solution is to control robots to move from areas without network coverage to those with network coverage.

For mobility control, according to the willingness of users, a closed-loop system model based on spatial-temporal mobility control was developed in [25], [26]. However, this model is intended for humans, not robots. For mobile robots, the authors of [27] analyzed the effect of velocity on the stability of the wireless control system to make a tradeoff between vehicle velocity, control stability, and channel quality. In [28], based on the prediction of the channel quality between a robot and a cellular BS, a co-optimization of the motion and communication costs was proposed to minimize the energy consumption. Furthermore, a class of communication-aware motion planning methods has been proposed for robotic applications, including trajectory planning for channel assessment and target tracking [29], distributed control based on robotic mobility to maintain end-to-end connections [30], [31], energy-efficient trajectory planning for a UAV [32], and communication-constrained path planning for robotic surveillance [33]. However, in these existing works, data offloading and service migration in satellite-UAV networks have not been considered. Furthermore, when the radio communication is unavailable, mobility control for multi-robot offloading is not considered. The communication unavailability might be due to heavy network loads, severe channel fading, damage to network infrastructures, and so on. The joint velocity control and offloading decision has been proposed in [34], whose system model focuses on satellite-terrestrial networks. However, in remote/depopulated areas, terrestrial network infrastructures may not be deployed. Moreover, the optimization problem and the corresponding RL-based algorithm aimed at a single robot rather than multiple robots. In the multi-robot optimization problem, the RL-based algorithm in [34] may be inefficient in resource allocation due to the lack of global resource observation and reward calculation for each agent. This can lead to increased service latency, especially when computation and communication resources are limited.

Against this backdrop, this paper proposes to jointly optimize velocity control and MEC-based offloading for multiple robots in task-oriented satellite-UAV networks. We consider scenarios where multiple mobile robots cooperate in groups to accomplish multiple tasks. When these robots move to the locations of targeted tasks, they periodically offload perceived data to UAV/gateway-based MEC servers for processing, and subsequently receive computational results from selected MEC servers. When wireless communications are unavailable in an access point (AP) coverage, the low-speed movements of mobile robots will increase local computations. When the wireless communication is available in the AP coverage, high-speed movements of mobile robots will lead to frequent service migration. Thus, the wireless communication availability and the velocities of mobile robots are two key factors that affect service latency. Moreover, a large number of mobile robots may compete for limited bandwidth resources in satellite and UAV communications, potentially increasing the communication delay in data offloading. In this paper, our contributions are listed below.

- The optimization problem concerning data offloading, velocity control, and resource allocation for multiple robots is formulated to minimize the average completion time of offloading. Then, inspired by the idea that Lyapunov optimization can transform the optimization problem with long-term constraints into one with short-term constraints, the proposed optimization problem for robots over their entire journey is decomposed into the optimization problem for robots over individual AP coverage regions.
- We propose a multi-agent Q-learning algorithm that utilizes multi-group dual-agent Q-learning to solve the proposed problem, while considering the observed wireless communication availability, the reduced computational resource state, and a global reward. In the dual-agent Q-learning framework, one agent is responsible for offloading decision-making and computational resource allocation, while the other is responsible for velocity control and communication resource allocation.
- The convergence of the proposed multi-agent Q-learning algorithm is analyzed in terms of Q functions, optimal Q functions, and convergence rate. Simulation results validate the effectiveness of the proposed scheme in terms of convergence and offloading time.

The rest of this paper is organized as follows. In Section II, the system model of data offloading, service migration, and velocity control for multiple robots in the satellite-UAV network is introduced. In Section III, the optimization problem and its simplified version are formulated, and the novel multi-agent Q-learning algorithm is proposed. In Section IV, the convergence of the proposed multi-agent Q-learning algorithm is analyzed. In Section V, simulation results are provided. Finally, in Section VI, conclusions are drawn.

II. SYSTEM MODEL

As shown in Fig. 1, in a multi-robot environment, we assume a task-oriented satellite-UAV network that is responsible for serving a collection of tasks. Note that tasks in this paper refer to tasks accomplished by collaborative robots, such as environmental exploration, rather than tasks within a network. These tasks are related to specific locations in the environment. Serving a task means that a group of robots are actually located in the vicinity of that task, and they move along their pre-defined trajectories to their destination locations within a given time to complete the task. Meanwhile, as they move, robots continuously sense their surrounding environments, send the sensed data to edge servers for computation and processing, and finally take appropriate actions according to the computational results, such as localization and mapping, and obstacle/risk avoidance.

We assume that there are V tasks and each of them is cooperatively completed by U mobile robots. The uth robot in the vth task passes through a communication coverage region involving N_{uv1} UAVs with the index set $\mathcal{N}_{uv1} = \{\mathrm{AP}_{\mathrm{G},uv,1}, \mathrm{AP}_{\mathrm{G},uv,2}, \ldots, \mathrm{AP}_{\mathrm{G},uv,N_{uv1}}\}$ and one satellite. The satellite covers all regions where all robots operate. The region covered only by the satellite is divided into N_{uv2} non-overlapped sub-regions with the index set $\mathcal{N}_{uv2} = \{\mathrm{AP}_{\mathrm{S},uv,1}, \mathrm{AP}_{\mathrm{S},uv,2}, \ldots, \mathrm{AP}_{\mathrm{S},uv,N_{uv2}}\}$, $\mathcal{N}_{uv1} \cap \mathcal{N}_{uv2} = \varnothing$, $N_{uv} = N_{uv1} + N_{uv2}$, $u = 1, 2, \ldots, U$, and $v = 1, 2, \ldots, V$. UAVs are pre-deployed according to task locations and satellite navigation, and hover in fixed positions in the air until all tasks are completed. Each AP is equipped with an independent MEC server, so the total numbers of APs and MEC servers are both N. According to [35], the MEC server provides resources of computation, communication, and storage through virtual machines (VMs). The service provider leases VMs from the MEC server to offer services. We assume that an offloading service occupies a VM. Under the assumption that orthogonal multiple access technology, such as orthogonal frequency division multiple access (OFDMA),

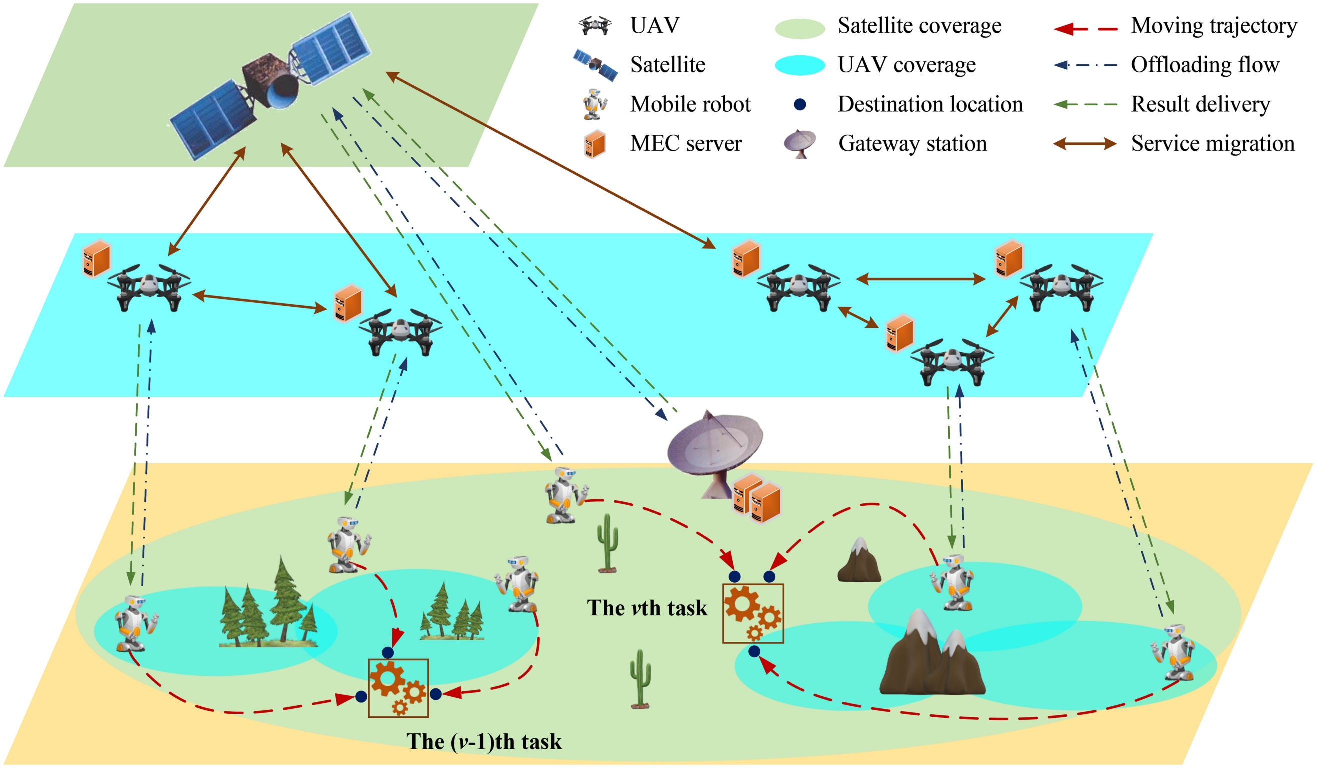


FIGURE 1. Illustration of the system model for multi-group robots in the task-oriented satellite-UAV network with MEC.

is utilized, there will be no interference between satellite-robot and UAV-robot links, between UAV-UAV links, and between UAV-robot links. Other communication technologies are not specified, because our results are general enough. The UAV-UAV links and satellite-UAV links are used for service migration purposes only. The terrestrial gateway with MEC servers are selected over satellite with MEC servers because they have lower complexity in terms of coverage and maintenance [36]. The satellite is responsible for transferring data from robots to the gateway and sending computational results from the gateway to robots.

In multi-robot offloading, a robot can offload its data either directly to the MEC server on a UAV or via satellite to the MEC server based on the gateway. The following section describes the models for local computation, MEC computation, communication, service migration, and velocity control.

A. LOCAL COMPUTATION MODEL

According to [10], the local computation delay $T_{1,u,v,n}(t)$ of the uth mobile robot in the vth task in the time slot t is expressed as

$$T_{1,u,v,n}(t) = (1 - \alpha_{u,v,n}(t))\frac{D_{u,v}(t)\Phi}{f_{\text{local},u,v}(t)}, \tag{1}$$

where the binary variable $\alpha_{u,v,n}(t)$ takes values from the discrete set $\{0, 1\}$, $\alpha_{u,v,n}(t) = 0$ denotes local computation, $\alpha_{u,v,n}(t) = 1$ denotes that data is offloaded to the nth MEC server, $D_{u,v}(t)$ is the size of the data generated by the uth mobile robot in the vth task in time slot t, $f_{\text{local},u,v}(t)$ is the computation frequency of the mobile robot in each CPU cycle, and the constant Φ represents the number of CPU cycles per bit.

B. MEC COMPUTATION MODEL

When the offloaded data is processed by the nth MEC server in the tth slot, according to [10], the computation delay $T_{2,u,v,n}(t)$ is expressed as

$$T_{2,u,v,n}(t) = \alpha_{u,v,n}(t)\frac{D_{u,v}(t)\Phi}{f_{u,v,n}(t)}, \tag{2}$$

where $f_{u,v,n}(t)$ denotes the computation frequency allocated by the nth MEC server to the offloaded data of the uth robot in the vth task.

C. COMMUNICATION MODEL

We assume that the wireless communication uplink and downlink are allocated the same bandwidth, between a mobile robot and a UAV, between a mobile robot and a satellite, and between the satellite and the gateway. Moreover, to avoid an excessive number of states and actions in the proposed reinforcement learning algorithm in Section III, we assume that the uplink and downlink have the same communication rate for the channels between mobile robot and UAV, between mobile robot and satellite, and between satellite and gateway. In reality, uplink and downlink may have different rates. When the offloaded data is transmitted through a UAV, according to [10], the communication delay $T_{3,u,v,m,n}(t)$ in the mth AP coverage region ($m = 1, 2, \ldots, N_{u,v}$) is expressed as

$$T_{3,u,v,m,n}(t) = \frac{\alpha_{u,v,n}(t)\big(D_{u,v}(t) + \bar{D}_{u,v}(t)\big)}{W_{u,v,m}\log_2\left(1 + \frac{p_{\mathrm{C}} h_{\mathrm{C}}^2}{\sigma_{\mathrm{C}}^2}\right)}, \tag{3}$$

where $\bar{D}_{u,v}(t)$ is the size of the computational result corresponding to $D_{u,v}(t)$, $W_{u,v,m}$ is the communication bandwidth allocated by the UAV in the mth AP coverage region to the uth robot in the vth task, p_{C}, h_{C}, and σ_{C}^2 denote the transmit power, channel gain, and channel noise power in UAV communications, respectively.

When offloaded data is transferred through the satellite, according to [10], the communication delay $T_{4,u,v,m,n}(t)$ in the mth AP coverage region is expressed as

$$T_{4,u,v,m,n}(t) = \alpha_{u,v,n}(t)\left(\frac{2(d_{\mathrm{GS}} + d_{\mathrm{SE}})}{c} + \big(D_{u,v}(t) + \bar{D}_{u,v}(t)\big) \times \left(\frac{1}{r_{\mathrm{GS},u,v,m}} + \frac{1}{r_{\mathrm{SE},u,v,m}}\right)\right), \tag{4}$$

where d_{GS} and d_{SE} represent the distances from a mobile robot to the satellite and from the satellite to the gateway, respectively. $r_{\mathrm{GS},u,v,m}$ and $r_{\mathrm{SE},u,v,m}$ denote the communication rates between a mobile robot and the satellite and between the satellite and the gateway station, respectively. Compared to the large total bandwidth between the satellite and the gateway station, the total bandwidth in the uplink from the mobile robot to the satellite is limited. Thus, (4) is rewritten as

$$T_{4,u,v,m,n}(t) = \alpha_{u,v,n}(t)\left(\frac{2(d_{\mathrm{GS}} + d_{\mathrm{SE}})}{c} + \frac{D_{u,v}(t) + \bar{D}_{u,v}(t)}{W_{u,v,m}\log_2\left(1 + \frac{p_{\mathrm{S}} h_{\mathrm{S}}^2}{\sigma_{\mathrm{S}}^2}\right)} + \frac{D_{u,v}(t) + \bar{D}_{u,v}(t)}{r_{\mathrm{SE},u,v,m}}\right), \tag{5}$$

where $W_{u,v,m}$ is the communication bandwidth allocated by the satellite to the uth robot of the vth task, and p_{S}, h_{S}, and σ_{S}^2 denote the transmit power, channel gain, and channel noise power in satellite communication, respectively. It is noted that, when the bandwidth in the satellite-robot link is significantly larger than that in the satellite-gateway link, the delay for data transmission and result delivery in (4) and (5) can be ignored.

Upon assuming that each robot can choose only one communication mode in the mth AP coverage region, the communication delay is expressed as

$$T_{5,u,v,m,n}(t) = \beta_{u,v,m} T_{3,u,v,m,n}(t) + \big(1 - \beta_{u,v,m}\big) T_{4,u,v,m,n}(t), \tag{6}$$

where the binary variable $\beta_{u,v,m}$ takes values from the discrete set $\{0, 1\}$, and $\beta_{u,v,m} = 1$ and $\beta_{u,v,m} = 0$ denote the UAV communications and satellite communications, respectively.

D. SERVICE MIGRATION MODEL

We assume that a VM-based service migration occurs when the MEC server $\mathrm{M}_{u,v,t-1}$ in the $(t-1)$th slot is different from the MEC server $\mathrm{M}_{u,v,t}$ in the tth slot. Thus, according to [37], the service migration delay in the tth slot is expressed as

$$\begin{aligned} T_{6,u,v}(t) &= I\Big\{\mathrm{M}_{u,v,t-1} \neq \mathrm{M}_{u,v,t} \cap \big(\mathrm{M}_{u,v,t-1} \neq 0 \cap \mathrm{M}_{u,v,t} \neq 0\big) \\ &\quad \cap \big(\mathrm{M}_{u,v,t-1} \in \mathcal{N}_{uv1} \cup \mathrm{M}_{u,v,t} \in \mathcal{N}_{uv1}\big)\Big\} G_{u,v} \\ &\quad + I\Big\{\big(\mathrm{M}_{u,v,t-1} \in \mathcal{N}_{uv1} \cap \mathrm{M}_{u,v,t} \in \mathcal{N}_{uv2}\big) \\ &\quad \cup \big(\mathrm{M}_{u,v,t-1} \in \mathcal{N}_{uv2} \cap \mathrm{M}_{u,v,t} \in \mathcal{N}_{uv1}\big)\Big\} \Delta G, \end{aligned} \tag{7}$$

where $\mathrm{M}_{u,v,t} = 0$ stands for the local computation and $\mathrm{M}_{u,v,t} \in \mathcal{N}_{uv} = \mathcal{N}_{uv1} \cup \mathcal{N}_{uv2}$ represents the MEC computation. If the condition in $I\{\cdot\}$ that the MEC servers in the previous and current slots are different is satisfied, we have $I\{\cdot\} = 1$; otherwise $I\{\cdot\} = 0$. $G_{u,v}$ is the migration delay in the UAV network. According to the linear relationship between throughput and velocity in the "always migration" [35], we assume $G_{u,v} = \rho_{u,v}(t) T_{2,u,v,n}(t)$. The scaling factor $\rho_{u,v}(t) = \frac{v_{u,v}(t)}{v_{\max}}$ indicates that the higher the velocity is, the higher the migration delay will be. Since a satellite can only cover a region for a limited time, the handover between the satellite network and the UAV network should be considered. Based on the second term in (7), a handover occurs when the MEC servers at the $(t-1)$th slot and the tth slot are in the UAV network and the satellite network (or the satellite network and the UAV network), respectively. Compared to the migration delay $G_{u,v}$ in the UAV network, this handover results in an extra migration delay, which is expressed as a long-term migration cost ΔG. In practice, this cost is predefined as a constant determined monthly or annually by the network operator [17].

E. VELOCITY CONTROL MODEL

Due to the predefined moving trajectories of all robots, velocity control is reduced to adjust the velocity value. We suppose that the target velocity of the mobile robot in the mth AP coverage region is $v^*_{u,v,m} \in [v_{\min}, v_{\max}]$ with the minimal velocity $v_{\min}$ and the maximal velocity $v_{\max}$, and $v_{u,v,m}(l_0)$ and $v_{u,v,m}(l_{\mathrm{E}})$ stands for the initial velocity and final velocity of a mobile robot in the mth AP coverage region, respectively. Thus, the instantaneous velocity $v_{u,v,m}(l)$ in the lth slot of the mth AP coverage region is expressed as

$$v_{u,v,m}(l) = \begin{cases} \min\left\{v_{u,v,m}(l_0) + \mu l, v^*_{u,v,m}\right\}, & v_{u,v,m}(l_0) < v^*_{u,v,m}, \\ v_{u,v,m}(l_0), & v_{u,v,m}(l_0) = v^*_{u,v,m}, \\ \max\left\{v_{u,v,m}(l_0) - \mu l, v^*_{u,v,m}\right\}, & v_{u,v,m}(l_0) > v^*_{u,v,m}, \end{cases} \tag{8}$$

where $v_{u,v,m}(l_0) < v^*_{u,v,m}$, $v_{u,v,m}(l_0) = v^*_{u,v,m}$, and $v_{u,v,m}(l_0) > v^*_{u,v,m}$ correspond to acceleration, constant speed, and deceleration, respectively. $\mu > 0$ is the absolute value of acceleration/deceleration, $l \in \mathcal{L}_{u,v,m} = \{1, 2, \ldots, L_{u,v,m}\}$ is the time slot index in the mth AP coverage region, and $L_{u,v,m}$ denotes the total number of offloading slots in the mth AP coverage region with

$$L_{u,v,m} = \left\lfloor \frac{T^*_{u,v,m}}{\Delta T} \right\rfloor, \tag{9}$$

where ΔT is the offloading slot, $T^*_{u,v,m}$ denotes the moving time of the mobile robot passing through the mth AP coverage region, i.e.,

$$T^*_{u,v,m} = \begin{cases} \frac{c_m + \frac{(v_{u,v,m}(l_\mathrm{E}) - v_{u,v,m}(l_0))^2}{2\mu}}{v_{u,v,m}(l_\mathrm{E})}, & v^*_{u,v,m} \geq v_{u,v,m}(l_0), \\ \frac{c_m - \frac{(v_{u,v,m}(l_0) - v_{u,v,m}(l_\mathrm{E}))^2}{2\mu}}{v_{u,v,m}(l_\mathrm{E})}, & v^*_{u,v,m} < v_{u,v,m}(l_0), \end{cases} \tag{10}$$

where c_m represents the moving distance of the mobile robot in the mth AP coverage region. To simplify our model, we assume that the moving distances of different robots in the same AP coverage region are the same. If the mobile robot can accelerate or decelerate to the target velocity when leaving the mth AP coverage region, we have $v_{u,v,m}(l_\mathrm{E}) = v^*_{u,v,m}$.

Thus, the completion time of the offloading in the tth slot is expressed as

$$T_{u,v,m,n}(t) = T_{1,u,v,n}(t) + T_{2,u,v,n}(t) + T_{5,u,v,m,n}(t) + T_{6,u,v}(t). \tag{11}$$

Averaging the completion time of all robots in their entire journey yields

$$T_\mathrm{mean} = \frac{\sum_{u=1}^{U} \sum_{v=1}^{V} \sum_{m=1}^{N_{u,v}} \sum_{n=1}^{N} \sum_{l=1}^{L_{u,v,m}} T_{u,v,m,n}(l)}{\sum_{u=1}^{U} \sum_{v=1}^{V} \sum_{m=1}^{N_{u,v}} L_{u,v,m}}. \tag{12}$$

III. REINFORCEMENT LEARNING-BASED OPTIMIZATION

A. OPTIMIZATION PROBLEM

Based on (12), the optimization problem is formulated as

$$\begin{aligned} \min_{P_{u,v,m,n}} \quad & T_\mathrm{mean} && \text{(13a)} \\ \text{s.t.} \quad & T_{u,v,m,n}(t) \leq T_{\max,u,v}(t) && \text{(13b)} \\ & \sum_{m=1}^{N_{u,v}} T^*_{u,v,m} \leq T_{\mathrm{move},u,v} && \text{(13c)} \\ & \sum_{n=1}^{N} \alpha_{u,v,n}(t) = 1 && \text{(13d)} \\ & \sum_{m=1}^{N_{u,v}} \beta_{u,v,m} = 1 && \text{(13e)} \\ & \sum_{u=1}^{U} \sum_{v=1}^{V} \alpha_{u,v,n}(t) f_{u,v,n}(t) \leq F_n(t) && \text{(13f)} \\ & \sum_{u=1}^{U} \sum_{v=1}^{V} \alpha_{u,v,n}(t) W_{u,v,m} \leq B_{\mathrm{S},m}, \quad m \in \mathcal{M}_\mathrm{S} && \text{(13g)} \\ & \sum_{u=1}^{U} \sum_{v=1}^{V} \alpha_{u,v,n}(t) W_{u,v,m} \leq B_{\mathrm{C},m}, \quad m \in \mathcal{M}_\mathrm{C} && \text{(13h)} \end{aligned}$$

where variable $P_{u,v,m,n}$ involves $\alpha_{u,v,n}(t)$, $\beta_{u,v,m}$, $v^*_{u,v,m}$, $f_{u,v,n}(t)$, and $W_{u,v,m}$. Constraint (13b) indicates that the completion time of data offloading should be less than or equal to delay $T_{\max,u,v}(t)$ when all data are locally computed; In constraint (13c), $T_{\mathrm{move},u,v}$ is the maximal allowable moving time of the uth robot in the vth task over the entire journey; Constraint (13d) means that the mobile robot can only choose one MEC server for offloading in each time slot; Constraint (13e) means that the mobile robot can only select one communication mode in each AP coverage region; In constraint (13f), $F_n(t)$ indicates the available computational resources of the nth MEC server in the tth slot; Constraint (13g) indicates the available bandwidth of the robot-satellite communication link with the index set of satellites $\mathcal{M}_\mathrm{S} = \bigcup_{u=1}^{U} \bigcup_{v=1}^{V} \mathcal{N}_{uv2}$; Constraint (13h) indicates the available bandwidth of the UAV communication with the index set of UAVs $\mathcal{M}_\mathrm{C} = \bigcup_{u=1}^{U} \bigcup_{v=1}^{V} \mathcal{N}_{uv1}$.

Solving the optimization problem (13) is challenging due to the long-term moving time constraint (13c), as the offloading decisions of a robot at different time slots are correlated with the moving time. If long moving time is currently caused by a slow-moving robot in the area without network coverage, it will lead to excessive local computation. Based on the Lyapunov optimization in [37], an optimization problem with long-term constraints can be transformed into one with short-term constraints. By transforming the long-term constraint (13c) into the short-term constraint for each AP, the optimization problem (13) for the whole moving process can be decomposed into multiple sub-optimization problems for the moving process in the AP coverage region. Thus, the optimization model is simplified into minimizing the total average delay of all robots in their AP coverage regions, i.e.,

$$\begin{aligned} \min_{P_{u,v,m,n}} \quad & \frac{1}{\sum_{u=1}^{U} \sum_{v=1}^{V} L_{u,v,m}} \sum_{u=1}^{U} \sum_{v=1}^{V} \sum_{n=1}^{N} \sum_{l=1}^{L_{u,v,m}} T_{u,v,m,n}(l) && \text{(14a)} \\ \text{s.t.} \quad & T_{u,v,m,n}(t) \leq T_{\max,u,v}(t) && \text{(14b)} \\ & T_{\mathrm{g},u,v,m} \leq k_{u,v,m} T_{\mathrm{move},u,v} && \text{(14c)} \\ & \sum_{n=1}^{N} \alpha_{u,v,n}(t) = 1 && \text{(14d)} \\ & \sum_{m=1}^{N_{u,v}} \beta_{u,v,m} = 1 && \text{(14e)} \\ & \sum_{u=1}^{U} \sum_{v=1}^{V} \alpha_{u,v,n}(t) f_{u,v,n}(t) \leq F_n(t) && \text{(14f)} \\ & \sum_{u=1}^{U} \sum_{v=1}^{V} \alpha_{u,v,n}(t) W_{u,v,m} \leq B_{\mathrm{S},m}, \quad m \in \mathcal{M}_\mathrm{S} && \text{(14g)} \\ & \sum_{u=1}^{U} \sum_{v=1}^{V} \alpha_{u,v,n}(t) W_{u,v,m} \leq B_{\mathrm{C},m}, \quad m \in \mathcal{M}_\mathrm{C} && \text{(14h)} \end{aligned}$$

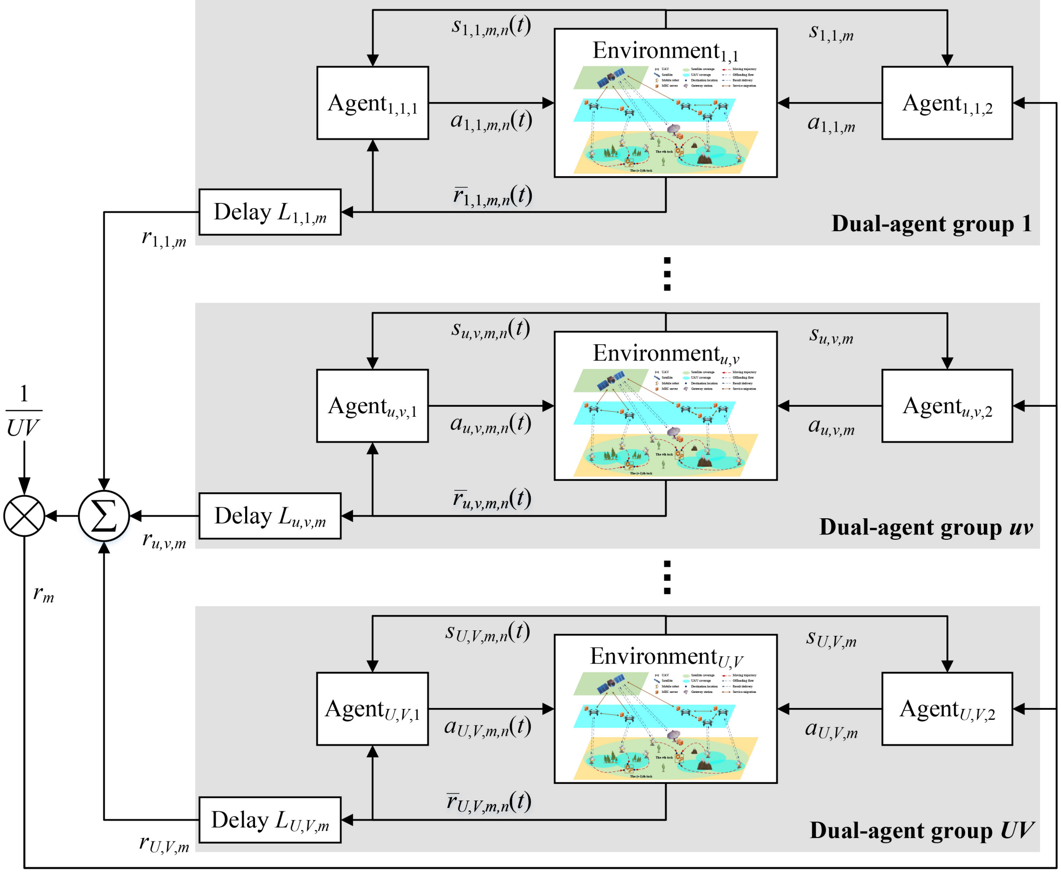


FIGURE 2. Framework of multi-group dual-agent *Q*-learning.

where $k_{u,v,m} = c_m / \sum_{m=1}^{N_{u,v}} c_m$ denotes the ratio of the robot's moving distance in the mth AP coverage region to its moving distance in the whole trip. Since the single-robot optimization problem has been proved NP-hard in [34], it can be inferred that the multi-robot optimization problems in (13) and (14) are also NP-hard.

B. REINFORCEMENT LEARNING-BASED OPTIMIZATION ALGORITHM

As shown in Fig. 2, we propose a multi-agent Q-learning algorithm composed of multi-group dual-agent Q-learning, where each group of dual agents corresponds to a robot. In the optimization problems (13) and (14), resource allocation and policy decision-making involve two timescales: 1) time slot-based offloading decision and computational resource allocation; 2) AP coverage region-based velocity control and communication resource allocation. Therefore, in our improved Q-learning algorithm, we design two different subagents, namely offloading subagents Agent$_{u,v,1}$ and velocity-control subagents Agent$_{u,v,2}$.

Based on the Markov decision process (MDP), the state, reward, and action for these two subagents are formulated as follows.

1) OFFLOADING-ORIENTED *Q*-LEARNING

The state includes: the current AP coverage region, the availability of wireless communication, the generated data size, the size of computational results, the available computational resources of the mobile robot, the velocity of the mobile robot, the selected MEC server in the previous slot, the available computational resources of all MEC servers. We have

$$s_{u,v,m,n}(t) = \big\{\mathrm{AP}_{u,v,t}, \bar{\beta}_{u,v}, D_{u,v}(t), \bar{D}_{u,v}(t), f_{\mathrm{local},u,v}(t), v_{u,v,m}(t), \mathrm{M}_{u,v,t-1}, \mathbf{F}(t)\big\}, \tag{15}$$

where $\mathrm{AP}_{u,v,t} \in \mathcal{N}_{uv1} \cup \mathcal{N}_{uv2}$ and $\bar{\beta}_{u,v} = \beta_{u,v} + 1$ for UAV/satellite communications and $\bar{\beta}_{u,v} = 0$ for local computation. According to (15), the state space size of each subagent is $3ND\bar{D}F_1V_1NF_2^N$, where D, $\bar{D}$, F_1, V_1, and F_2 are the space sizes of offloading data, computational result, local computational resource, velocity, computational resource of all MEC servers, respectively. Compared to the state space size $(3ND\bar{D}F_1V_1NF_2)^N$ in a single-agent Q-learning for the optimization problem (13), such as the dual-agent Q-learning in [34], the design in (15) has significantly reduced the space size. However, observing the available computational resources of all MEC servers still requires an exponential state space. By reducing this state to the available computational resources of the selected MEC server in the previous slot, the state space size of each subagent is reduced to $3ND\bar{D}F_1V_1NF_2$. Thus, (15) is reduced to

$$s_{u,v,m,n}(t)=\big\{\mathrm{AP}_{u,v,t},\bar{\beta}_{u,v},D_{u,v}(t),\bar{D}_{u,v}(t),f_{\mathrm{local},u,v}(t),\\ v_{u,v,m}(t),\mathrm{M}_{u,v,t-1},F_n(t-1)\big\}. \tag{16}$$

The action includes: the offloading decision and the allocated computational resource of the MEC server, which is expressed as

$$a_{u,v,m,n}(t)=\big\{\alpha_{u,v,n}(t),f_{u,v,n}(t)\big\}. \tag{17}$$

The instantaneous reward consists of the offloading reward and the penalty of the moving time larger than $k_{u,v,m}T_{\mathrm{move},u,v}$. When a robot moves at some target velocity, its moving time is less than or equal to the maximal moving time. All possible target velocities have the same penalty. The difference between the moving time $T^*_{u,v,m}$ and the maximal moving time $T_{\mathrm{move},u,v}$ is considered the moving penalty unit, that is $T^*_{u,v,m}-k_{u,v,m}T_{\mathrm{move},u,v}$. Thus, the moving penalty in each offloading slot is expressed as $\max\{\frac{T^*_{u,v,m}-k_{u,v,m}T_{\mathrm{move},u,v}}{L_{u,v,m}},0\}$. Based on (9), this instantaneous penalty can be simplified into $\max\{\Delta T-\frac{k_{u,v,m}}{L_{u,v,m}}T_{\mathrm{move},u,v},0\}$. When the elements of all Q tables are initialized to zero, the rewards of legal actions should be greater than zero to avoid the zero-valued reward caused by the local computation when wireless communications are available. Therefore, the instantaneous reward is designed as

$$r_{u,v,m,n}(t)=(1-\theta)\exp\left(1-\frac{T_{u,v,m,n}(t)}{T_{\max,u,v}(t)}\right)\\ +\theta\exp\left(1-\frac{\max\left\{\Delta T-\frac{k_{u,v,m}}{L_{u,v,m}}T_{\mathrm{move},u,v},0\right\}}{\frac{k_{u,v,m}}{L_{u,v,m}}\left(T_{\mathrm{slow},u,v}-T_{\mathrm{move},u,v}\right)}\right), \tag{18}$$

where θ $(0<\theta<1)$ is a preference factor and $T_{\mathrm{slow},u,v}=\sum_{m=1}^{N_{u,v}}\frac{c_m}{v_{\min}}$ is the total moving time when the robot moves at the lowest velocity $v_{\min}$.

Finally, when a legal action is executed, the reward is calculate by (17). Otherwise, when an illegal action is executed, the reward is set to -1. The reward update rule is expressed as

$$\bar{r}_{u,v,m,n}(t)=\begin{cases} r_{u,v,m,n}(t), & \text{legal action},\\ -1, & \text{illegal action}.\end{cases} \tag{19}$$

2) VELOCITY CONTROL-ORIENTED *Q*-LEARNING

When a mobile robot switches between adjacent AP coverage regions, the subagent $\mathrm{Agent}_{u,v,2}$ employs Q-learning to obtain the policies of velocity control, communication mode, and bandwidth resource allocation. The state, action, and reward are designed as follows.

The State Includes: the AP coverage region in the previous slot, the current AP coverage region, the initial velocity in the current AP coverage region, the available bandwidth in the satellite communication, and the available bandwidth in the UAV communication. We have

$$s_{u,v,m}=\{\mathrm{AP}_{u,v,t-1},\mathrm{AP}_{u,v,t},v_{u,v,m}(t_0),B_{\mathrm{S},m},B_{\mathrm{C},m}\}. \tag{20}$$

The Action Includes: the target velocity, communication mode, the bandwidth allocation for satellite and UAV communications, which is expressed as

$$a_{u,v,m}=\{v^*_{u,v,m},\beta_{u,v,m},W_{u,v,m}\}. \tag{21}$$

The reward r_m is the average of the accumulated rewards $r_{u,v,m}$ of all robots, where the accumulated reward is defined as the sum of all instantaneous rewards for the mobile robot in its current AP coverage region. Thus, r_m is expressed as

$$\begin{aligned} r_m &= \frac{1}{UV}\sum_{u=1}^{U}\sum_{v=1}^{V}\sum_{n=1}^{N}\sum_{t=1}^{L_{u,v,m}} r_{u,v,m,n}(t)\\ &= \frac{1}{UV}\sum_{u=1}^{U}\sum_{v=1}^{V} r_{u,v,m}. \end{aligned} \tag{22}$$

When an illegal action is executed, we set $r_m=-1$.

The detailed algorithm is shown in Algorithm 1, where $t'=t+1$ and $m'=m+1$. T_{Epi} represents the maximum number of episodes. In reinforcement learning, an episode is characterized by the system taking steps until it achieves the goal state or reaches a maximum number of steps. Here, a step is considered as a single state-action-reward pair. In each episode of Algorithm 1, all robots continue to take state-action-reward steps from their initial locations until they arrive at their target locations. The discrete-time sets are configured for Q-learning. Thus, the available satellite communication bandwidth, the available UAV communication bandwidth, the available MEC computation resource, and the local computation resource are expressed as $B_{\mathrm{S},m}\in\{\Delta W_{\mathrm{S}},2\Delta W_{\mathrm{S}},\ldots,F_{\mathrm{S}}\Delta W_{\mathrm{S}}\}$, $B_{\mathrm{C},m}\in\{\Delta W_{\mathrm{C}},2\Delta W_{\mathrm{C}},\ldots,F_{\mathrm{C}}\Delta W_{\mathrm{C}}\}$, $F_n(t)\in\{\Delta f,2\Delta f,\ldots,F_{\mathrm{MEC}}\Delta f\}$, and $f_{\mathrm{local},u,v}(t)\in\{\Delta\bar{f},2\Delta\bar{f},\ldots,F_{\mathrm{local}}\Delta\bar{f}\}$, respectively. ΔW_{S}, ΔW_{C}, Δf, and $\Delta\bar{f}$ denote the satellite communication bandwidth unit, the UAV communication bandwidth unit, the MEC computation resource unit, and the local computation resource unit, respectively. The constants of F_{S}, F_{C}, F_{MEC}, and F_{local} are defined as the maximum numbers of units for satellite bandwidth, UAV bandwidth, MEC computation resources, and local computation resources, respectively. In addition, to address the issue of slow convergence caused by numerous illegal actions in multi-agent Q-learning with multiple states and actions, two algorithms are presented in Algorithms 2 and 3, which facilitate the selection of legal actions for communication resource allocation and computational resource allocation, respectively. Furthermore, upon incorporating the reward design for illegal actions in (19), the improved convergence of the proposed algorithm implies effective stability. This is the reason that the Lyapunov functions are not considered for the optimization problem in this paper.

Algorithm 1 Joint Offloading and Velocity Control Based on the Multi-Group Dual-Agent Q-Learning Algorithm

```
Input: Initialize the table entry $Q_{u,v,1}(s,a) = 0$ and
    $Q_{u,v,2}(s,a) = 0$, velocity $v_{u,v,m}(t)$, moving distance $c_m$, available
    satellite communication bandwidth $B_{\mathrm{S},m}$, available UAV
    communication bandwidth $B_{\mathrm{C},m}$, available MEC computation
    resource $F_n(t)$, local computation resource $f_{\mathrm{local},u,v}(t)$, learning
    rate $\lambda_{u,v}$, greedy factor $\varepsilon_{u,v}$, discount factor $\gamma_{u,v}$.
Output: Offloading decision $\alpha_{u,v,n}(t)$, computational resource
    allocation $f_{u,v,n}(t)$, target velocity $v^*_{u,v,m}$, communication mode
    $\beta_{u,v,m}$, bandwidth allocation $W_{u,v,m}$.
1:  for $j = 0, 1, 2, \ldots, T_{\mathrm{Epi}}$ do
2:    Reset $m_{u,v} = 0$, $s_{u,v,m,n}(t)$, $s_{u,v,m}$, and $N_{\mathrm{rob}} = UV$;
3:    while $\bigcup_{u,v} I\{m_{u,v} \le N_{u,v}\}$ do
4:      if $\mathrm{AP}_{u,v,t-1} \ne \mathrm{AP}_{u,v,t}$ then
5:        Observe state $s_{u,v,m}$;
6:        Chose the legal action $a_{u,v,m}$ based on Algorithm 2;
7:        Calculate reward $r_m$ and next state $s_{u,v,m+1}$;
8:        Update the $Q$-table for velocity control and communication resource allocation:
          $Q_{u,v,2}\big(s_{u,v,m}, a_{u,v,m}\big)$
          $= \big(1-\lambda_{u,v}\big)Q_{u,v,2}\big(s_{u,v,m}, a_{u,v,m}\big) + \lambda_{u,v}$
          $\times \big(r_m + \gamma_{u,v} \max\big\{Q_{u,v,2}\big(s_{u,v,m'}, a_{u,v,m'}\big)\big\}\big)$   (23)
9:        Update state $s_{u,v,m} = s_{u,v,m'}$;
10:       $m_{u,v} = m_{u,v} + 1$;
11:     end if
12:     Observe state $s_{u,v,m,n}(t)$;
13:     Chose the legal computational resource allocation $f_{u,v,n}(t)$ based on Algorithm 3;
14:     if $m_{u,v} \ge N_{u,v} + 1$ then
15:       $N_{\mathrm{rob}} = N_{\mathrm{rob}} - 1$ and $r_{u,v,m,n}(t) = 0$;
16:     else
17:       Calculate reward $r_{u,v,m,n}(t)$ and next state $s_{u,v,m,n}(t')$;
18:       Update the $Q$-table for data offloading and computational resource allocation:
          $Q_{u,v,1}\big(s_{u,v,m,n}(t), a_{u,v,m,n}(t)\big)$
          $= \big(1-\lambda_{u,v}\big)Q_{u,v,1}\big(s_{u,v,m,n}(t), a_{u,v,m,n}(t)\big)$
          $+ \lambda_{u,v}\big(r_{u,v,m,n}(t) + \gamma_{u,v}$
          $\times \max\big\{Q_{u,v,1}\big(s_{u,v,m,n}(t'), a_{u,v,m,n}(t')\big)\big\}\big)$   (24)
19:       Update state $s_{u,v,m,n}(t) = s_{u,v,m,n}(t')$;
20:     end if
21:   end while
22: end for
```

In addition, we present a potential practical application of the proposed algorithm. According to the MEC system reference architecture in [38], [39], the proposed algorithm will operate at the MEC orchestrator. This orchestrator can monitor the deployed MEC hosts and the available network resources, ultimately selecting the most appropriate MEC host for data offloading and service migration. It can also allocate appropriate radio communication resources to the satellite-robot and UAV-robot links for data transmission.

Algorithm 2 Legal Actions of Bandwidth Resource Allocation

```
Output: Communication mode $\beta_{u,v,m}$, legal bandwidth allocation $W_{u,v,m}$.
1: Release the bandwidth resources of the previous AP;
2: Observe states $s_{u,v,m}$ of all robots;
3: Chose action $a_{u,v,m}$ with $\varepsilon$-greedy algorithm;
4: if $\left(\sum_{u=1}^{U}\sum_{v=1}^{V} \alpha_{u,v,n}(t)W_{u,v,m} \le B_{\mathrm{S},m}, m \in \mathcal{M}_{\mathrm{S}}\right) \cap \left(\sum_{u=1}^{U}\sum_{v=1}^{V} \alpha_{u,v,n}(t)W_{u,v,m} \le B_{\mathrm{C},m}, m \in \mathcal{M}_{\mathrm{C}}\right)$ then
5:   Output the legal action;
6: else
7:   Set $Q_{u,v,2}(s_{u,v,m}, a_{u,v,m}) = -1$ due to the illegal action;
8:   Go back to Step 3.
9: end if
```

Algorithm 3 Legal Actions of Offloading Decision and Computational Resource Allocation

```
Output: Offloading decision $\alpha_{u,v,n}(t)$, computational resource allocation $f_{u,v,n}(t)$.
1: Chose action $a_{u,v,m,n}(t)$ with $\varepsilon$-greedy algorithm;
2: if $\sum_{u=1}^{U}\sum_{v=1}^{V} \alpha_{u,v,n}(t)f_{u,v,n}(t) \le F_n(t)$ then
3:   Output the legal action;
4: else
5:   Set $r_{u,v,m,n}(t) = -1$ due to the illegal action;
6:   $Q_{u,v,1}(s_{u,v,m,n}(t), a_{u,v,m,n}(t)) = r_{u,v,m,n}(t)$;
7:   Go back to Step 1.
8: end if
```

IV. CONVERGENCE ANALYSIS

To analyze the convergence of the multi-group dual-agent Q-learning, according to [40], we first present the following lemma.

Lemma 1: Given a non-negative learning rate sequence $\{\lambda_j\}$ with $0 \le \lambda_j \le 1$, when $\lim\limits_{T_{\mathrm{Epi}}\to\infty} \sum_{j=0}^{T_{\mathrm{Epi}}} \lambda_j \to \infty$, for $0 \le \phi < \infty$, we have

$$\prod_{j=0}^{\infty}\left(1 - \frac{\lambda_j}{\phi+1}\right) = 0. \tag{25}$$

Proof: See Appendix A. ■

Then, based on [40], a unified form of updating Q tables for (23) and (24) is expressed as

$$Q_j(s_t,a_t) = \big(1-\lambda_j\big)Q_j(s_t,a_t) + \lambda_j\left(r(s_t,a_t) + \gamma \max_{a_{t'}\in\mathcal{A}}\big\{Q_j(s_{t'},a_{t'})\big\}\right), \tag{26}$$

where $j = 0, 1, \ldots, T_{\mathrm{Epi}}$ and $\mathcal{A}$ is the action space. Based on Lemma 1, we derive the following theorem.

Theorem 1: Let the optimal Q value of $Q_j(s_t,a_t)$ in a given state space be $Q^*(s_t,a_t) = r(s_t,a_t) +$

$\gamma \max\{Q^*(s_{t'}, a_{t'})\}$. The initialized Q function $Q_0(s_t, a_t)$ is a semi-positive definite function. When the learning rate satisfies $0 \le \lambda_j \le 1$ and $\lim\limits_{T_{\text{Epi}}\to\infty} \sum_{j=0}^{T_{\text{Epi}}} \lambda_j = \infty$, for $j \to \infty$, $Q_j(s_t, a_t)$ can converge to the optimal value $Q^*(s_t, a_t)$, which is given as

$$\lim_{j\to\infty} Q_j(s_t, a_t) = Q^*(s_t, a_t). \tag{27}$$

Proof: See Appendix B. ■

It is implied in Theorem 1 that based on the local state observation, the Q-learning algorithms in (23) and (24) can converge to a local optimal solution.

Next, we analyze the factors that affect the convergence of Q values to the optimal Q values. According to [41], the following convergence of (23) and (24) is achieved.

Theorem 2: We assume that $Q^*_{u,v,1}$ and $Q^*_{u,v,2}$ are the optimal Q values of $Q_{u,v,1,j}(s_{u,v,m,n}(t), a_{u,v,m,n}(t))$ and $Q_{u,v,2,j}(s_{u,v,m}, a_{u,v,m})$, respectively. We also assume that $\|Q_{u,v,1,0}(s_{u,v,m,n}(t), a_{u,v,m,n}(t))\| \le \frac{r_{u,v,\max}}{1-\gamma}$ and $\|Q_{u,v,2,0}(s_{u,v,m}, a_{u,v,m})\| \le \frac{L_{u,v,\max} r_{u,v,\max}}{1-\gamma}$. Then, we obtain

$$\left\|Q_{u,v,1,j}\big(s_{u,v,m,n}(t), a_{u,v,m,n}(t)\big)\right\| \le \frac{r_{u,v,\max}}{1-\gamma}, \tag{28}$$

$$\left\|Q_{u,v,1,j}\big(s_{u,v,m,n}(t), a_{u,v,m,n}(t)\big) - Q^*_{u,v,1}\right\| \le \frac{2r_{u,v,\max}}{1-\gamma}, \tag{29}$$

$$\left\|Q_{u,v,2,j}\big(s_{u,v,m}, a_{u,v,m}\big)\right\| \le \frac{L_{u,v,\max} r_{u,v,\max}}{1-\gamma}, \tag{30}$$

$$\left\|Q_{u,v,2,j}\big(s_{u,v,m}, a_{u,v,m}\big) - Q^*_{u,v,2}\right\| \le \frac{2L_{u,v,\max} r_{u,v,\max}}{1-\gamma}, \tag{31}$$

where $r_{u,v,\max} = \max_{m,n}\{\|\bar{r}_{u,v,m,n}(t)\|\}$, $L_{u,v,\max} = \max_m\{L_{u,v,m}\}$, $j = 0, 1, \ldots, T_{\text{Epi}}$, $u = 1, 2, \ldots, U$, $v = 1, 2, \ldots, V$, $m = 1, 2, \ldots, N_{u,v}$, and $n = 1, 2, \ldots, N$.

Proof: See Appendix C. ■

Theorem 2 indicates that the reduced discount factor γ can improve the convergence performance of Q-learning in (23) and (24).

According to [42], we assume that the action-value function of the single Q-learning for the optimization problem (14) can be approximately linearly decomposed into the value functions of multiple agents as follows

$$\begin{aligned} Q((\mathbf{s}(t), \mathbf{s}_m), (\mathbf{a}(t), \mathbf{a}_m)) &\approx \sum_{u=1}^{U}\sum_{v=1}^{V} Q_{u,v,2}\big(s_{u,v,m}, a_{u,v,m}\big) \\ &+ \sum_{u=1}^{U}\sum_{v=1}^{V} \Big(Q_{u,v,1}\big(s_{u,v,m,n}(t), a_{u,v,m,n}(t)\big)\Big), \end{aligned} \tag{32}$$

where $\mathbf{s}(t) = \{s_{u,v,m,n}(t)\}$, $\mathbf{s}_m = \{s_{u,v,m}\}$, $\mathbf{a}(t) = \{a_{u,v,m,n}(t)\}$, and $\mathbf{a}_m = \{a_{u,v,m}\}$ for $u = 1, 2, \ldots, U$, $v = 1, 2, \ldots, V$, $m = 1, 2, \ldots, N_{u,v}$, $n = 1, 2, \ldots, N$. In Algorithm 1, $Q_{u,v,2}(s_{u,v,m}, a_{u,v,m})$ is updated by the accumulated rewards of Agent$_{u,v,1}$, while $Q_{u,v,1}(s_{u,v,m,n}(t), a_{u,v,m,n}(t))$ is updated by the actions of Agent$_{u,v,2}$. Based on this, each subagent can be deployed in a distributed manner, and centralized learning can be carried out by the accumulation of reward values.

According to Theorem 2, it can be derived that

$$\begin{aligned} &\|Q((\mathbf{s}(t), \mathbf{s}_m), (\mathbf{a}(t), \mathbf{a}_m))\| \\ &\le \sum_{u=1}^{U}\sum_{v=1}^{V} \Big(\big\|Q_{u,v,1}\big(s_{u,v,m,n}(t), a_{u,v,m,n}(t)\big)\big\| \\ &\quad + \big\|Q_{u,v,2}\big(s_{u,v,m}, a_{u,v,m}\big)\big\|\Big) \\ &\le \sum_{u=1}^{U}\sum_{v=1}^{V} \big(1 + L_{u,v,\max}\big)\frac{r_{u,v,\max}}{1-\gamma}. \end{aligned} \tag{33}$$

Therefore, the reduced discount factor γ can improve the convergence performance of Algorithm 1.

To further analyze other factors affecting the convergence of Algorithm 1, the following theorem is obtained.

Theorem 3: The convergence of the proposed multi-group dual-agent Q-learning is expressed as

$$\begin{aligned} &E\Big\{\big\|Q_{u,v,1,T_{\text{Epi}}}\big(s_{u,v,m,n}(t), a_{u,v,m,n}(t)\big) - Q^*_{u,v,1}\big\|_\infty\Big\} \\ &\le \sum_{j=0}^{T_{\text{Epi}}} \gamma^{T_{\text{Epi}}-j}\sqrt{\theta_{u,v,1,j}} + \gamma^{T_{\text{Epi}}} \\ &\quad \times E\Big\{\big\|Q_{u,v,1,0}\big(s_{u,v,m,n}(t), a_{u,v,m,n}(t)\big) - Q^*_{u,v,1}\big\|_\infty\Big\}, \end{aligned} \tag{34}$$

$$\begin{aligned} &E\Big\{\big\|Q_{u,v,2,T_{\text{Epi}}-1}\big(s_{u,v,m}, a_{u,v,m}\big) - Q^*_{u,v,2}\big\|_\infty\Big\} \\ &\le \sum_{j=0}^{T_{\text{Epi}}} \gamma^{T_{\text{Epi}}-j}\sqrt{L_{u,v,\max}\theta_{u,v,1,j}} \\ &\quad + \gamma^{T_{\text{Epi}}} E\Big\{\big\|Q_{u,v,2,0}\big(s_{u,v,m}, a_{u,v,m}\big) - Q^*_{u,v,2}\big\|_\infty\Big\}. \end{aligned} \tag{35}$$

When $\theta_{u,v,1,j} = \theta_{u,v}$, for $T_{\text{Epi}} \to \infty$, Eqs. (34) and (35) are reduced to

$$\begin{aligned} &E\Big\{\big\|Q_{u,v,1,T_{\text{Epi}}}\big(s_{u,v,m,n}(t), a_{u,v,m,n}(t)\big) - Q^*_{u,v,1}\big\|_\infty\Big\} \\ &\le \frac{\sqrt{\theta_{u,v}}}{1-\gamma} + \gamma^{T_{\text{Epi}}} \\ &\quad \times E\Big\{\big\|Q_{u,v,1,0}\big(s_{u,v,m,n}(t), a_{u,v,m,n}(t)\big) - Q^*_{u,v,1}\big\|_\infty\Big\}, \end{aligned} \tag{36}$$

$$\begin{aligned} &E\Big\{\big\|Q_{u,v,2,T_{\text{Epi}}}\big(s_{u,v,m}, a_{u,v,m}\big) - Q^*_{u,v,2}\big\|_\infty\Big\} \\ &\le \frac{\sqrt{L_{u,v,\max}\theta_{u,v}}}{1-\gamma} \\ &\quad + \gamma^{T_{\text{Epi}}} E\Big\{\big\|Q_{u,v,2,0}\big(s_{u,v,m}, a_{u,v,m}\big) - Q^*_{u,v,2}\big\|_\infty\Big\}. \end{aligned} \tag{37}$$

Proof: See Appendix D. ■

Theorem 3 reveals that the convergence of Algorithm 1 is affected by four aspects: 1) the approximation error $\theta_{u,v,1,j}$ of Agent$_{u,v,1}$; 2) the approximation error $\theta_{u,v,2,j}$ of Agent$_{u,v,2}$; 3) the Bellman operation for $Q_{u,v,1,j}$; and 4) the Bellman operation for $Q_{u,v,2,j}$. It is also concluded in Theorem 3 that with reduced approximation errors (or discount factor γ), the convergence performance of Algorithm 1 can be enhanced.

In addition, according to [43], [44], [45], the effect of the learning rate λ and greedy factor ε on the convergence of the multi-group dual-agent Q-learning algorithm is analyzed below.

In Agent$_{u,v,1}$, we assume that all robots have the same state space $\mathcal{S}_1$ and action space $\mathcal{A}_1$. By removing the subscripts u and v, (24) is simplified to

$$\begin{aligned} &Q_1\big(s_{m,n}(t'), a_{u,v,m,n}(t')\big) \\ &= Q_1\big(s_{m,n}(t), a_{m,n}(t)\big) + \lambda\Big(r_{m,n}(t) - Q_1\big(s_{m,n}(t), a_{m,n}(t)\big) \\ &\quad + \gamma \max_{a_{m,n}(t')\in\mathcal{A}_1} \big\{Q_1\big(s_{m,n}(t'), a_{m,n}(t')\big)\big\}\Big), \end{aligned} \tag{38}$$

where $t' = t+1$. Let $\Delta Q_1(s_{m,n}(t), a_{m,n}(t)) = Q_1(s_{m,n}(t'), a_{m,n}(t')) - Q_1(s_{m,n}(t), a_{m,n}(t))$. According to [44], [46], the dynamics of Q functions with multiple states can be expressed as

$$\begin{aligned} &E\big\{\Delta Q_1\big(s_{m,n}(t), a_{m,n}(t)\big)\big\} \\ &= p\big(s_{m,n}(t)\big)p\big(a_{m,n}(t)\big) \\ &\quad \times \lambda\Big(E\big\{r_{m,n}(t)\big\} - Q_1\big(s_{m,n}(t), a_{m,n}(t)\big) \\ &\quad + \gamma \sum_{s_{m,n}(t')\in\mathcal{S}_1} p\big(s_{m,n}(t')\big|s_{m,n}(t), a_{m,n}(t)\big) \\ &\quad \times \max_{a_{m,n}(t')\in\mathcal{A}_1} \big\{Q_1\big(s_{m,n}(t'), a_{m,n}(t')\big)\big\}\Big), \end{aligned} \tag{39}$$

where $p(s_{m,n}(t))$ is the probability of state $s_{m,n}(t)$ in the tth slot, $p(s_{m,n}(t')|s_{m,n}(t), a_{m,n}(t))$ represents the transition probability from state $s_{m,n}(t)$ to state $s_{m,n}(t')$ with action $a_{m,n}(t)$, and $p(a_{m,n}(t))$ is the probability of executing action $a_{m,n}(t)$, i.e.,

$$p\big(a_{m,n}(t)\big) = \begin{cases} 1-\varepsilon+\frac{\varepsilon}{|\mathcal{A}_1|}, & \text{for}\max\big\{Q_1\big(s_{m,n}(t), a_{m,n}(t)\big)\big\}, \\ \frac{\varepsilon}{|\mathcal{A}_1|}, & \text{otherwise}, \end{cases} \tag{40}$$

where $|\mathcal{A}_1|$ denotes the size of the set $\mathcal{A}_1$. Taking the limit of (39) yields

$$\begin{aligned} &\frac{dQ_1\big(s_{m,n}(t), a_{m,n}(t)\big)}{dt} \\ &= p\big(s_{m,n}(t)\big)p\big(a_{m,n}(t)\big) \\ &\quad \times \lambda\Big(E\big\{r_{m,n}(t)\big\} - Q_1\big(s_{m,n}(t), a_{m,n}(t)\big) \\ &\quad + \gamma \sum_{s_{m,n}(t')\in\mathcal{S}_1} p\big(s_{m,n}(t')\big|s_{m,n}(t), a_{m,n}(t)\big) \\ &\quad \times \max_{a_{m,n}(t')\in\mathcal{A}_1} \big\{Q_1\big(s_{m,n}(t'), a_{m,n}(t')\big)\big\}\Big). \end{aligned} \tag{41}$$

Under the assumption of $p(s_{m,n}(t)) = \frac{1}{|\mathcal{S}_1|}$, for $\gamma \to 0$, (41) can be simplified to

$$\begin{aligned} \frac{dQ_1\big(s_{m,n}(t), a_{m,n}(t)\big)}{dt} = \frac{\lambda p\big(a_{m,n}(t)\big)}{|\mathcal{S}_1|}\Big(E\big\{r_{m,n}(t)\big\} \\ - Q_1\big(s_{m,n}(t), a_{m,n}(t)\big)\Big). \end{aligned} \tag{42}$$

When $E\{r_{m,n}(t)\}$ is a constant, solving the first-order differential equation of (42) yields

$$\begin{aligned} &Q_1\big(s_{m,n}(t), a_{m,n}(t)\big) \\ &= \frac{1}{y(t)}\Bigg(E\big\{r_{m,n}(t)\big\}\int_0^t \frac{\lambda}{|\mathcal{S}_1|}y(l)p\big(a_{m,n}(l)\big)dl \\ &\quad + y(0)Q_1\big(s_{m,n}(0), a_{m,n}(0)\big)\Bigg), \end{aligned} \tag{43}$$

where $y(t) = \exp(\int_0^t \frac{\lambda}{|\mathcal{S}_1|}p(a_{m,n}(l))dl)$. If the random action selection occurs in interval $[0, t_0]$ (corresponding to $t_0/\Delta T$ times in discrete time), we have $y(t) = e^{\frac{\lambda}{|\mathcal{S}_1|}((\frac{\varepsilon}{|\mathcal{A}_1|})t_0+(1-\varepsilon+\frac{\varepsilon}{|\mathcal{A}_1|})(t-t_0))}$. Thus, (43) can be expressed as

$$Q_1\big(s_{m,n}(t), a_{m,n}(t)\big) = Ye^{-\frac{\lambda}{|\mathcal{S}_1|}\left(1-\frac{|\mathcal{A}_1|-1}{|\mathcal{A}_1|}\varepsilon\right)t} + E\big\{r_{m,n}(t)\big\}, \tag{44}$$

where $Y = y(0)Q_1(s_{m,n}(0), a_{m,n}(0)) - \frac{E\{r_{m,n}(t)\}}{1-\varepsilon+\frac{\varepsilon}{|\mathcal{A}_1|}}(\frac{\varepsilon}{|\mathcal{A}_1|} + (1-\varepsilon)e^{\frac{\lambda}{|\mathcal{S}_1|}(1-\varepsilon+\frac{\varepsilon}{|\mathcal{A}_1|})t_0})$. Taking the limit of (44) yields

$$\begin{aligned} &\lim_{t\to\infty} Q_1\big(s_{m,n}(t), a_{m,n}(t)\big) \\ &\quad = \lim_{t\to\infty} Ye^{-\frac{\lambda}{|\mathcal{S}_1|}\left(1-\frac{|\mathcal{A}_1|-1}{|\mathcal{A}_1|}\varepsilon\right)t} + \lim_{t\to\infty} E\big\{r_{m,n}(t)\big\} \\ &\quad = E\big\{r_{m,n}(t)\big\}. \end{aligned} \tag{45}$$

It is concluded from (44) and (45) that, when $|\mathcal{S}_1|$ and $|\mathcal{A}_1|$ are given, as ε decreases or λ increases, the convergence rate of $Q_1(s_{m,n}(t), a_{m,n}(t))$ to $E\{r_{m,n}(t)\}$ increases.

Similarly, for Agent$_{u,v,2}$, based on [44], [46], we can derive

$$\begin{aligned} \frac{dQ_2(s_m, a_m)}{dm} &= p(s_m)p(a_m)\lambda\Big(E\{r_m\} - Q_2(s_m, a_m) \\ &\quad + \gamma \sum_{s_{m'}\in\mathcal{S}_2} p(s_{m'}|s_m, a_m) \max_{a_{m'}\in\mathcal{A}_2} \{Q_2(s_{m'}, a_{m'})\}\Big), \end{aligned} \tag{46}$$

where $p(s_m)$ is the probability of state s_m, $p(s_{m'}|s_m, a_m)$ is the transition probability from state s_m to state $s_{m'}$ with action a_m, $p(a_m)$ is the probability of selecting the action a_m, i.e.,

$$p(a_m) = \begin{cases} 1-\varepsilon+\frac{\varepsilon}{|\mathcal{A}_2|}, & \text{for}\max\{Q_2(s_m, a_m)\}, \\ \frac{\varepsilon}{|\mathcal{A}_2|}, & \text{otherwise}. \end{cases} \tag{47}$$

Thus, under the consumption of $p(s_m) = \frac{1}{|\mathcal{S}_2|}$ and the constant $E\{r_m\}$, when $\gamma \to 0$, solving (46) yields

$$\begin{aligned} &Q_2(s_m, a_m) \\ &\quad = \frac{1}{x_m}\Bigg(E\{r_m\}\int_0^m \frac{\lambda}{|\mathcal{S}_2|}x_l p(a_l)dl + y(0)Q_2(s_0, a_0)\Bigg), \end{aligned} \tag{48}$$

where $x_m = \exp(\int_0^m \frac{\lambda}{|\mathcal{S}_2|}p(a_l)dl)$. We assume that the random actions are selected from interval $[0, m_0]$. Then, it follows that

$$Q_2(s_m, a_m) = Xe^{-\frac{\lambda}{|\mathcal{S}_2|}\left(1-\frac{|\mathcal{A}_2|-1}{|\mathcal{A}_2|}\varepsilon\right)m} + E\{r_m\}, \tag{49}$$

where $X = x_0Q_2(s_0,a_0) - \frac{E\{r_m\}}{1-\varepsilon+\frac{\varepsilon}{|\mathcal{A}_2|}}(\frac{\varepsilon}{|\mathcal{A}_2|} + (1-\varepsilon) e^{\frac{\lambda}{|\mathcal{S}_2|}(1-\varepsilon+\frac{\varepsilon}{|\mathcal{A}_2|})m_0})$. Eqs. (48) and (49) reveal that for the given $|\mathcal{S}_2|$ and $|\mathcal{A}_2|$, with the reduced ε or the increased λ, the convergence rate of $Q_2(s_m,a_m)$ to $E\{r_m\}$ is improved.

Finally, based on the effects of ε and λ on the convergence of Agent$_{u,v,1}$ and Agent$_{u,v,2}$, it can be concluded that the reduced ε and the increased λ are beneficial to enhance the convergence rate of Algorithm 1.

V. NUMERICAL RESULTS AND PERFORMANCE COMPARISON

This section demonstrates the performance of the proposed algorithm and compares it with local execution and conventional offloading at constant moving velocity. Additionally, we compare our proposed algorithm with the single-robot algorithm in [34]. It noted that the algorithm from [34] targets the single-robot optimization without consideration of resource allocation among multiple robots. When it is employed to the multi-robot optimization, we assume that the communication resources in each AP and computation resources in each MEC server are uniformly distributed among all robots.

In our simulation, three tasks are cooperatively accomplished by 24 mobile robots. 21 APs and 21 MEC servers are deployed, including 20 UAV-based MEC servers and one satellite-based MEC server. The region only covered by the satellite is divided into five sub-regions. The computation frequencies of UAV- and satellite-based MEC servers are draw from sets $\{1, 2, 3\}$ (GHz) and $\{10, 20, 30\}$ (GHz), respectively. The moving distance c_m is randomly chosen from set $\{100, 200, 300\}$ (m) for the UAVs and from set $\{1000, 2000, 3000\}$ (m) for the satellite. In UAV communications, the available bandwidth is selected from the set $\{9, 10.8, 12.6\}$ (MHz). In satellite communications, we set the distances as $d_{\text{GS}} = d_{\text{SE}} = 1000$ km, and the transmission rates as $r_{\text{SE}} = 100$ Mbps. The available bandwidth is selected from the set $\{90, 180, 270\}$ (KHz). In the UAV and satellite communications, the transmit power, the channel noise power, and the channel gain are $p = 0.2$ W, $\sigma^2 = 2 \times 10^{-12}$ W, and $h^2 = 10^{-6}$, respectively. The extra migration cost ΔG is set to be the average delay of 500 ms. In each offloading interval $\Delta T = 1$ s, the generated data size is randomly selected from the set $\{100, 350, 600\}$ (KB) with $\Phi = 800$ CPU cycles/bit, and the size of output data is randomly selected from the set $\{50, 70, 90\}$ (KB). In local computation, the computing capacity of the robot is randomly chosen from a finite set $\{0.5, 0.7, 0.9\}$ (GHz). During the movements of all robots with $a = 2$ m/s^2, their velocities are from a finite set $\{5, 6, \ldots, 15\}$ (m/s). In Q-learning, the hyperparameters are set as $\lambda = 0.1$, $\gamma = 0.9$, and ε decreases from 0.99 to 0.01 with an exponential discount of $6.125 \times 10^{-4} e^{-6.125 \times 10^{-4} t}$.

In Fig. 3, the convergence of the proposed scheme is compared with those of the conventional offloading scheme with various velocities and the algorithm in [34]. The number of APs with unavailable wireless communication is $N_{\text{CH}} = 6$. The preference factor is set as $\theta = 0.1$. To clearly demonstrate the convergence performance, this figure plots the difference between the average reward of legal actions and the accumulated penalty of illegal actions. As can be observed from Fig. 3, at the beginning of training, the proposed scheme has a higher reward than conventional offloading at low velocity and a reward close to conventional offloading at high velocity. The algorithm in [34] has the lowest reward and the slowest convergence rate, because it does not mask out illegal actions. As the number of training episodes increases, the proposed scheme can converge effectively, which is consistent with the convergence analysis in Section IV. Furthermore, the convergence behavior of the proposed scheme is investigated in Fig. 4. In Fig. 4(a), the increased value of N_{CH} results in a reduced reward of the proposed scheme. In Fig. 4(b), the increased value of θ leads to a slightly increased reward of the proposed scheme. It is inferred that the impact on convergence is attributed more to the wireless connectivity than to the preference factor.

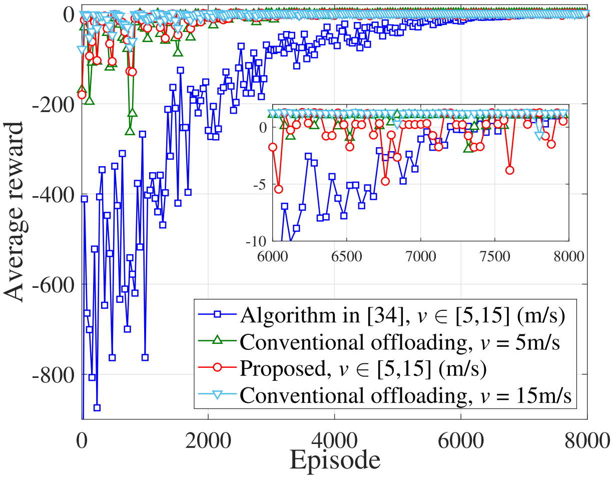


FIGURE 3. Average rewards of the conventional offloading and the proposed scheme, where $N_{\text{CH}} = 6$ and $\theta = 0.1$.

In Fig. 5, the average completion time of offloading and the average moving time of robots in conventional offloading, local execution, and the proposed schemes are plotted for different values of N_{CH}.

- First, the average completion times of the conventional and proposed schemes are compared for $N_{\text{CH}} = 2, 4, 6, 8$ and $\theta = 0.1$. As can be seen from Fig. 5(a), with the increase of N_{CH}, the algorithm in [34] has the reduced completion time, and conventional offloading and the proposed scheme have increased completion times. Compared to conventional offloading, local execution, and the algorithm in [34], the proposed scheme can reduce the average completion time by 3% to 18%. Moreover, the algorithm in [34] utilizes equal resource allocation for all robots. As a result, the computation resources of an MEC server allocated for

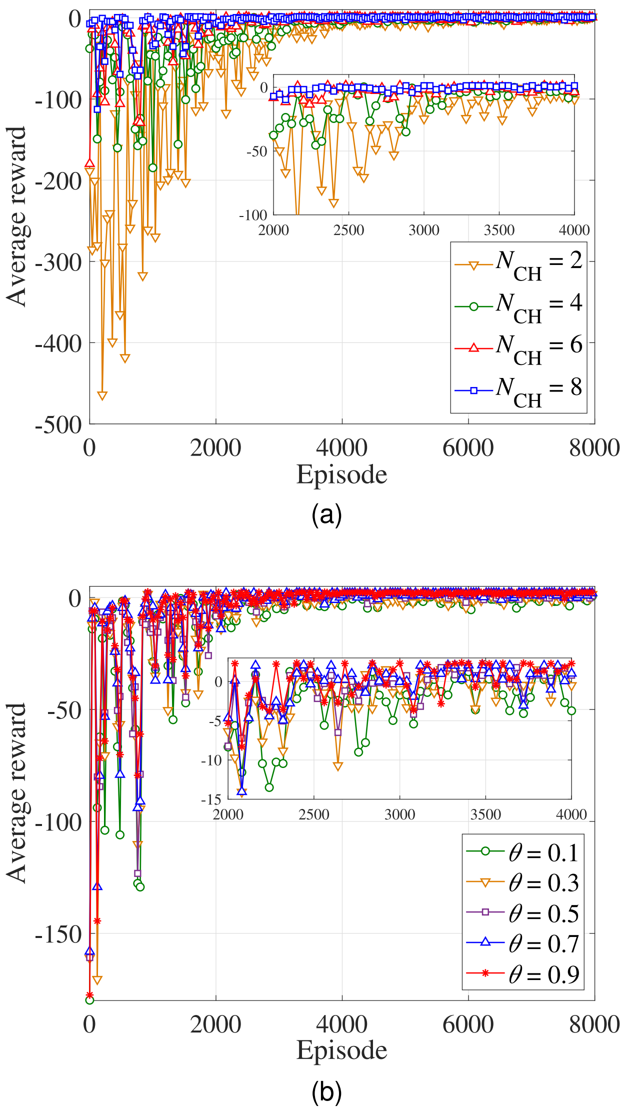


FIGURE 4. Average rewards of the proposed scheme in training. (a) N_{CH} = 2, 4, 6, 8 and $\theta = 0.1$; and (b) θ = 0.1, 0.3, 0.5, 0.7, 0.9 and $N_{CH} = 6$.

each robot may be lower than those of individual robots. The algorithm in [34] has the longest completion time.

- Second, to show the effect of velocity control of multiple robots on service latency, the completion time versus the moving time is investigated in Fig. 5(b). Two important observations can be obtained from Fig. 5(b). 1) In the absence of velocity control, conventional schemes use the minimum velocity (v = 5 m/s) and the maximum velocity (v = 15 m/s), resulting in excessively high and low moving times, respectively. Velocity control provides a more flexible approach to adjusting the moving time; 2) Velocity control is beneficial to reduce the completion time. Since a constant velocity is adopted, there is no effect of the velocity control on the completion time in local execution and a limited reduction in completion time in conventional offloading. The velocity control in [34] is related to the offloading decision, and is independent of resource allocation among multiple robots. Thus, velocity control cannot efficiently reduce the average completion time. With the aid of velocity control for data offloading and resource allocation, the proposed scheme can achieve the smallest completion time with a moderate moving time.

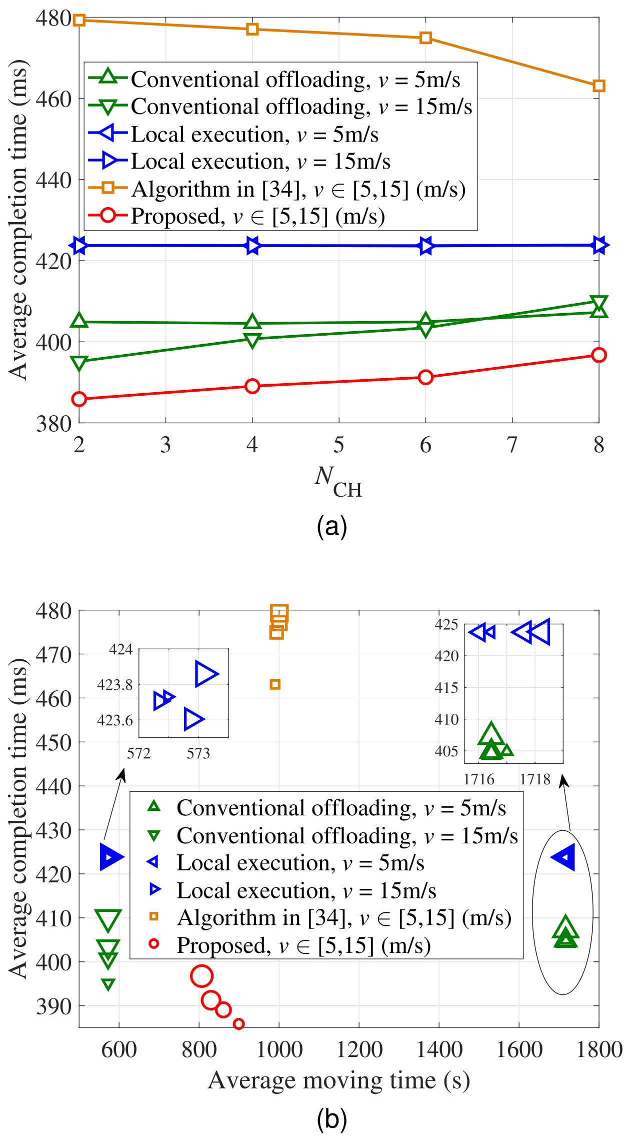


FIGURE 5. Performance comparison among local execution, conventional offloading, and the proposed scheme for N_{CH} = 2, 4, 6, 8 and $\theta = 0.1$. (a) Average completion time of offloading versus N_{CH}; and (b) Average completion time of offloading versus average moving time of robots. When the marker size increases, the value of N_{CH} increases from 2 to 8.

- It is also implied in Fig. 5 that the proposed scheme is sensitive to N_{CH} due to the communication state-based velocity control.

In Fig. 6, the average completion time and the average moving time of conventional offloading, local execution, the algorithm in [34], and the proposed scheme are plotted for $N_{CH} = 6$ and $\theta = 0.1, 0.3, 0.5, 0.7, 0.9$. It is shown

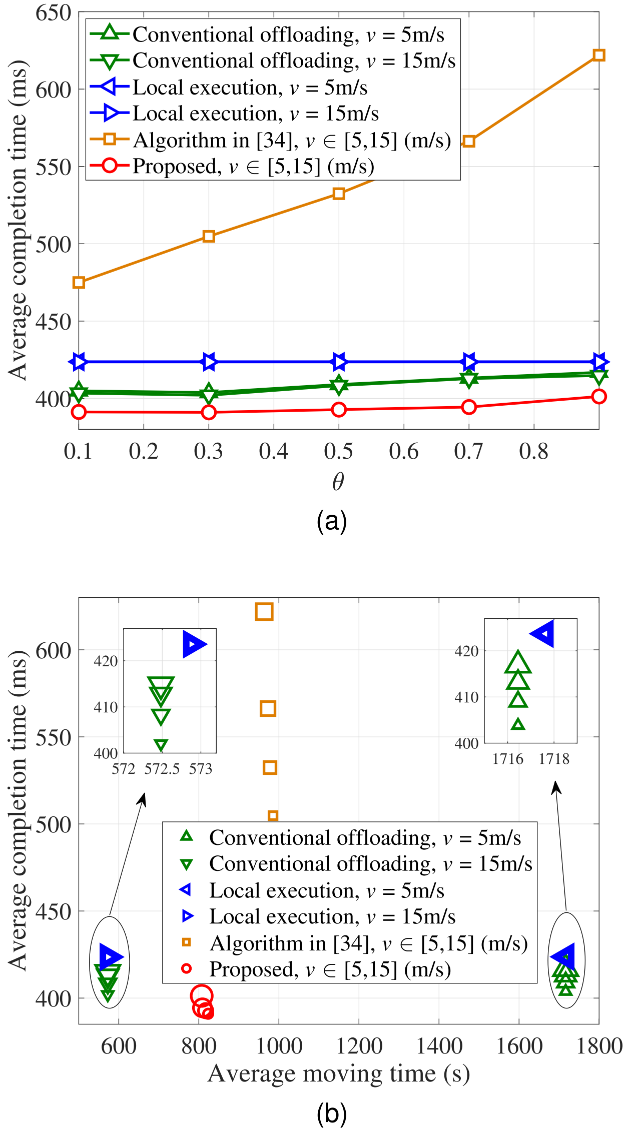


FIGURE 6. Performance comparison among local execution, conventional offloading, and the proposed scheme for $\theta = 0.1, 0.3, 0.5, 0.7, 0.9$ and $N_{\text{CH}} = 6$. (a) Average completion time of offloading versus θ; and (b) Average completion time of offloading versus average moving time of robots. When the marker size increases, the value of θ increases from 0.3 to 0.9.

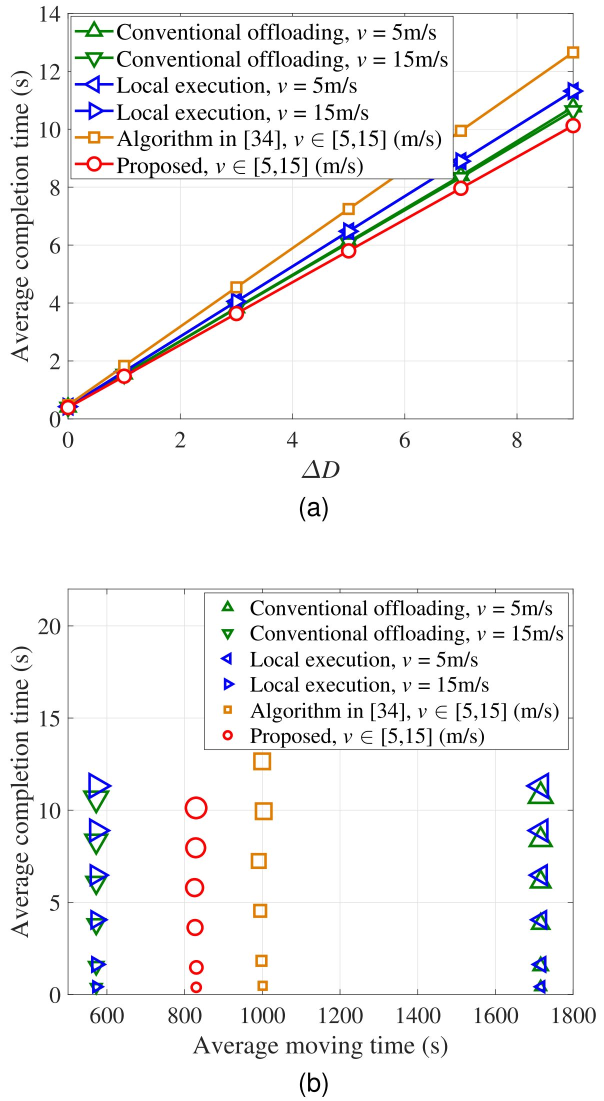


FIGURE 7. Performance comparison among local execution, conventional offloading, and the proposed scheme for $\Delta D = 0, 1, 3, 5, 7, 9$, $N_{\text{CH}} = 6$ and $\theta = 0.1$. (a) Average completion time of offloading versus ΔD; and (b) Average completion time of offloading versus average moving time of robots. When the marker size increases, the value of ΔD increases from 0 to 9.

in Fig. 6(a) that as the value of θ increases, conventional offloading, the algorithm in [34], and the proposed scheme have increased completion time. The proposed scheme can reduce the average completion time of data offloading by 3% to 35% over the conventional schemes. Similar to Fig. 5(b), Fig. 6(b) also shows that the velocity control-enabled data offloading in the proposed scheme can achieve the lowest completion time with a moderate moving time.

Furthermore, Fig. 7 plots the average completion time versus the size of the offloaded data. The data size in each offloading interval is randomly selected from the set $\{100, 350, 600\}(\text{KB}) + \Delta D$, where the incremental parameter ΔD belongs to $\{0, 1, 3, 5, 7, 9\}$ (MB). It can be seen in Fig. 7(a) that the proposed scheme has the best completion time performance for varying data sizes. Compared to conventional schemes, the proposed scheme achieves an average completion time reduction of 3% to 19%. Moreover, as opposed to conventional schemes, with an increase in data size, a much higher reduction in completion time can be obtained in the proposed scheme. In addition, similar to Figs. 5(b) and 6(b), Fig. 7(b) shows that the proposed scheme can also obtain the lowest completion time with a moderate moving time.

VI. CONCLUSION

In this paper, we proposed a joint optimization scheme concerning multi-robot offloading, resource allocation, and velocity control for MEC in a task-oriented satellite-UAV network. To solve the optimization problem with long-term constraints, based on Lyapunov optimization, the original optimization was reduced to multiple AP coverage-based subproblems for velocity control and data offloading. Then, a multi-agent Q-learning algorithm, composed of multi-group dual-agent Q-learning, was presented. The wireless communication availability and reduced computational resource status were observed, and a global reward was calculated. The convergence of the multi-agent Q-learning algorithm was analyzed in terms of Q function, optimal Q function, and convergence rate. Simulation results were presented to show that, compared with conventional schemes, based on velocity control, the proposed scheme can achieve an effective reduction in offloading time for multiple robots. It was concluded that mobility control is beneficial for providing a high-quality offloading performance in the multi-robot environment with time-varying bandwidth and dynamic computational resources.

In practical multi-robot environments, satellite communications and UAV communications may utilize the same frequency band. To avoid interference between the satellite-robot and UAV-robot links, a spectrum sharing strategy for multiple robots within the satellite-UAV network will be developed in our future work. Based on the robots' demand for UAV communications or satellite communications, we plan to modify the calculation of signal-to-noise ratio (SNR) in (3) or (5) by replacing the channel noise in the denominator with the sum of channel noise and interference, thus forming the signal-to-interference-plus-noise ratio (SINR). Moreover, when the positions of tasks change constantly, it may be necessary to dynamically adjust the UAV deployment. A high UAV velocity will significantly increase the dynamics of network coverage and service migration. Therefore, the velocity control for UAVs will be an extension of the velocity control model presented in this paper.

A. PROOF OF LEMMA 1

According to [40], Lemma 1 is proved by three cases of $\lim_{j\to\infty}\lambda_j$. First, when $\lim_{j\to\infty}\lambda_j = h > 0$, according to the definition of limit, for any $w > 0$, there is a positive integer T, such that when $t > T$, we have $|\lambda_j - h| < w$, that is $h - w < \lambda_j < h + w$. If $h - w > 0$ and $0 \le \phi < \infty$, we have

$$\begin{aligned}\prod_{j=0}^{\infty}\left(1-\frac{\lambda_j}{\phi+1}\right) &= \prod_{j=0}^{T-1}\left(1-\frac{\lambda_j}{\phi+1}\right)\prod_{j=0}^{\infty}\left(1-\frac{\lambda_j}{\phi+1}\right)\\ &\le \prod_{j=0}^{T-1}\left(1-\frac{\lambda_j}{\phi+1}\right)\prod_{j=T}^{\infty}\left(1-\frac{h-w}{\phi+1}\right)\\ &= \prod_{j=0}^{T-1}\left(1-\frac{\lambda_j}{\phi+1}\right)\lim_{j\to\infty}\left(1-\frac{h-w}{\phi+1}\right)^j\\ &= 0. \qquad \text{(A.1)}\end{aligned}$$

Moreover, due to $0 \le \lambda_j \le 1$ and $0 \le \phi$, we have $\prod_{j=0}^{\infty}(1-\frac{\lambda_j}{\phi+1}) \ge 0$. Therefore, $\prod_{j=0}^{\infty}(1-\frac{\lambda_j}{\phi+1}) = 0$ is achieved.

Second, when $\lim_{j\to\infty}\lambda_j = 0$, for $0 < \phi < \infty$ and $0 \le \lambda_j \le 1$, we obtain

$$\ln\left(\prod_{j=0}^{\infty}\left(1-\frac{\lambda_j}{\phi+1}\right)\right) = \sum_{j=0}^{\infty}\ln\left(1-\frac{\lambda_j}{\phi+1}\right) \le 0. \qquad \text{(A.2)}$$

Furthermore, according to the property of limit for the quotient of functions [40], we have

$$\lim_{j\to\infty}\frac{-\ln\left(1-\frac{\lambda_j}{\phi+1}\right)}{\lambda_j} = \frac{1}{\phi+1}. \qquad \text{(A.3)}$$

Since $\frac{1}{\phi+1}$ is a constant, it is inferred that $\prod_{j=0}^{\infty}(1-\frac{\lambda_j}{\phi+1})$ and $\sum_{j=0}^{\infty}\lambda_j$ have the same convergence. If $\sum_{j=0}^{\infty}\lambda_j \to \infty$, we obtain $-\sum_{j=0}^{\infty}\ln(1-\frac{\lambda_j}{\phi+1}) \to \infty$. Thus, it is proved that $\prod_{j=0}^{\infty}(1-\frac{\lambda_j}{\phi+1}) = e^{-\infty} = 0$.

Finally, when $\lim_{j\to\infty}\lambda_j$ does not exist, we assume that there exists a lower bound y $(0 < y < 1)$, such that infinite λ_j belong to the set $\mathcal{Y} = \{\lambda_j | y < \lambda_j \le 1\}$. Then, it can be concluded that

$$\prod_{j=0,\lambda_j\in\mathcal{Y}}^{\infty}\left(1-\frac{\lambda_j}{\phi+1}\right) \le \lim_{j\to\infty}\left(1-\frac{y}{\phi+1}\right)^j = 0. \qquad \text{(A.4)}$$

For $0 \le \phi < \infty$, we have $\prod_{j=0,\lambda_j\in\mathcal{Y}}^{\infty}(1-\frac{\lambda_j}{\phi+1}) \ge 0$. It is inferred that $\prod_{j=0,\lambda_j\in\mathcal{Y}}^{\infty}(1-\frac{\lambda_j}{\phi+1}) = 0$. Therefore, for $\prod_{j=0,\lambda_j\notin\mathcal{Y}}^{\infty}(1-\frac{\lambda_j}{\phi+1}) < \infty$, we obtain

$$\begin{aligned}&\prod_{j=0}^{\infty}\left(1-\frac{\lambda_j}{\phi+1}\right)\\ &\quad= \prod_{j=0,\lambda_j\in\mathcal{Y}}^{\infty}\left(1-\frac{\lambda_j}{\phi+1}\right)\prod_{j=0,\lambda_j\notin\mathcal{Y}}^{\infty}\left(1-\frac{\lambda_j}{\phi+1}\right)\\ &\quad= 0. \qquad \text{(A.5)}\end{aligned}$$

Finally, upon combining the value ranges of λ_j and ϕ in the above three cases, Lemma 1 is proved.

B. PROOF OF THEOREM 1

Given an arbitrarily small non-negative value $\zeta \ge 0$, we define a set as $\mathcal{X}_\zeta = \{.(s_t, a_t)|s_t \in \mathcal{X}_s, a_t \in \mathcal{X}_a, \|s_t\| + \|a_t\| \le \zeta\}$. According to [47], it can be concluded that for $\zeta \to 0$, we have $Q_j(s_t, a_t) \to 0$. Moreover, since $Q^*(s_t, a_t) = r(s_t, a_t) + \gamma \max\{\sum_{i=0}^{\infty} r(s_{t+i}, a_{t+i})\}$ and $Q_0(s_t, a_t)$ is semi-positive definite, we can derive that $Q^*(s_t, a_t)$ is positive definite. Thus, for $\zeta = 0$, we obtain $Q_j(s_t, a_t) = Q^*(s_t, a_t)$.

Next, we use mathematical induction to prove the properties of $Q_j(s_t, a_t)$ for $\zeta > 0$. Based on [48], for any $\zeta > 0$ and $(s_t, a_t) \notin \mathcal{X}_\zeta$, there are three constants $\underline{y}$ $(0 \le \underline{y} \le 1)$,

$\bar{y}$ $(1 \leq \bar{y} \leq \infty)$, and $(y_0\ 0 < y_0 < \infty)$ satisfying the following condition

$$\left(1+\left(\underline{y}-1\right)\prod_{l=0}^{j-1}\left(1-\frac{\lambda_l}{y_0+1}\right)\right)Q^*(s_t,a_t) \leq Q_j(s_t,a_t)$$
$$\leq \left(1+(\bar{y}-1)\prod_{l=0}^{j-1}\left(1-\frac{\lambda_l}{y_0+1}\right)\right)Q^*(s_t,a_t). \tag{B.1}$$

First, when $j=0$, we have $\underline{y}Q^*(s_t,a_t) \leq Q_0(s_t,a_t) \leq \bar{y}Q^*(s_t,a_t)$, which satisfies the condition (B.1).

Then, when $j=1$, we can derive

$$\begin{aligned}
&Q_1(s_t,a_t)\\
&= (1-\lambda_0)Q_0(s_t,a_t)+\lambda_0(r(s_t,a_t)+\gamma\max\{Q_0(s_{t'},a_{t'})\})\\
&\leq \bar{y}(1-\lambda_0)Q^*(s_t,a_t)+\lambda_0\left(1+\frac{y_0(\bar{y}-1)}{y_0+1}\right)\\
&\quad\times\left(r(s_t,a_t)+\gamma\max\left\{Q^*(s_{t'},a_{t'})\right\}\right)\\
&= \left(1+(\bar{y}-1)\left(1-\frac{\lambda_0}{y_0+1}\right)\right)Q^*(s_t,a_t),
\end{aligned}\tag{B.2}$$

$$\begin{aligned}
&Q_1(s_t,a_t)\\
&\geq (1-\lambda_0)Q_0(s_t,a_t)+\lambda_0\left(\left(1+\frac{y_0\left(\underline{y}-1\right)}{y_0+1}\right)r(s_t,a_t)\right.\\
&\quad\left.+\left(\underline{y}-\frac{\underline{y}-1}{y_0+1}\right)\gamma\max\left\{Q^*(s_{t'},a_{t'})\right\}\right)\\
&= \left(1+\left(\underline{y}-1\right)\left(1-\frac{\lambda_0}{y_0+1}\right)\right)Q^*(s_t,a_t),
\end{aligned}\tag{B.3}$$

which also satisfies condition (B.1).

When $j=l+1$, under the assumption of $y_0 r(s_t,a_t) \geq \gamma\max\{Q^*(s_{t'},a_{t'})\}$, we derive

$$\begin{aligned}
&Q_{l+1}(s_t,a_t)\\
&= (1-\lambda_l)Q_l(s_t,a_t)+\lambda_l(r(s_t,a_t)+\gamma\max\{Q_l(s_{t'},a_{t'})\})\\
&\leq (1-\lambda_l)Q_l(s_t,a_t)+\lambda_l\left(r(s_t,a_t)+\left(1+(\bar{y}-1)\right.\right.\\
&\quad\times\left.\prod_{j=0}^{l-1}\left(1-\frac{\lambda_j}{y_0+1}\right)\right)\gamma\max\left\{Q^*(s_{t'},a_{t'})\right\}+\frac{\bar{y}-1}{y_0+1}\\
&\quad\times\left.\prod_{j=0}^{l-1}\left(1-\frac{\lambda_j}{y_0+1}\right)\left(y_0 r(s_t,a_t)-\gamma\max\left\{Q^*(s_{t'},a_{t'})\right\}\right)\right)\\
&\leq (1-\lambda_l)\left(1+(\bar{y}-1)\prod_{j=0}^{l-1}\left(1-\frac{\lambda_j}{y_0+1}\right)\right)Q^*(s_t,a_t)\\
&\quad+\lambda_l\left(1+\frac{y_0(\bar{y}-1)}{y_0+1}\prod_{j=0}^{l-1}\left(1-\frac{\lambda_j}{y_0+1}\right)\right)Q^*(s_t,a_t)\\
&\leq \left(1+(\bar{y}-1)\prod_{j=0}^{l-1}\left(1-\frac{\lambda_j}{y_0+1}\right)\right)Q^*(s_t,a_t).
\end{aligned}\tag{B.4}$$

Furthermore, $Q_{l+1}(s_t,a_t)$ satisfies

$$\begin{aligned}
&Q_{l+1}(s_t,a_t)\\
&\geq (1-\lambda_l)Q_l(s_t,a_t)+\lambda_l\left(r(s_t,a_t)\right.\\
&\quad\left.+\left(1+\left(\underline{y}-1\right)\prod_{j=0}^{l-1}\left(1-\frac{\lambda_j}{y_0+1}\right)\right)\gamma\max\left\{Q^*(s_{t'},a_{t'})\right\}\right)\\
&\geq (1-\lambda_l)Q_l(s_t,a_t)+\lambda_l\left(1+\left(\underline{y}-1\right)\prod_{j=0}^{l-1}\left(1-\frac{\lambda_j}{y_0+1}\right)\right)\\
&\quad\times\left(r(s_t,a_t)+\gamma\max\left\{Q^*(s_{t'},a_{t'})\right\}\right)\\
&\geq (1-\lambda_l)\left(1+\left(\underline{y}-1\right)\prod_{j=0}^{l-1}\left(1-\frac{\lambda_j}{y_0+1}\right)\right)Q^*(s_t,a_t)\\
&\quad+\lambda_l\left(1+\left(\underline{y}-1\right)\prod_{j=0}^{l-1}\left(1-\frac{\lambda_j}{y_0+1}\right)\right)Q^*(s_t,a_t)\\
&= \left(1+\left(\underline{y}-1\right)\prod_{j=0}^{l-1}\left(1-\frac{\lambda_j}{y_0+1}\right)\right)Q^*(s_t,a_t),
\end{aligned}\tag{B.5}$$

which satisfies the condition (B.1).

Consequently, for $j\to\infty$, we have

$$\lim_{j\to\infty}\left(1+\left(\underline{y}-1\right)\prod_{l=0}^{j-1}\left(1-\frac{\lambda_l}{y_0+1}\right)\right)Q^*(s_t,a_t)=Q^*(s_t,a_t), \tag{B.6}$$

$$\lim_{j\to\infty}\left(1+(\bar{y}-1)\prod_{l=0}^{j-1}\left(1-\frac{\lambda_l}{y_0+1}\right)\right)Q^*(s_t,a_t)=Q^*(s_t,a_t). \tag{B.7}$$

Based on (B.6) and (B.7), we obtain (27). Thus, Theorem 1 is proved.

C. PROOF OF THEOREM 2

Based on the mathematical induction, when $j=0$, $Q_{u,v,1,0}(s_{u,v,m,n}(t),a_{u,v,m,n}(t))$ and $Q_{u,v,2,0}(s_{u,v,m},a_{u,v,m})$ can be initialized based on (28)-(31). For example, the values of $Q_{u,v,1,0}(s_{u,v,m,n}(t),a_{u,v,m,n}(t))$ and $Q_{u,v,2,0}(s_{u,v,m},a_{u,v,m})$ can be selected from the intervals $[-\frac{r_{u,v,\max}}{1-\gamma},\frac{r_{u,v,\max}}{1-\gamma}]$ and $[-\frac{L_{u,v,\max}r_{u,v,\max}}{1-\gamma},\frac{L_{u,v,\max}r_{u,v,\max}}{1-\gamma}]$, respectively.

When $j=l+1$, we derive

$$\begin{aligned}
&\left\|Q_{u,v,1,l+1}\left(s_{u,v,m,n}(t),a_{u,v,m,n}(t)\right)\right\|\\
&\quad\leq (1-\lambda_l)\left\|Q_{u,v,1,l}\left(s_{u,v,m,n}(t),a_{u,v,m,n}(t)\right)\right\|\\
&\qquad+\lambda_l\left\|r_{u,v,m,n}(t)\right\|+\lambda_l\gamma\\
&\qquad\times\left\|\max_{a_{u,v,m,n}(t')\in\mathcal{A}_{u,v,1}}\left\{Q_{u,v,1,l}\left(s_{u,v,m,n}\left(t'\right),a_{u,v,m,n}\left(t'\right)\right)\right\}\right\|\\
&\quad\leq (1-\lambda_l)\frac{r_{u,v,\max}}{1-\gamma}+\lambda_l r_{u,v,\max}+\lambda_l\gamma\frac{r_{u,v,\max}}{1-\gamma}\\
&\quad=\frac{r_{u,v,\max}}{1-\gamma},
\end{aligned}\tag{C.1}$$

where $t' = t + 1$ and $l = 0, 1, \ldots, T_{\text{Epi}} - 2$. Similarly, $Q_{u,v,2,j}(s_{u,v,m}, a_{u,v,m})$ can be expressed as

$$\begin{aligned}
&\left\| Q_{u,v,2,l+1}\big(s_{u,v,m}, a_{u,v,m}\big) \right\| \\
&\quad \leq (1-\lambda_l)\left\| Q_{u,v,2,l}\big(s_{u,v,m}, a_{u,v,m}\big)\right\| + \lambda_l \left\| r_{u,v,m} \right\| \\
&\qquad + \lambda_j \gamma \left\| \max_{a_{u,v,m'} \in \mathcal{A}_{u,v,2}} \left\{ Q_{u,v,2,l}\big(s_{u,v,m'}, a_{u,v,m'}\big)\right\} \right\| \\
&\quad \leq (1-\lambda_l)\frac{L_{u,v,\max} r_{u,v,\max}}{1-\gamma} + \lambda_l L_{u,v,\max} r_{u,v,\max} \\
&\qquad + \lambda_l \gamma \frac{L_{u,v,\max} r_{u,v,\max}}{1-\gamma} \\
&\quad = \frac{L_{u,v,\max} r_{u,v,\max}}{1-\gamma},
\end{aligned} \tag{C.2}$$

where $m' = m + 1$. Therefore, the following relationships are achieved as

$$\begin{aligned}
&\left\| Q_{u,v,1,j}\big(s_{u,v,m,n}(t), a_{u,v,m,n}(t)\big) - Q^*_{u,v,1} \right\| \\
&\quad \leq \left\| Q_{u,v,1,j}\big(s_{u,v,m,n}(t), a_{u,v,m,n}(t)\big) \right\| + \left\| Q^*_{u,v,1} \right\| \\
&\quad \leq \frac{2r_{u,v,\max}}{1-\gamma},
\end{aligned} \tag{C.3}$$

$$\begin{aligned}
&\left\| Q_{u,v,2,j}\big(s_{u,v,m}, a_{u,v,m}\big) - Q^*_{u,v,2} \right\| \\
&\quad \leq \left\| Q_{u,v,2,j}\big(s_{u,v,m}, a_{u,v,m}\big) \right\| + \left\| Q^*_{u,v,2} \right\| \\
&\quad \leq \frac{2L_{u,v,\max} r_{u,v,\max}}{1-\gamma}.
\end{aligned} \tag{C.4}$$

Therefore, Theorem 2 is proved.

D. PROOF OF THEOREM 3

According to [49], we first define the Bellman operation as

$$\mathrm{T}\{Q(s,a)\} = \sum_{s' \in \mathcal{S}} p_a\big(s, s'\big)\left(r(s,a) + \gamma \max_{a' \in \mathcal{A}} \left\{ Q\big(s', a'\big)\right\}\right), \tag{D.1}$$

where $p_a(s, s')$ represents the transition probability from state s to state s', $\mathcal{S}$ is the state space. Thus, for $i = 1, 2$ and $j = 0, 1, \ldots, T_{\text{Epi}}$, the approximation error $\theta_{u,v,i,j}$ is defined as

$$E\left\{ \left\| Q_{u,v,i,j+1} - \mathrm{T}\left\{Q_{u,v,i,j}\right\} \right\|_2^2 \right\} \leq \theta_{u,v,i,j}. \tag{D.2}$$

According to $Q^*_{u,v,1} = \mathrm{T}\{Q^*_{u,v,1}\}$ and the γ-contraction property of the Bellman operator in [41], we obtain

$$\begin{aligned}
&E\left\{ \left\| Q_{u,v,i,j+1} - Q^*_{u,v,i} \right\|_\infty \right\} \\
&\quad \leq E\left\{ \left\| Q_{u,v,i,j+1} - \mathrm{T}\left\{Q_{u,v,i,j}\right\} \right\|_\infty \right\} \\
&\qquad + E\left\{ \left\| \mathrm{T}\left\{Q_{u,v,i,j}\right\} - Q^*_{u,v,i} \right\|_\infty \right\} \\
&\quad \leq \sqrt{E\left\{ \left\| Q_{u,v,i,j+1} - \mathrm{T}\left\{Q_{u,v,i,j}\right\} \right\|_2^2 \right\}} \\
&\qquad + E\left\{ \left\| \mathrm{T}\left\{Q_{u,v,i,j}\right\} - \mathrm{T}\left\{Q^*_{u,v,i}\right\} \right\|_\infty \right\} \\
&\quad \leq \sqrt{\theta_{u,v,i,j+1}} + \gamma E\left\{ \left\| Q_{u,v,i,j} - Q^*_{u,v,i} \right\|_\infty \right\}.
\end{aligned} \tag{D.3}$$

Based on (D.3), it follows recursively that

$$\begin{aligned}
&E\left\{ \left\| Q_{u,v,i,T_{\text{Epi}}} - Q^*_{u,v,i} \right\|_\infty \right\} \\
&\quad \leq \sqrt{\theta_{u,v,i,T_{\text{Epi}}}} + \gamma E\left\{ \left\| Q_{u,v,i,T_{\text{Epi}}-1} - Q^*_{u,v,i} \right\|_\infty \right\} \\
&\quad \leq \sqrt{\theta_{u,v,i,T_{\text{Epi}}}} \\
&\qquad + \gamma\left(\sqrt{\theta_{u,v,i,T_{\text{Epi}}-1}} + \gamma E\left\{ \left\| Q_{u,v,i,T_{\text{Epi}}-2} - Q^*_{u,v,i} \right\|_\infty \right\}\right) \\
&\qquad \cdots \\
&\quad \leq \sum_{j=0}^{T_{\text{Epi}}} \gamma^{T_{\text{Epi}}-j} \sqrt{\theta_{u,v,i,j}} + \gamma^{T_{\text{Epi}}} E\left\{ \left\| Q_{u,v,i,0} - Q^*_{u,v,i} \right\|_\infty \right\}.
\end{aligned} \tag{D.4}$$

According to Theorem 2, we have $\theta_{u,v,2,j} = L_{u,v,\max}\theta_{u,v,1,j}$. Then, based on (D.4), (34) and (35) are proved.

For $\theta_{u,v,1,j} = \theta_{u,v}$, (D.4) is simplified as

$$\begin{aligned}
&E\left\{ \left\| Q_{u,v,i,T_{\text{Epi}}} - Q^*_{u,v,i} \right\|_\infty \right\} \\
&\quad \leq \frac{1-\gamma^{T_{\text{Epi}}+1}}{1-\gamma} \sqrt{\theta_{u,v,i,j}} + \gamma^{T_{\text{Epi}}} E\left\{ \left\| Q_{u,v,i,0} - Q^*_{u,v,i} \right\|_\infty \right\}.
\end{aligned} \tag{D.5}$$

When $T_{\text{Epi}} \to \infty$, due to $0 \leq \gamma < 1$, we have $\gamma^{T_{\text{Epi}}+1} \to 0$. Then, (D.5) is expressed as

$$\begin{aligned}
&E\left\{ \left\| Q_{u,v,i,T_{\text{Epi}}} - Q^*_{u,v,i} \right\|_\infty \right\} \\
&\quad \leq \frac{\sqrt{\theta_{u,v,i,j}}}{1-\gamma} + \gamma^{T_{\text{Epi}}} E\left\{ \left\| Q_{u,v,i,0} - Q^*_{u,v,i} \right\|_\infty \right\}.
\end{aligned} \tag{D.6}$$

Consequently, Theorem 3 is proved.

REFERENCES

[1] *Next G Alliance Report: 6G Applications and Use Cases*," Alliance Telecommun. Ind. Sol., Washington, DC, USA, 2022.

[2] J. Gielis, A. Shankar, and A. Prorok, "A critical review of communications in multi-robot systems," *Curr. Robot. Rep.*, vol. 3, no. 4, pp. 213–225, Dec. 2022.

[3] K. Tiwari and N. Y. Chong, *Multi-robot Exploration for Environmental Monitoring: The Resource Constrained Perspective*. Cambridge, MA, USA: Academic, 2020.

[4] Z. Jiang et al., "A multirobot system for autonomous deployment and recovery of a blade crawler for operations and maintenance of offshore wind turbine blades," *J. Field. Rob.*, vol. 40, no. 1, pp. 73–93, Jan. 2023.

[5] Y. Wang, W. Feng, J. Wang, and T. Q. S. Quek, "Hybrid satellite-UAV-terrestrial networks for 6G ubiquitous coverage: A maritime communications perspective," *IEEE J. Sel. Areas Commun.* vol. 39, no. 11, pp. 3475–3490, Nov. 2021.

[6] X. Li, W. Feng, J. Wang, Y. Chen, N. Ge, and C.-X. Wang, "Enabling 5G on the ocean: A hybrid satellite-UAV-terrestrial network solution," *IEEE Wireless Commun.* vol. 27, no. 6, pp. 116–121, Dec. 2020.

[7] X. Fang, W. Feng, T. Wei, Y. Chen, N. Ge, and C.-X. Wang, "5G embraces satellites for 6G ubiquitous IoT: Basic models for integrated satellite terrestrial networks," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 14399–14417, Sep. 2021.

[8] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., Mar. 2017.

[9] B. Mao, F. Tang, Y. Kawamoto, and N. Kato, "Optimizing computation offloading in satellite-UAV-served 6G IoT: A deep learning approach," *IEEE Netw.*, vol. 35, no. 4, pp. 102–108, Jul./Aug. 2021.

[10] N. Cheng et al., "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.* vol. 37, no. 5, pp. 1117–1129, May 2019.

[11] Y.-H. Chao, C.-H. Chung, C.-H. Hsu, Y. Chiang, H.-Y. Wei, and C.-T. Chou, "Satellite-UAV-MEC collaborative architecture for task offloading in vehicular networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Taipei, Taiwan, 2020, pp. 1–6.

[12] Y. Liu, S. Xie, and Y. Zhang, "Cooperative offloading and resource management for UAV-enabled mobile edge computing in power IoT system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12229–12239, Oct. 2020.

[13] Y. Chen, B. Ai, Y. Niu, H. Zhang, and Z. Han, "Energy-constrained computation offloading in space-air-ground integrated networks using Distributionally robust optimization," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 12113–12125, Nov. 2021.

[14] H. Zeng et al., "Collaborative computation offloading for UAVs and USV fleets in communication networks," in *Proc. Int. Wirel. Commun. Mob. Comput. (IWCMC)*, Dubrovnik, Croatia, 2022, pp. 949–954.

[15] F. Tang, C. Wen, L. Luo, M. Zhao, and N. Kato, "Blockchain-based trusted traffic offloading in space-air-ground integrated networks (SAGIN): A federated reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3501–3516, Dec. 2022.

[16] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, "Satellite-terrestrial integrated edge computing networks: Architecture, challenges, and open issues," *IEEE Netw.*, vol. 34, no. 3, pp. 224–231, May/Jun. 2020.

[17] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing," *IEEE J. Sel. Areas Commun.* vol. 36, no. 10, pp. 2333–2345, Oct. 2018.

[18] Y. Li, J. Huang, Q. Sun, T. Sun, and S. Wang, "Cognitive service architecture for 6G core network," *IEEE Trans. Ind. Inform.*, vol. 17, no. 10, pp. 7193–7203, Oct. 2021.

[19] S. Yu, X. Gong, Q. Shi, X. Wang, and X. Chen, "EC-SAGINs: Edge-computing-enhanced space-air-ground-integrated networks for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5742–5754, Apr. 2022.

[20] Z. Li, C. Jiang, and J. Lu, "Distributed service migration in satellite mobile edge computing," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Madrid, Spain, Dec. 2021, pp. 1–6.

[21] H. Han, H. Wang, and S. Cao, "Space edge cloud enabling service migration for on-orbit service," in *Proc. IEEE Int. Conf. Commun. Softw. Netw. (ICCSN)*, Chongqing, China, 2020, pp. 233–239.

[22] W. Lan, K. Chen, Y. Li, J. Cao, and Y. Sahni, "Deep reinforcement learning for privacy-preserving task offloading in integrated satellite-terrestrial networks," 2023, *arXiv:2306.17183*.

[23] J. Li, W. Shi, H. Wu, S. Zhang, and X. Shen, "Cost-aware dynamic SFC mapping and scheduling in SDN/NFV-enabled space–air–ground-integrated networks for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5824–5838, Apr. 2022.

[24] Q. Zhao, P. Du, M. Gerla, A. J. Brown, and J. H. Kim, "Software defined multi-path TCP solution for mobile wireless tactical networks," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Los Angeles, CA, Oct. 2018, pp. 1–9.

[25] I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Spatial configuration of agile wireless networks with drone-BSs and user-in-the-loop," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 753–768, Feb. 2019.

[26] R. Schoenen and H. Yanikomeroglu, "User-in-the-loop: Spatial and temporal demand shaping for sustainable wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 196–203, Feb. 2014.

[27] P. M. de Sant Ana, N. Marchenko, P. Popovski, and B. Soret, "Wireless control of autonomous guided vehicle using reinforcement learning," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Taipei, Taiwan, 2020, pp. 1–7.

[28] Y. Yan and Y. Mostofi, "Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments," *IEEE Trans. Wireless Commun.* vol. 12, no. 4, pp. 1562–1572, Apr. 2013.

[29] A. Ghaffarkhah and Y. Mostofi, "Communication-aware motion planning in mobile networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2478–2485, Oct. 2011.

[30] Y. Kantaros and M. M. Zavlanos, "Global planning for multi-robot communication networks in complex environments," *IEEE Trans. Robot.* vol. 32, no. 5, pp. 1045–1061, Oct. 2016.

[31] Y. Kantaros and M. M. Zavlanos, "Distributed intermittent connectivity control of mobile robot networks," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3109–3121, Jul. 2017.

[32] D. B. Licea, M. Bonilla, M. Ghogho, S. Lasaulce, and V. S. Varma, "Communication-aware energy efficient trajectory planning with limited channel knowledge," *IEEE Trans. Robot.*, vol. 36, no. 2, pp. 431–442, Apr. 2020.

[33] A. Ghaffarkhah and Y. Mostofi, "Path planning for networked robotic surveillance," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3560–3575, Jul. 2012.

[34] P. Wei, W. Feng, Y. Wang, Y. Chen, N. Ge, and C.-X. Wang, "Joint mobility control and MEC offloading for hybrid satellite-terrestrial-network-enabled robots," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 8483–8497, Nov. 2023.

[35] Y. Peng, L. Liu, Y. Zhou, J. Shi, and J. Li, "Deep reinforcement learning-based dynamic service migration in vehicular networks," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Waikoloa, HI, 2019, pp. 1–6.

[36] Z. Zhang, W. Zhang, and F.-H. Tseng, "Satellite mobile edge computing: Improving QoS of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE Netw.* vol. 33, no. 1, pp. 70–76, Jan./Feb. 2019.

[37] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6G," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12240–12251, Oct. 2020.

[38] *Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV Environment, V1.1.1*, ETSI, Sophia Antipolis, France, ETSI, Rep. GR MEC 017, 2018.

[39] "*Multi-Access Edge Computing (MEC); Framework and Reference Architecture, V2.2.1*," ETSI, Sophia Antipolis, France, ETSI, Rep. ETSI-GS-MEC 003, 2020.

[40] Q. Wei, F. L. Lewis, Q. Sun, P. Yan, and R. Song, "Discrete-time deterministic Q-learning: A novel convergence analysis," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1224–1237, May 2017.

[41] H. Xiong, L. Zhao, Y. Liang, and W. Zhang, "Finite-time analysis for double Q-learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 16628–16638.

[42] P. Sunehag et al., "Value-decomposition networks for cooperative multi-agent learning," 2017, *arXiv:1706.05296*.

[43] E. Rodrigues Gomes and R. Kowalczyk, "Dynamic analysis of multiagent Q-learning with ϵ-greedy exploration," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Montreal, QC, Canada, 2009, pp. 369–376.

[44] M. Wunder, M. Littman, and M. Babes, "Classes of Multiagent Q-learning dynamics with ϵ-greedy exploration," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Haifa, Israel, 2010, pp. 1167–1174.

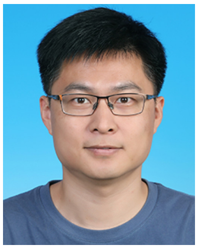
[45] K. Tuyls, K. Verbeeck, and T. Lenaerts, "A selection-mutation model for Q-learning in multi-agent systems," in *Proc. Int. Conf. Autonom. Agents (AAMAS)*, Melbourne, VIC, Australia, 2003, pp. 693–700.

[46] M. Kaisers, "Learning against learning: Evolutionary dynamics of reinforcement learning algorithms in strategic interactions," Ph.D. dissertation, Dept. Adv. Comput. Sci., Maastricht Univ., Maastricht, Maastricht, The Netherlands, Jan. 2012. [Online]. Available: https://doi.org/10.26481/dis.20121217mk

[47] Q. Wei and D. Liu, "A novel iterative θ-adaptive dynamic programming for discrete-time nonlinear systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1176–1190, Oct. 2014.

[48] B. Lincoln and A. Rantzer, "Relaxing dynamic programming," *IEEE Trans. Autom. Control* vol. 51, no. 8, pp. 1249–1260, Aug. 2006.

[49] D. Lee and N. He, "Periodic Q-learning," in *Proc. Conf. Learn. Dyn. Control (L4DC)*, 2020, pp. 582–598.



PENG WEI (Member, IEEE) received the Ph.D. degree in communication and information systems from the University of Electronic Science and Technology of China in 2017. He was also a visiting student with the Department of Electrical and Computer Engineering, University of Delaware from 2014 to 2016. He was a Lecturer with the School of Electronics and Information Engineering, Tiangong University from 2017 to 2020. He has been a Postdoctoral Fellow with the Department of Electronic Engineering, Tsinghua University Since 2020. His research interests are in wireless communication, multicarrier system, and signal processing.

WEI FENG (Senior Member, IEEE) received the B.S. and Ph.D. degrees from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2005 and 2010, respectively. He is currently a Professor with the Department of Electronic Engineering, Tsinghua University. His research interests include maritime communication networks, large-scale distributed antenna systems, and coordinated satellite-UAV-terrestrial networks. He serves as the Assistant to the Editor-in-Chief for CHINA COMMUNICATIONS and an Editor for IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING.

YUNFEI CHEN (Senior Member, IEEE) received the B.E. and M.E. degrees in electronics engineering from Shanghai Jiaotong University, Shanghai, China, in 1998 and 2001, respectively, and the Ph.D. degree from the University of Alberta in 2006. He is currently a Professor with the Department of Engineering, University of Durham, U.K. His research interests include wireless communications, cognitive radios, wireless relaying, and energy harvesting.

NING GE (Member, IEEE) received the B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 1993 and 1997, respectively. From 1998 to 2000, he was with ADC Telecommunications, Dallas, TX, USA, where he researched the development of ATM switch fabric ASIC. Since 2000, he has been a Professor with the Department of Electronics Engineering, Tsinghua University. He has published over 100 papers. His current research interests include communication ASIC design, short-range wireless communication, and wireless communications. He is a Senior Member of the China Institute of Communications and the Chinese Institute of Electronics.

WEI XIANG (Senior Member, IEEE) is the Cisco Research Chair of AI and IoT and the Director of the Cisco-La Trobe Centre for AI and IoT, La Trobe University. Previously, he was the Foundation Chair and the Head of Discipline of IoT Engineering, James Cook University, Cairns, Australia. He has published over 300 peer-reviewed papers, including three books and 220 journal articles. His research interest includes the Internet of Things (IoT), wireless communications, machine learning for IoT data analytics, and computer vision. Due to his instrumental leadership in establishing Australia's first accredited Internet of Things Engineering degree program, he was inducted into Pearcy Foundation's Hall of Fame in October 2018. He received the TNQ Innovation Award in 2016 and Pearcey Entrepreneurship Award in 2017, and Engineers Australia Cairns Engineer of the Year in 2017. He has been awarded several prestigious fellowship titles. He was a co-recipient of four Best Paper Awards at WiSATS'2019, WCSP'2015, IEEE WCNC'2011, and ICWMC'2009. He was named a Queensland International Fellow from 2010 to 2011 by the Queensland Government of Australia, an Endeavour Research Fellow from 2012 to 2013 by the Commonwealth Government of Australia, a Smart Futures Fellow from 2012 to 2015 by the Queensland Government of Australia, and a JSPS Invitational Fellow jointly by the Australian Academy of Science and Japanese Society for Promotion of Science from 2014 to 2015. He has served in a large number of international conferences in the capacity of the general co-chair, the TPC co-chair, and the symposium chair. He was the Vice Chair of the IEEE Northern Australia Section from 2016 to 2020. He is currently an Associate Editor of IEEE COMMUNICATIONS SURVEYS & TUTORIALS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE INTERNET OF THINGS JOURNAL, IEEE ACCESS, and *Scientific Reports* (Nature). He is a TEDx Speaker and an Elected Fellow of the IET in U.K., and Engineers Australia.

SHIWEN MAO (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Polytechnic University, Brooklyn, NY, USA, in 2004. He is currently a Professor and the Earle C. Williams Eminent Scholar Chair, and the Director of the Wireless Engineering Research and Education Center, Auburn University. His research interest includes wireless networks, multimedia communications, and smart grid. He received the IEEE MMTC Outstanding Researcher Award in 2023, the IEEE TC-CSR Distinguished Technical Achievement Award in 2019, and the NSF CAREER Award in 2010. He is a co-recipient of the 2022 Best Journal Paper Award of IEEE ComSoc eHealth Technical Committee (TC), the 2021 Best Paper Award of Elsevier/KeAi *Digital Communications and Networks Journal*, the 2021 IEEE INTERNET OF THINGS JOURNAL Best Paper Award, the 2021 IEEE Communications Society Outstanding Paper Award, the IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, the 2018 Best Journal Paper Award and the 2017 Best Conference Paper Award from IEEE ComSoc Multimedia TC, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He is a co-recipient of the Best Paper Awards from IEEE ICC 2022 and 2013, IEEE GLOBECOM 2023, 2019, 2016, and 2015, and IEEE WCNC 2015, and the Best Demo Awards from IEEE INFOCOM 2022 and IEEE SECON 2017. He is the Editor-in-Chief of IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING and a Distinguished Lecturer of IEEE Communications Society.