Joint Mobility Control and MEC Offloading for Hybrid Satellite-Terrestrial-Network-Enabled Robots

Peng Wei, Wei Feng, *Senior Member, IEEE*, Yanmin Wang, Yunfei Chen, *Senior Member, IEEE*, Ning Ge, *Member, IEEE*, and Cheng-Xiang Wang, *Fellow, IEEE*

***Abstract*—Benefiting from the fusion of communication and intelligent technologies, network-enabled robots have become important to support future machine-assisted and unmanned applications. To provide high-quality services for robots in wide areas, hybrid satellite-terrestrial networks are a key technology. Through hybrid networks, computation-intensive and latency-sensitive tasks can be offloaded to mobile edge computing (MEC) servers. However, due to the mobility of mobile robots and unreliable wireless network environments, excessive local computations and frequent service migrations may significantly increase the service delay. To address this issue, this paper aims to minimize the average task completion time for MEC-based offloading initiated by satellite-terrestrial-network-enabled robots. Different from conventional mobility-aware schemes, the proposed scheme makes the offloading decision by jointly considering the mobility control of robots. A joint optimization problem of task offloading and velocity control is formulated. Using Lyapunov optimization, the original optimization is decomposed into a velocity control subproblem and a task offloading subproblem. Then, based on the Markov decision process (MDP), a dual-agent reinforcement learning (RL) algorithm is proposed. The convergence and complexity of the improved RL algorithm are theoretically analyzed, and the simulation results show that the proposed scheme can effectively reduce the offloading delay.**

***Index Terms*—Mobile edge computing, reinforcement learning, satellite-terrestrial network, task offloading, velocity control.**

I. Introduction

With the rapid development of communication and intelligent technologies, network-enabled robots have become an important application for the advancement of the future society, such as in assisted-living, industry, and transport environments [1]–[5]. When robots operate in wide areas, the hybrid satellite-terrestrial network is key to provide ubiquitous coverage and information perception [6], [7]. Since robots, especially mobile robots, always have limited computing capabilities and storage capacities, their computation-intensive and delay-sensitive tasks can be uploaded to powerful edge servers with the aid of hybrid networks. This is known as task offloading in mobile edge computing (MEC) [8]. Meanwhile, to efficiently complete specific missions within a given time, such as in cooperation among multiple robots, these robots also need to perform some constrained movements. However, due to the mobility of robots and time-varying requirements of task offloading, hybrid satellite-terrestrial networks are highly dynamic. A service migration occurs when a robot moves away from its original location, and thus, its current MEC server that provides mobile service is different from the previous MEC server [9]. Furthermore, compared with conventional clouds, MEC systems have limited computation and storage resources [8]. On the other hand, wireless environments are unreliable [10]–[15]. In this regard, when a large number of mobile robots access the hybrid network, frequent service migrations may deteriorate the hybrid network environment, such as network overload and packet loss. As a result, the service delay of offloaded tasks for hybrid satellite-terrestrial-network-enabled robots will be significantly increased.

Many methods have been proposed to address the migration problem in MEC-based terrestrial networks [16]–[21]. In [16], considering distributed user mobility, a multi-agent reinforcement learning (RL) algorithm was presented. To minimize the task offloading delay with the accumulated service migration cost, the MEC-based digital twin network was optimized by RL in [17]. To balance the high quality of services (QoS) and migration cost, [18] proposed a deep RL enabled optimization scheme in a vehicular network. Furthermore, according to predictable trajectories and mobility-induced communication rates, a mobility-aware task offloading policy was designed in [19]. By assigning velocity-based access priorities to mobile devices [20], speed-aware task offloading was optimized by RL. Leveraging mobility, [21] devised a microservice coordination scheme to minimize the overall service delay. However, when the satellite communication is incorporated, the heterogeneity between satellite and cellular communication systems, such as different propagation delays and different communication rates, may result in higher service latencies.

Thus, an increasing number of studies on MEC based on satellite-terrestrial networks have been conducted for co-

The work of Peng Wei, Wei Feng, and Ning Ge was supported by the National Key Research and Development Program of China under Grant 2020YFA0711301, the National Natural Science Foundation of China under Grants U22A2002 and 61901298, and the Suzhou Science and Technology Project. The work of Yunfei Chen was supported by the King Abdullah University of Science and Technology Research Funding (KRF) under Award ORA-2021-CRG10-4696. The work of Cheng-Xiang Wang was supported by the National Natural Science Foundation of China under Grant 61960206006, the Key Technologies R&D Program of Jiangsu (Prospective and Key Technologies for Industry) under Grants BE2022067 and BE2022067-1, and the EU H2020 RISE TESTBED2 project under Grant 872172. Part of this paper has been accepted by the IEEE ICC 2023. *(Corresponding author: Wei Feng.)*

Peng Wei, Wei Feng, and Ning Ge are with the Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: wpwwwhttp@163.com; fengwei@tsinghua.edu.cn; gening@tsinghua.edu.cn).

Yanmin Wang is with the School of Information Engineering, Minzu University of China, Beijing 100041, China (e-mail: yanmin-226@163.com).

Yunfei Chen is with the Department of Engineering, University of Durham, Durham DH1 3LE, U.K. (e-mail: yunfei.chen@durham.ac.uk).

Cheng-Xiang Wang is with the National Mobile Communications Research Laboratory, School of Information Science and Engineering, Southeast University, Nanjing 210096, China, and also with the Purple Mountain Laboratories, Nanjing 211111, China (e-mail: chxwang@seu.edu.cn).

operative offloading [22]–[26] and service migrations [27]–[30]. In [22], a cooperative computation offloading model was designed to provide high-speed services. In [23], to minimize energy consumption in computation offloading, a cloud-edge collaboration problem was optimized by RL and successive convex approximation algorithms. By considering user preference and evolved satellite processing capabilities, [24] proposed satellite-terrestrial cooperation-based double-edge networks to relieve terrestrial backhaul burdens. To efficiently allocate the distributed MEC servers, the joint optimization of energy consumption and delay was considered in double-edge networks [25]. To jointly optimize the user association, resource allocation, and offloading policy in Internet of Things (IoT) networks using multiple satellites and their gateways, the cost of delay and energy consumption was minimized by the Lagrange multiplier and RL algorithms [26]. In addition, to reduce the migration cost, a service migration model was devised in [27] based on task characteristics to make a tradeoff between task completion time and energy consumption. In [28], the live migration of a virtual network function (VNF) with its reinstantiation and scheduling was studied. Then, two Tabu search-based algorithms were employed for dynamic VNF mapping and scheduling. For low-delay airplane applications in [29], the in-flight service provisioning problem was formulated by routing and reconfiguration and solved by the online heuristic algorithm. Furthermore, a distributed two-layer decomposition model was proposed to minimize the migration cost in [30]. These existing works consider how to optimize task offloading based on the mobility of devices, but they do not consider how to optimize task offloading based on mobility control.

Even in MEC-based terrestrial networks, most mobility-related network optimization works for task processing and resource allocation are from the supply side that involve resource scheduling [31]; wireless control [32], [33]; and task offloading [34], [35]. In these methods, the use of mobility includes mobility prediction [31]; mobility state sharing [32]; mobility control and its stability [33]; and velocity-based task classification [34], [35]. Additionally, in [36], Wu *et al.* studied computation offloading in multi-access MEC to minimize the overall offloading delay of mobile users. In [37], using wireless power transfer for mobile devices with limited energy capacities, a joint optimization of total energy consumption and the learning convergence latency was proposed. Although the effect of mobility control has been analyzed in [33], task offloading and service migration in hybrid satellite-terrestrial networks have not been considered, and the case of no wireless network coverage caused by damage to network infrastructures or heavy network loads is not included either.

Different from conventional network optimization approaches [31]–[37], a demand shaping-based approach was designed from the user side in [38]–[41]. Based on the willingness of users to move, a closed-loop system model for spatial control and temporal control was developed. In spatial control, users are encouraged to move from a severely congested location to a less congested location. In temporal control, an incentive design for reducing the data demand of users in a severely congested location is proposed. However, these approaches are intended for humans, not robots. Furthermore, when all wireless channels are unavailable, mobility control for mobile robots with offloading requirements is not considered. The channel unavailability might be due to a large number of mobile robots accessing limited communication resources, severe channel fading, damage to the access point (AP), and so on. In this case, in the AP coverage where all channels are unavailable, the low-speed movements of mobile robots increase local computations. When wireless channels are available in the AP coverage, high-speed movements of mobile robots lead to more service migrations. As a result, the total service delay is significantly increased.

Motivated by the above observations, in this work, we develop a joint velocity control and MEC-based offloading strategy to improve the QoS in hybrid satellite-terrestrial networks. We consider a scenario where multiple mobile robots cooperate with each other to complete the assigned mission. At the same time, they periodically sense their surroundings, offload data to cellular/satellite MEC servers for processing, and receive computational results from selected MEC servers. Among the surrounding information, the availability of wireless communication is the crucial state information for velocity control, where wireless communication refers to radio communication in this paper. Our objective is to minimize the average task completion time for MEC-based offloading when a mobile robot travels an entire trip. We formulate the long-term offloading problem as a Markov decision process (MDP) problem and propose a joint optimization for velocity control and task offloading. Our main contributions are summarized as follows.

- A joint optimization problem of task offloading, velocity control, and service migration is proposed in this paper, which is formulated to minimize the average task completion time. The non-deterministic polynomial-time hardness (NP-hardness) of the optimization problem is proven. Then, the relationship between the velocity control and task offloading is explored.
- According to the coverage regions of APs, a Lyapunov optimization-based decomposition is employed, which can achieve task offloading optimization and velocity control optimization. Then, based on the MDP, a dual-agent RL algorithm is employed to obtain the effective decision-making of task offloading and velocity control. Furthermore, the convergence and complexity of the improved RL algorithm are analyzed in terms of Q functions. Finally, simulation results show that compared with conventional offloading schemes, the proposed scheme can save up to 40% of the average task completion time.

The rest of this paper is organized as follows. In Section II, the system model of task offloading and velocity control for satellite-terrestrial-network-enabled robots is introduced. In Section III, the optimization model and its NP-hardness are given, and the dual-agent RL algorithm and its convergence are proposed. In Section IV, simulation results are provided. Finally, in Section V, conclusions are drawn.

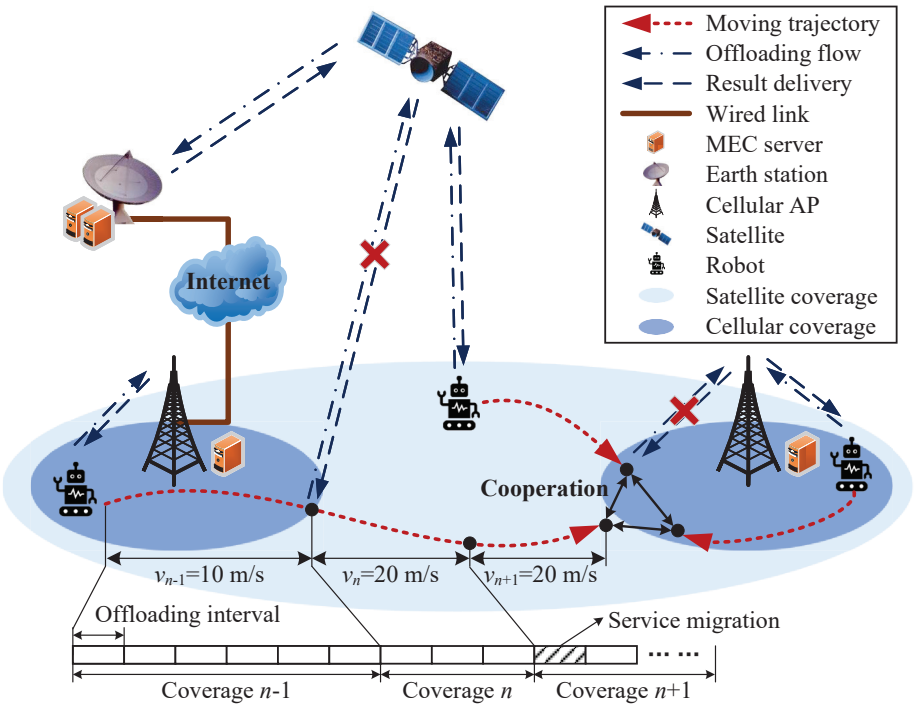


Fig. 1. System model for multiple robots in the hybrid satellite-terrestrial-network with MEC, where v_n denotes the average moving velocity in the nth AP coverage region.

II. System Model

A. System Description

As shown in Fig. 1, a hybrid satellite-terrestrial network with MEC is considered, which can enable multiple robots. Through the hybrid network, these robots intelligently cooperate to complete a specific mission within a given time. To avoid collisions and to complete the collaborative mission on time, efficient control of robot mobility is needed, such as velocity control. Using appropriate velocity control, these robots can move to their designated positions to complete the collaborative mission in time. During the movement, mobile robots also need to complete other tasks to make autonomous decisions. For example, to handle computation-intensive or latency-sensitive tasks, they periodically offload these tasks (such as sensor data) to and receive the computational results from MEC servers through APs. Thus, these robots are multitasking robots that not only complete mobility-controlled cooperative missions but also execute MEC-empowered offloading. For the multi-mobile robot scenario, we focus on a mobile robot moving from one place to another place within a given time in a complex network environment. The reasons include the following. 1) The joint optimization of velocity control and MEC-powered offloading for a satellite-terrestrial-network-enabled robot has not been studied in existing works. Since joint optimization involves two different systems, a hybrid network system and a motion control system, it is also a complex optimization problem. 2) The autonomous decision-making behaviors and mobility behaviors of other robots may complicate the network environment, such as generating the dynamic network load and intermittent communication environment. Hence, other robot-induced network dynamics are regarded as an important part of the complex network environment. A severely degraded network environment will significantly affect data uploading and result downloading. As a result, the velocity control and task offloading decisions of the selected mobile robot cannot be executed in time.

The availability of wireless communication is also included in the periodically perceived surrounding information of the mobile robot. When wireless communication is available, periodic offloading is executed by the mobile robot. Each period is referred to as an offloading interval ΔT. The wireless communication system has N APs, including N_1 cellular APs and N_2 satellites, i.e., $N = N_1 + N_2$. When wireless communication is unavailable, the mobile robot can process tasks using its limited computing capability. When the velocity of the robot is too low, it cannot reach the destination on time and complete the cooperative mission with other robots. Thus, an average velocity should be maintained by the appropriate velocity control to guarantee that the moving time of the mobile robot is smaller than or equal to the due time T_{move} in the whole trip. To coordinate offloading and mobility, we focus on the joint optimization of task offloading and velocity control as a complement to conventional offloading approaches.

In the sequel, we focus on cellular AP-based MEC servers and satellite-based MEC servers, indexed by $\mathcal{N}_1$ and $\mathcal{N}_2$ ($\mathcal{N}_1 \cup \mathcal{N}_2 = \{1, 2, \ldots, N\}$), respectively. For each cellular AP/Earth station, an independent MEC server is equipped, where the MEC server equipped with the Earth station has more computing capabilities than the MEC server equipped with the cellular AP. The MEC server receives and computes the offloaded tasks, and sends the computational results back to the mobile robot. Optical connections are assumed between

MEC servers, between APs, and between APs and MEC servers. It is noted that the satellite MEC servers and cellular MEC servers are connected through the Internet.

In each AP coverage region, when the mobile robot is assumed to move a fixed distance, the number of offloading intervals depends on its moving velocity. A low velocity increases the offloading intervals as well as the moving time. As a result, the mobile robot may not be able to reach the destination on time. More importantly, when wireless communication is unavailable in the current AP coverage region, more computational tasks are undertaken by the selected mobile robot. Since the mobile robot has much smaller computational resources than the MEC server, the task completion time will be significantly increased. On the other hand, a high velocity increases the handover of the mobile robot between multiple APs. In this case, we assume that the information of all robots (in terms of locations and offloading requests) served by different MEC servers is available [42]. Based on such global information, the incomplete task in the current MEC server may be migrated to another MEC server closest to the mobile robot. As a result, the migration time is increased [17], [42], [43]. Therefore, due to the uncertainty of wireless link availability, inappropriate velocities could increase the task completion time in MEC-based offloading.

An example of task offloading and velocity control is shown in Fig. 1. A mobile robot periodically generates tasks (e.g., in each offloading interval). In the $(n-1)$th coverage region, the mobile robot with $v_{n-1} = 10$ m/s has twice as many offloading intervals as the mobile robot with $v_n = 20$ m/s. As a result, when wireless communication is unavailable in the $(n-1)$th AP, a slower movement will further increase the offloading intervals and the number of local computations. In contrast, in the nth AP with available wireless communication, the rapid movement reduces the dwell time of the mobile robot. In this case, some tasks offloaded to the nth MEC server may not be fully processed, and their computational results cannot be returned by the nth AP. Based on the global information of the robot-located coverage region and offloading request, the uncompleted tasks can be migrated from the nth MEC server to the $(n+1)$th MEC server, and their computational results will be sent back by the $(n+1)$th AP. As a result, the rapid movement increases the number of service migrations. According to [34], the low velocity has a high delay tolerance, and the high velocity has a low delay tolerance. Thus, considering the velocity control of the mobile robot and wireless communication availability, we focus on minimizing the process-oriented average task completion time in MEC-based offloading using the hybrid satellite-terrestrial network.

B. Offloading Model

For the task offloading of the mobile robot, the local computation, wireless communication, MEC computation, and service migration are described below.

1) Local computation model: When wireless communication is unavailable in the current AP, the task is computed by the mobile robot. The local computation delay $T_{\mathrm{local}}(t)$ in the tth interval is expressed as

$$T_{\mathrm{local}}(t) = \left(1 - \mathbf{1}_{\alpha}(m,t)\right) \frac{D(t)\Phi}{f_{\mathrm{local}}(t)}, \tag{1}$$

where the indicator function $\mathbf{1}_{\alpha}(m,t)$ stands for the offloading decision in the tth interval, $\mathbf{1}_{\alpha}(m,t) = 1$ for $m \in \alpha = \{1, 2, \ldots, N\}$ denotes that all data are offloaded to the mth MEC server in the tth interval, $\mathbf{1}_{\alpha}(m,t) = 0$ for $m = 0$ indicates the local computation, $D(t)$ is the data size generated by the mobile robot in the tth interval, Φ is the required CPU cycles per bit, and $f_{\mathrm{local}}(t)$ denotes the CPU frequency at the mobile robot.

2) Communication model: We assume that the same communication rate exists in the uplink and downlink. When wireless communication is available in the current cellular AP, the task generated from the mobile robot can be offloaded to an MEC server over wireless channels. In the tth interval, the communication delay $T_{\mathrm{com}}(t)$ is expressed as

$$T_{\mathrm{com}}(t) = \frac{\mathbf{1}_{\alpha}(m,t)\left(D(t) + \bar{D}(t)\right)}{W \log_2\left(1 + \frac{ph^2}{\sigma^2}\right)}, \tag{2}$$

where $\bar{D}(t)$ is the data size of the computational results, p is the transmit power, and h, W, and σ^2 denote the channel gain, bandwidth, and noise power, respectively.

When the offloaded task is transferred from the satellite to the Earth station, the communication delay $T_{\mathrm{com}}(t)$ is

$$T_{\mathrm{com}}(t) = \mathbf{1}_{\alpha}(m,t)\left(2\frac{d_{\mathrm{GS}} + d_{\mathrm{SE}}}{c} + \left(D(t) + \bar{D}(t)\right)\left(\frac{1}{r_{\mathrm{GS}}} + \frac{1}{r_{\mathrm{SE}}}\right)\right), \tag{3}$$

where d_{GS} and d_{SE} denote the distances from the mobile robot to the satellite and from the satellite to the Earth station, respectively, c is the speed of light, and $r_{\mathrm{GS}} = W_{\mathrm{GS}} \log_2\left(1 + \frac{p_{\mathrm{GS}} h_{\mathrm{GS}}^2}{\sigma_{\mathrm{GS}}^2}\right)$ and $r_{\mathrm{SE}} = W_{\mathrm{SE}} \log_2\left(1 + \frac{p_{\mathrm{SE}} h_{\mathrm{SE}}^2}{\sigma_{\mathrm{SE}}^2}\right)$ denote the communication rates in the links between the mobile robot and the satellite and between the satellite and the Earth station, respectively. W_{GS} (or W_{SE}), p_{GS} (or p_{SE}), h_{GS} (or h_{SE}), and σ_{GS}^2 (or σ_{SE}^2) stand for bandwidth, transmit power, channel gain, and noise power in robot-satellite (or satellite-Earth station) transmission.

3) MEC model: With the aid of wireless transmission in the current AP, when the task is offloaded to the mth MEC server in the tth interval, the computation delay $T_{\mathrm{MEC}}(t)$ is given by

$$T_{\mathrm{MEC}}(t) = \frac{\mathbf{1}_{\alpha}(m,t) D(t)\Phi}{f_{\mathrm{MEC},m}(t)}, \tag{4}$$

where $f_{\mathrm{MEC},m}(t)$ is the CPU frequency of the mth MEC server.

4) Migration model: Due to the mobility of the robot, the corresponding service provider (e.g., virtual machine (VM)) is migrated from the initial MEC server to the current counterpart through one- or multi-hop optical communications. In this case, the service downtime may cause a delay that cannot be ignored. In our model, when the MEC server in the $(t-1)$th

interval is different from the MEC server in the tth interval, a migration delay will be incurred, which is expressed as

$$\begin{aligned} T_{\mathrm{mig}}(t) =& I\big\{\mathrm{M}(t-1) \neq \mathrm{M}(t) \cap (\mathrm{M}(t-1) \neq 0 \cup \mathrm{M}(t) \neq 0) \\ & \cap (\mathrm{M}(t-1) \in \mathcal{N}_1 \cup \mathrm{M}(t) \in \mathcal{N}_1) \big\} C \\ & + I\big\{ (\mathrm{M}(t-1) \in \mathcal{N}_1 \cap \mathrm{M}(t) \in \mathcal{N}_2) \\ & \cup (\mathrm{M}(t-1) \in \mathcal{N}_2 \cap \mathrm{M}(t) \in \mathcal{N}_1) \big\} \Delta C, \end{aligned} \tag{5}$$

where $\mathrm{M}(t) \in \mathcal{N}_1 \cup \mathcal{N}_2 \cup \{0\}$, $\mathrm{M}(t) = 0$ denotes the local computing, $\mathrm{M}(t) = 1, 2, \ldots, N$ denotes the MEC server, and $I\{\cdot\} = 1$ if the condition in $\{\cdot\}$ is satisfied and otherwise, $I\{\cdot\} = 0$. C denotes the migration time cost, which is replaced by $\rho T_{\mathrm{MEC}}(t)$ with the scaling factor ρ ($\rho \in [0, 1]$) in this paper. ΔC is the extra migration cost between the cellular MEC server and the satellite-based MEC server.

C. Velocity Control Model

We first assume that the robot has a preplanned timestamped reference trajectory to reduce its mobility model to one-dimensional motion, that is, velocity control without considering direction. Then, we assume that $v_n(l_0)$ and $v_n(l_{\mathrm{end}})$ are the initial velocity and final velocity in the nth AP coverage region ($n = 1, 2, \ldots, N$), respectively. We also assume $v_n(l_{\mathrm{end}}) = v_{\mathrm{goal},n}$ with the target velocity $v_{\mathrm{goal},n} \in [v_{\min}, v_{\max}]$ and a given acceleration $a > 0$. Note that the target velocity $v_{\mathrm{goal},n}$ is not known in advance and needs to be solved in the joint optimization problem in the next section. When the mobile robot enters the nth AP coverage region, its velocity control policy should be first determined by the relationship between $v_n(l_0)$ and $v_{\mathrm{goal},n}$. Thus, the instantaneous velocity $v_n(l)$ in the lth interval of the nth AP coverage region can be expressed as

$$v_n(l) = \begin{cases} \min\{v_n(l_0) + al, v_{\mathrm{goal},n}\}, & v_n(l_0) < v_{\mathrm{goal},n}, \\ v_n(l_0), & v_n(l_0) = v_{\mathrm{goal},n}, \\ \max\{v_n(l_0) - al, v_{\mathrm{goal},n}\}, & v_n(l_0) > v_{\mathrm{goal},n}, \end{cases} \tag{6}$$

where $v_n(l_0) < v_{\mathrm{goal},n}$, $v_n(l_0) = v_{\mathrm{goal},n}$, and $v_n(l_0) > v_{\mathrm{goal},n}$ correspond to the velocity controls of acceleration, constant movement, and deceleration, $l \in \mathcal{L}_n = \{1, 2, \ldots, L_n\}$, and L_n is the number of offloading intervals in the nth AP coverage region, which is calculated by

$$L_n = \left\lfloor \frac{T_{\mathrm{goal},n}}{\Delta T} \right\rfloor, \tag{7}$$

where $T_{\mathrm{goal},n}$ is the travel time across the nth AP coverage region, given as

$$T_{\mathrm{goal},n} = \begin{cases} \frac{1}{v_{\mathrm{goal},n}} \left(c_n + \frac{(v_{\mathrm{goal},n} - v_n(l_0))^2}{2a} \right), & v_n(l_0) \leqslant v_{\mathrm{goal},n}, \\ \frac{1}{v_{\mathrm{goal},n}} \left(c_n - \frac{(v_n(l_0) - v_{\mathrm{goal},n})^2}{2a} \right), & v_n(l_0) > v_{\mathrm{goal},n}, \end{cases} \tag{8}$$

where c_n is the moving distance in the nth AP coverage region.

Therefore, when the mobile robot passes through the nth AP, for $t = 1, 2, \ldots, \sum_{n=1}^{N} L_n$, the task completion time $T_n(t)$ in the tth interval can be formulated as

$$T_n(t) = T_{\mathrm{local}}(t) + \mu_n \left(T_{\mathrm{com}}(t) + T_{\mathrm{MEC}}(t) + T_{\mathrm{mig}}(t) \right), \tag{9}$$

where $\mu_n = 0$ and 1 denote the states when all wireless channels in the nth AP are unavailable or available, respectively. In the entire journey, the average task completion time is expressed as

$$T_{\mathrm{mean}} = \frac{1}{\sum_{n=1}^{N} L_n} \sum_{n=1}^{N} \sum_{l=1}^{L_n} T_n(l). \tag{10}$$

III. Joint Mobility Control and MEC Offloading

A. Optimization Problem

To enhance the QoS for delay-sensitive applications, the optimization problem is formulated as

$$\min_{v_{\mathrm{goal},n}, \mathbf{1}_\alpha(m,t)} T_{\mathrm{mean}} \tag{11a}$$

$$\text{s.t.} \quad T_n(t) \leqslant T_{n,\max}(t) \tag{11b}$$

$$\sum_{n=1}^{N} T_{\mathrm{goal},n} \leqslant T_{\mathrm{move}} \tag{11c}$$

$$\sum_{m=1}^{N} \mathbf{1}_\alpha(m,t) \leqslant 1 \tag{11d}$$

$$v_{\mathrm{goal},n} \in [v_{\min}, v_{\max}] \tag{11e}$$

$$\mathbf{1}_\alpha(m,t) \in \{0, 1\} \tag{11f}$$

Constraint (11b) indicates that the task completion time cannot be larger than the maximal completion time, where $T_{n,\max}(t)$ is the computation time for all data to be computed locally. Constraint (11c) denotes the tolerable total moving time across the whole journey. Constraint (11d) guarantees that only one MEC server is selected or only one local computation is performed per offloading interval. Constraints (11e) and (11f) involve the decisions of the velocity control and task offloading.

Theorem 1: The problem in (11) is NP-hard.

Proof: See Appendix A. ∎

Furthermore, for the process-oriented optimization problem (11), due to the robot's mobility, time-varying wireless channel, and dynamic computing capability, conventional optimization methods are computationally inefficient. Machine learning is a promising alternative [44], [45], such as Q-learning. However, the conventional Q-learning method cannot be directly employed to solve Problem (11). The first reason is the curse of dimensionality caused by the large-scale state space and action space. The second reason involves asynchronous actions, where the offloading decision is updated per offloading interval, and the velocity control decision is made per AP coverage region. Thus, based on Lyapunov optimization [17], decomposing the optimization problem (11) over the whole journey into multiple subproblems over each AP coverage region yields

$$\min_{v_{\mathrm{goal},n}, \alpha_m(l)} \frac{1}{L_n} \sum_{l=1}^{L_n} T_n(l) \tag{12a}$$

$$\text{s.t.} \quad T_{\mathrm{goal},n} \leqslant k_n T_{\mathrm{move}} \tag{12b}$$

$$(11\text{b}), (11\text{d}) - (11\text{f}) \tag{12c}$$

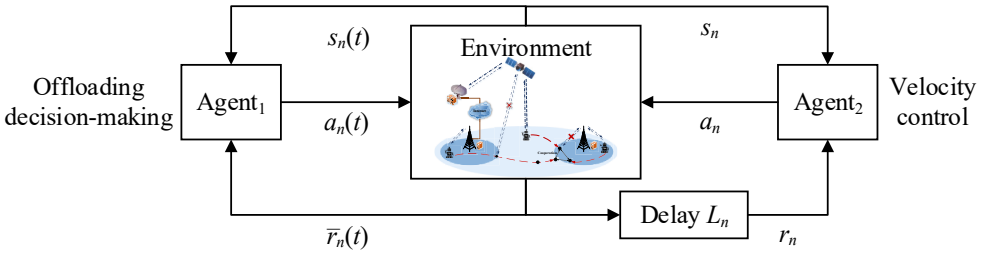


Fig. 2. The framework of dual-agent Q-learning.

where $k_n = c_n / \sum_{n=1}^{N} c_n$.

Theorem 2: Problem (12) is NP-hard.

Proof: See Appendix B. ■

B. Optimization Based on Dual-Agent Q-Learning

In this subsection, as shown in Fig. 2, a general framework of dual-agent Q-learning is provided to optimize the decision-making of offloading and velocity control in (12). In the nth AP coverage region, Agent_1 makes the offloading decision. Via accumulating the offloading reward and observing the channel state, the target velocity is determined by Agent_2. It is implied that Agent_1 is the local agent, and Agent_2 is the global agent.

Based on the MDP, the states, actions, and rewards for offloading-related Q-learning and mobility-related Q-learning are modeled below.

In offloading-based Q-learning, the state includes: the current AP coverage region $\mathrm{AP}(t)$, current channel state μ_n, data size $D(t)$, CPU frequency of the mobile robot $f_{\mathrm{local}}(t)$, current velocity $v_n(t)$, and previous MEC server $\mathrm{M}(t-1)$, given as

$$s_n(t) = \{\mathrm{AP}(t), \mu_n, D(t), f_{\mathrm{local}}(t), v_n(t), \mathrm{M}(t-1)\}. \quad (13)$$

The action includes the offloaded data ratio and current MEC server, which is expressed as

$$a_n(t) = \{\mathbf{1}_\alpha(m,t)\}. \quad (14)$$

In the reward design, the same reward is obtained in the velocity control when all possible target velocities $v_{\mathrm{goal},n}$ satisfy the moving constraint (12b). Thus, the instantaneous reward for the velocity control is designed as $\max\left\{\frac{1}{L_n}\left(T_{\mathrm{goal},n} - k_n T_{\mathrm{move}}\right), 0\right\}$. According to (7), this reward can be simplified as $\max\left\{\Delta T - \frac{k_n}{L_n} T_{\mathrm{move}}, 0\right\}$. By combining the offloading reward and velocity control reward, the instantaneous reward $\bar{r}_n(t)$ is given as

$$\begin{aligned}\bar{r}_n(t) =& (1-\theta)\exp\left(1 - \frac{T_n(t)}{T_{n,\max}(t)}\right) \\ &+ \theta \exp\left(1 - \frac{\max\left\{\Delta T - \frac{k_n}{L_n} T_{\mathrm{move}}, 0\right\}}{\frac{k_n}{L_n}\left(T_{\mathrm{low}} - T_{\mathrm{move}}\right)}\right), \end{aligned} \quad (15)$$

where θ is the preference factor between offloading and velocity control, $T_{\mathrm{low}} = \sum_{n=1}^{N} c_n / v_{\min}$. Finally, when legal action is obtained in the training, the reward in (15) will be employed. On the contrary, when an illegal action occurs, the reward value is set to -1. Thus, in the training, the instantaneous reward is expressed as

$$\bar{r}_n(t) = \begin{cases} (15), & \text{legal action}, \\ -1, & \text{illegal action}. \end{cases} \quad (16)$$

In mobility-controlled Q-learning, the state includes: the previous AP coverage region AP_{n-1}, the current AP coverage region AP_n, and the initial velocity $v_n(l_0)$, which is expressed as

$$s_n = \{\mathrm{AP}_{n-1}, \mathrm{AP}_n, v_n(l_0)\}. \quad (17)$$

The action is the target velocity, expressed as

$$a_n = \{v_{\mathrm{goal},n}\}. \quad (18)$$

The reward is the accumulated instantaneous reward $\bar{r}_n(t)$ in an AP coverage region, formulated as

$$r_n = \sum_{l=1}^{L_n} \bar{r}_n(l). \quad (19)$$

The details of the dual-agent Q-learning algorithm for joint velocity control and task offloading are shown in Algorithm 1. In this algorithm, the information of the robot-located coverage region (sequentially increasing from 1 to N) and offloaded data size is leveraged for service migration.

Remark: The relationship between task offloading and robot mobility is shown by the difference $k_n T_{\mathrm{move}} - \Delta T L_n$. Since the increased velocity decreases the number of offloading intervals L_n, when the reduced offloading time is smaller than the moving time per AP coverage region, that is, $\Delta T L_n < k_n T_{\mathrm{move}}$, a positive reward can be obtained. Otherwise, a negative/zero reward will be incurred.

In the following, the convergence of Algorithm 1 will be analyzed. According to [46], when $\sum_{j=1}^{T_{\mathrm{Epi}}} \lambda_j = \infty$ and $\sum_{j=1}^{T_{\mathrm{Epi}}} \lambda_j^2 < \infty$, the Q function can converge to the optimal Q function Q^* based on the following update rule

$$\begin{aligned} Q_{j+1}(s_j, a_j) =& Q_j(s_j, a_j) + \lambda_j\big(r_j + \max_{b \in \mathcal{A}} Q_j(s_{j+1}, b) \\ &- Q_j(s_j, a_j)\big), \end{aligned} \quad (20)$$

where $\mathcal{A}$ is the set of action spaces. In our proposed Algorithm 1, two Q functions are updated, where one is for task offloading and the other is for velocity control. Since the update rule in (20) is utilized for two Q functions and $\lambda_j = \lambda$ $(0 < \lambda < 1)$, the optimal Q functions can be obtained separately from these two Q tables. Thus, Algorithm 1 is convergent. In the sequel, we specify the convergence of Algorithm 1.

Algorithm 1 Joint Task Offloading and Velocity Control Based on a Dual-Agent Q-Learning Algorithm

Input: Initialize the table entry $Q_1(s,a)=0$ and $Q_2(s,a)=0$, velocity range $[v_{\min}, v_{\max}]$, moving distance c_n, learning rate λ, greedy factor ϵ, discount factor γ.

Output: Offloading decision $\mathbf{1}_\alpha(m,t)$, target velocity $v_{\text{goal},n}$.

for $j=1,2,\ldots,T_{\text{Epi}}$ **do**
  Reset $t=1$ and $s_n(t)$;
  for $n=1,2,\ldots,N$ **do**
    Observe state s_n;
    Chose action a_n with ϵ-greedy algorithm;
    while $\text{AP}(t)=n$ **do**
      Observe state $s_n(t)$;
      Chose action $a_n(t)$ with ϵ-greedy algorithm;
      Calculate reward $\bar{r}_n(t)$ and next state $s_n(t+1)$;
      Update the Q-table for task offloading:

$$\begin{aligned} Q_1(s_n(t),a_n(t)) =& Q_1(s_n(t),a_n(t))+\lambda\big(\bar{r}_n(t)+\gamma \\ &\cdot \max Q_1(s_n(t+1),a_n(t+1)) \\ &- Q_1(s_n(t),a_n(t))\big) \end{aligned} \tag{21}$$

      Update state $s_n(t)=s_n(t+1)$;
      $t=t+1$;
    end while
    Calculate reward r_n and next state s_{n+1};
    Update the Q-table for velocity control:

$$\begin{aligned} Q_2(s_n,a_n) =& Q_2(s_n,a_n)+\lambda\big(r_n-Q_2(s_n,a_n) \\ &+\gamma\max Q_2(s_{n+1},a_{n+1})\big) \end{aligned} \tag{22}$$

    Update state $s_n=s_{n+1}$.
  end for
end for

According to [47], we have the first convergence theorem as follows.

Theorem 3: Assume that $\|Q_{1,1}(s_n(t),a_n(t))\| \leqslant \frac{r_{\max}}{1-\gamma}$ and $\|Q_{2,1}(s_n,a_n)\| \leqslant \frac{L_{\max}r_{\max}}{1-\gamma}$. Then, one has

$$\|Q_{1,j}(s_n(t),a_n(t))\| \leqslant \frac{r_{\max}}{1-\gamma}, \tag{23}$$

$$\|Q_{1,j}(s_n(t),a_n(t))-Q_1^*\| \leqslant \frac{2r_{\max}}{1-\gamma}, \tag{24}$$

$$\|Q_{2,j}(s_n,a_n)\| \leqslant \frac{L_{\max}r_{\max}}{1-\gamma}, \tag{25}$$

$$\|Q_{2,j}(s_n,a_n)-Q_2^*\| \leqslant \frac{2L_{\max}r_{\max}}{1-\gamma}, \tag{26}$$

where $r_{\max}=\max_n\|\bar{r}_n(t)\|$, $L_{\max}=\max_n L_n$, $j=1,2,\ldots,T_{\text{Epi}}$, $n=1,2,\ldots,N$, Q_1^* and Q_2^* denote the optimal functions of $Q_{1,j}(s_n(t),a_n(t))$ and $Q_{2,j}(s_n,a_n)$, respectively.

Proof: Mathematical induction is employed to prove Theorem 3. First, for $j=1$, the two initialized Q tables satisfy $\|Q_{1,1}(s_n(t),a_n(t))\| \leqslant \frac{r_{\max}}{1-\gamma}$ and $\|Q_{2,1}(s_n,a_n)\| \leqslant \frac{L_{\max}r_{\max}}{1-\gamma}$. For example, the initial values of two Q tables can be from the intervals $\left[-\frac{r_{\max}}{1-\gamma},\frac{r_{\max}}{1-\gamma}\right]$ and $\left[-\frac{L_{\max}r_{\max}}{1-\gamma},\frac{L_{\max}r_{\max}}{1-\gamma}\right]$. When $\|Q_{1,j}(s_n(t),a_n(t))\| \leqslant \frac{r_{\max}}{1-\gamma}$, for the $(j+1)$th iteration, we derive

$$\begin{aligned} &\|Q_{1,j+1}(s_n(t),a_n(t))\| \\ &\leqslant (1-\lambda)\|Q_{1,j}(s_n(t),a_n(t))\|+\lambda\|\bar{r}_n(t)\| \\ &\quad+\lambda\gamma\max_{a_n(t+1)\in\mathcal{A}_1}\|Q_{1,j}(s_n(t+1),a_n(t+1))\| \\ &\leqslant \frac{r_{\max}}{1-\gamma}+\lambda r_{\max}+\lambda\gamma\frac{r_{\max}}{1-\gamma} \\ &= \frac{r_{\max}}{1-\gamma}. \end{aligned} \tag{27}$$

Thus, (23) is proven.

Similarly, when $\|Q_{2,j}(s_n,a_n)\| \leqslant \frac{L_{\max}r_{\max}}{1-\gamma}$, for the $(j+1)$th iteration, we obtain

$$\begin{aligned} \|Q_{2,j+1}(s_n,a_n)\| &\leqslant (1-\lambda)\|Q_{2,j}(s_n,a_n)\|+\lambda\|r_n\| \\ &\quad+\lambda\gamma\max_{a_{n+1}\in\mathcal{A}_2}\|Q_{2,j}(s_{n+1},a_{n+1})\| \\ &\leqslant \frac{L_{\max}r_{\max}}{1-\gamma}+\lambda L_{\max}r_{\max}+\lambda\gamma\frac{L_{\max}r_{\max}}{1-\gamma} \\ &= \frac{L_{\max}r_{\max}}{1-\gamma}. \end{aligned} \tag{28}$$

Thus, (25) is proven.

Based on the above proof, (24) and (26) can be proven as

$$\begin{aligned} \|Q_{1,j}(s_n(t),a_n(t))-Q_1^*\| &\leqslant \|Q_{1,j}(s_n(t),a_n(t))\|+\|Q_1^*\| \\ &\leqslant \frac{2r_{\max}}{1-\gamma}, \end{aligned} \tag{29}$$

$$\|Q_{2,j}(s_n,a_n)-Q_2^*\| \leqslant \|Q_{2,j}(s_n,a_n)\|+\|Q_2^*\| \leqslant \frac{2L_{\max}r_{\max}}{1-\gamma}. \tag{30}$$

This completes the proof. ■

Theorem 3 shows that, with the reduced value of γ, the convergence performance of Algorithm 1 will be improved. Based on [48], Theorem 3 can be extended to the second convergence theorem. First, the Bellman operation $\mathcal{T}\{\cdot\}$ is defined as

$$\mathcal{T}\{Q(s,a)\}=\sum_{s'\in\mathcal{S}}p_a(s,s')\Big(r(s,a)+\gamma\max_{a'\in\mathcal{A}}Q(s',a')\Big), \tag{31}$$

where $p_a(s,s')$ is the state transition probability from State s to State s' and $\mathcal{S}$ is the set of state spaces. Then, an approximation error $\theta_{i,j}$ $(i=1,2, j\in[1,T_{\text{Epi}}])$ is defined as

$$E\left\{\|Q_{i,j+1}-\mathcal{T}\{Q_{i,j}\}\|_2^2\right\}\leqslant\theta_{i,j}. \tag{32}$$

Based on Theorem 3, we assume that $\theta_{2,j}=L_{\max}\theta_{1,j}$.

Theorem 4: The convergence of the improved Q-learning in Algorithm 1 can be expressed as

$$\begin{aligned} &E\left\{\left\|Q_{1,T_{\text{Epi}}}(s_n(t),a_n(t))-Q_1^*\right\|_\infty\right\} \\ &\leqslant\sum_{j=1}^{T_{\text{Epi}}}\gamma^{T_{\text{Epi}}-j}\sqrt{\theta_{1,j}}+\gamma^{T_{\text{Epi}}}E\left\{\|Q_{1,0}(s_n(t),a_n(t))-Q_1^*\|_\infty\right\}, \end{aligned} \tag{33}$$

$$E\left\{\left\|Q_{2,T_{\rm Epi}}(s_n,a_n)-Q_2^*\right\|_\infty\right\} \leqslant \sum_{j=1}^{T_{\rm Epi}}\gamma^{T_{\rm Epi}-j}\sqrt{L_{\max}\theta_{1,j}}+\gamma^{T_{\rm Epi}}E\left\{\left\|Q_{2,0}(s_n,a_n)-Q_2^*\right\|_\infty\right\}. \quad (34)$$

When $\theta_{1,j}=\theta$, (33) and (34) can be rewritten by

$$E\left\{\left\|Q_{1,T_{\rm Epi}}(s_n(t),a_n(t))-Q_1^*\right\|_\infty\right\} \leqslant \frac{\sqrt{\theta}}{1-\gamma}+\gamma^{T_{\rm Epi}}E\left\{\left\|Q_{1,0}(s_n(t),a_n(t))-Q_1^*\right\|_\infty\right\}, \quad (35)$$

$$E\left\{\left\|Q_{2,T_{\rm Epi}}(s_n,a_n)-Q_2^*\right\|_\infty\right\} \leqslant \frac{\sqrt{L_{\max}\theta}}{1-\gamma}+\gamma^{T_{\rm Epi}}E\left\{\left\|Q_{2,0}(s_n,a_n)-Q_2^*\right\|_\infty\right\}. \quad (36)$$

Proof: Considering the γ-contraction property of the Bellman operator and $Q_i^*=\mathcal{T}\{Q_i^*\}$ in [47], we can derive

$$\begin{aligned}&E\left\{\left\|Q_{i,j+1}-Q_i^*\right\|_\infty\right\}\\ &\leqslant E\left\{\left\|Q_{i,j+1}-\mathcal{T}\left\{Q_{i,j}\right\}\right\|_\infty\right\}+E\left\{\left\|\mathcal{T}\left\{Q_{i,j}\right\}-Q_i^*\right\|_\infty\right\}\\ &\leqslant \sqrt{E\left\{\left\|Q_{i,j+1}-\mathcal{T}\{Q_{i,j}\}\right\|_2^2\right\}}+E\left\{\left\|\mathcal{T}\left\{Q_{i,j}\right\}-\mathcal{T}\left\{Q_i^*\right\}\right\|_\infty\right\}\\ &\leqslant \sqrt{\theta_{i,j+1}}+\gamma E\left\{\left\|Q_{i,j}-Q_i^*\right\|_\infty\right\}. \end{aligned} \quad (37)$$

Based on (37), we can derive

$$\begin{aligned}&E\left\{\left\|Q_{i,T_{\rm Epi}}-Q_i^*\right\|_\infty\right\}\\ &\leqslant \sqrt{\theta_{i,T_{\rm Epi}}}+\gamma E\left\{\left\|Q_{i,T_{\rm Epi}-1}-Q_i^*\right\|_\infty\right\}\\ &\leqslant \sqrt{\theta_{i,T_{\rm Epi}}}+\gamma\left(\sqrt{\theta_{i,T_{\rm Epi}-1}}+\gamma E\left\{\left\|Q_{i,T_{\rm Epi}-2}-Q_i^*\right\|_\infty\right\}\right)\\ &\cdots\\ &\leqslant \sum_{j=1}^{T_{\rm Epi}}\gamma^{T_{\rm Epi}-j}\sqrt{\theta_{i,j}}+\gamma^{T_{\rm Epi}}E\left\{\left\|Q_{i,0}-Q_i^*\right\|_\infty\right\}. \end{aligned} \quad (38)$$

Since $\theta_{2,j}=L_{\max}\theta_{1,j}$, based on (38), (33) and (34) can be obtained.

When $\theta_{i,j}=\theta$, (38) is simplified as

$$E\left\{\left\|Q_{i,T_{\rm Epi}}-Q_i^*\right\|_\infty\right\} \leqslant \frac{1-\gamma^{T_{\rm Epi}}}{1-\gamma}\sqrt{\theta}+\gamma^{T_{\rm Epi}}E\left\{\left\|Q_{i,0}-Q_i^*\right\|_\infty\right\}. \quad (39)$$

Since $0\leqslant\gamma<1$, when $T_{\rm Epi}\to\infty$, we have $\gamma^{T_{\rm Epi}}\to\infty$. Thus, (39) is reduced to

$$E\left\{\left\|Q_{i,T_{\rm Epi}}-Q_i^*\right\|_\infty\right\}\leqslant\frac{\sqrt{\theta}}{1-\gamma}+\gamma^{T_{\rm Epi}}E\left\{\left\|Q_{i,0}-Q_i^*\right\|_\infty\right\}. \quad (40)$$

Finally, according to (40), (35) and (36) can be proved. ■

Theorem 4 shows that the convergence of the proposed algorithm is affected by four factors: 1) the approximation error for Q_1; 2) the approximation calculation of the Bellman operator for Q_1; 3) the approximation error for Q_2; and 4) the approximation calculation of the Bellman operator for Q_2.

In the following, the complexity of the proposed dual-agent Q-learning is analyzed in terms of sample complexity. The sample complexity is referred to as the total number of samples needed to yield an entrywise ξ_1-accurate approximation of the optimal Q-function, to satisfy $\max_{s,a}\|Q(s,a)-Q^*(s,a)\|\leqslant\xi_1$ (or $E\{\max_{s,a}\|Q(s,a)-Q^*(s,a)\|\}\leqslant\xi_1$) with probability at least $1-\xi_2$ for any $\xi_2\in(0,1)$ [49]. According to [49], we assume a γ-discounted infinite-horizon MDP with state space $\mathcal{S}$ and action space $\mathcal{A}$. We also assume that the Markov chain induced by a behavior policy π_b is uniformly ergodic. The minimum state-action occupancy probability of the sample trajectory is defined as $\omega_{\min,i}=\min_{(s,a)\in\mathcal{S}_i\times\mathcal{A}_i}\omega_{\pi_b,i}(s,a)$, where $\omega_{\pi_b,i}$ denotes the stationary distribution of the Markov chain for the ith agent with $i=1,2$. Furthermore, the mixing time associated with the sample trajectory is defined as $t_{{\rm mix},i}=\min\left\{t\left|\max_{(s,a)\in\mathcal{S}_i\times\mathcal{A}_i}d_{\rm TV}\left(P^t(\cdot|s,a),\omega_{\pi_b,i}\right)\leq\frac{1}{4}\right.\right\}$, where $d_{\rm TV}(\omega,\nu)=0.5\sum_{x\in\mathcal{X}}|\omega(x)-\nu(x)|$ and $P^t(\cdot|s,a)$ indicates the distribution of (s_t,a_t) with the initialization of $(s_0,a_0)=(s,a)$. According to Theorem 3, in dual-agent Q-learning, the accuracy levels of the two agents are $\frac{2r_{\max}}{1-\gamma}$ and $\frac{2L_{\max}r_{\max}}{1-\gamma}$. According to the complexity analysis of asynchronous Q-learning in [49], the total sample complexity of dual-agent Q-learning can be approximated as the sum of the sample complexity of the offloading agent and that of the velocity control agent as

$$O\left(\frac{1}{4r_{\max}^2(1-\gamma)^2}\left(\frac{1}{\omega_{\min,1}}+\frac{1}{L_{\max}^2\omega_{\min,2}}\right)+\frac{1}{1-\gamma}\left(\frac{t_{\rm mix,1}}{\omega_{\min,1}}+\frac{t_{\rm mix,2}}{\omega_{\min,2}}\right)\right). \quad (41)$$

IV. Simulation Results and Discussion

In this section, we will evaluate the stochastic task offloading performance achieved from our proposed Q-learning-based algorithm.

In our simulation, 20 APs and MEC servers are deployed, where $\mathcal{N}_1=\{1,2,\ldots,7\}\cup\{14,15,\ldots,20\}$ and $\mathcal{N}_2=\{8,9,\ldots,13\}$. The computing capacities of cellular and satellite-based MEC servers are from finite sets $\{10,11,\ldots,19\}$ (GHz) and $\{50,51,\ldots,59\}$ (GHz). The moving distance c_n is randomly chosen from a set $\{100,200,300\}$ (m) for the cellular AP and from a set $\{1000,2000,3000\}$ (m) for the satellite. In the cellular network, the bandwidth is $W=10$ MHz, the transmit power of the mobile robot is $p=0.2$ W, the channel noise power is $\sigma^2=2\times10^{-12}$ W, and the channel power gain is $h^2=10^{-6}$. In the satellite communication, for simplicity, we set the distances as $d_{\rm GS}=d_{\rm SE}=1000$ km, and the transmission rates as $r_{\rm GS}=10$ Mbps and $r_{\rm SE}=100$ Mbps. The extra migration cost ΔC is set to the average delay of 500 ms. In each offloading interval $\Delta T=1$ s, the generated data size is randomly selected from the set $\{100,250,400,550,700\}$ (KB) with $\Phi=800$ CPU cycles/bit, and the computing capacity of the mobile robot is randomly chosen from a finite set $\{0.5,0.6,\ldots,1\}$ (GHz). The random waypoint model without considering direction [50] is used as the mobility model of the robot. For the mobile robot with $a=2$ m/s^2, its velocity is from a discrete set $\mathcal{V}=\{5,6,\ldots,20\}$ (m/s). In Q-learning, the hyperparameters are set as $\lambda=0.1$, $\gamma=0.9$, and $\epsilon=0.05$ with a discount interval of 4×10^{-6}.

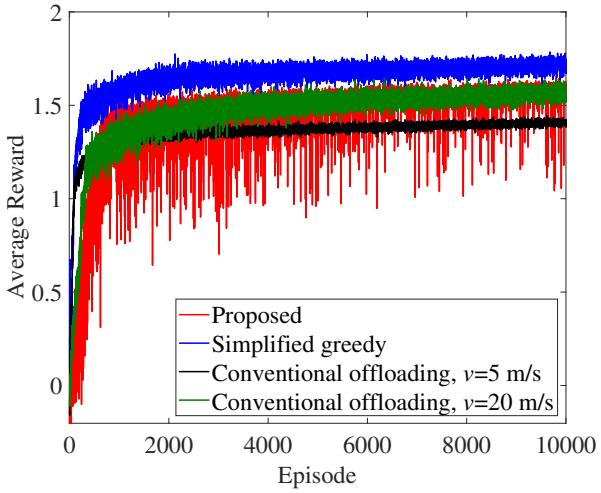


Fig. 3. Average reward in the training episode. The number of APs with unavailable wireless communication is $N_{\mathrm{CH}} = 4$. The migration ratio is $\rho = 0.1$. The moving factor is $\theta = 0.1$.

For a fair performance comparison, we simulate three baselines below.

- *Conventional Offloading:* The mobile robot has a constant velocity, as depicted in [30], and its offloading decision is made by Q-learning.
- *Local Execution:* All scheduled computation tasks are processed by a mobile robot with its available CPU frequency while maintaining a constant velocity.
- *Simplified Greedy:* Since conventional greedy searching for velocity decision-making has significant complexity, i.e., $O(|\mathcal{V}|^N)$, a simplified greedy algorithm using local searching for each AP is given for comparison. First, in the nth AP coverage region, the velocity with the maximal average of $\bar{r}_n(t)$ is searched from the candidate set $\mathcal{V}$. Then, upon searching the maximal average of r_n in all training, the target velocity for the whole trajectory will be selected.

In Fig. 3, the convergence of the reward function in (19) is plotted for the proposed scheme compared to conventional offloading and simplified greedy schemes. The number of cellular/satellite APs with unavailable wireless communication is set to $N_{\mathrm{CH}} = 4$. The migration ratio is set as $\rho = 0.1$. The moving factor is set as $\theta = 0.1$. As observed in Fig. 3, the proposed scheme has higher rewards than conventional offloading using low velocities and has close rewards to conventional offloading using the highest velocity. It also has slightly lower rewards than the simplified greedy algorithm. Thus, the proposed scheme can achieve convergence as the training increases, which is consistent with the convergence analysis in Section III-B.

In Fig. 4, the average task completion time of conventional offloading, local execution, simplified greedy, and the proposed schemes are plotted for different parameters. We first compare the proposed scheme with conventional schemes in Fig. 4(a) for different values of N_{CH}. It can be seen in Fig. 4(a) that as the value of N_{CH} increases, except for the local execution, the other schemes have increased task completion times. More importantly, the proposed scheme has a lower

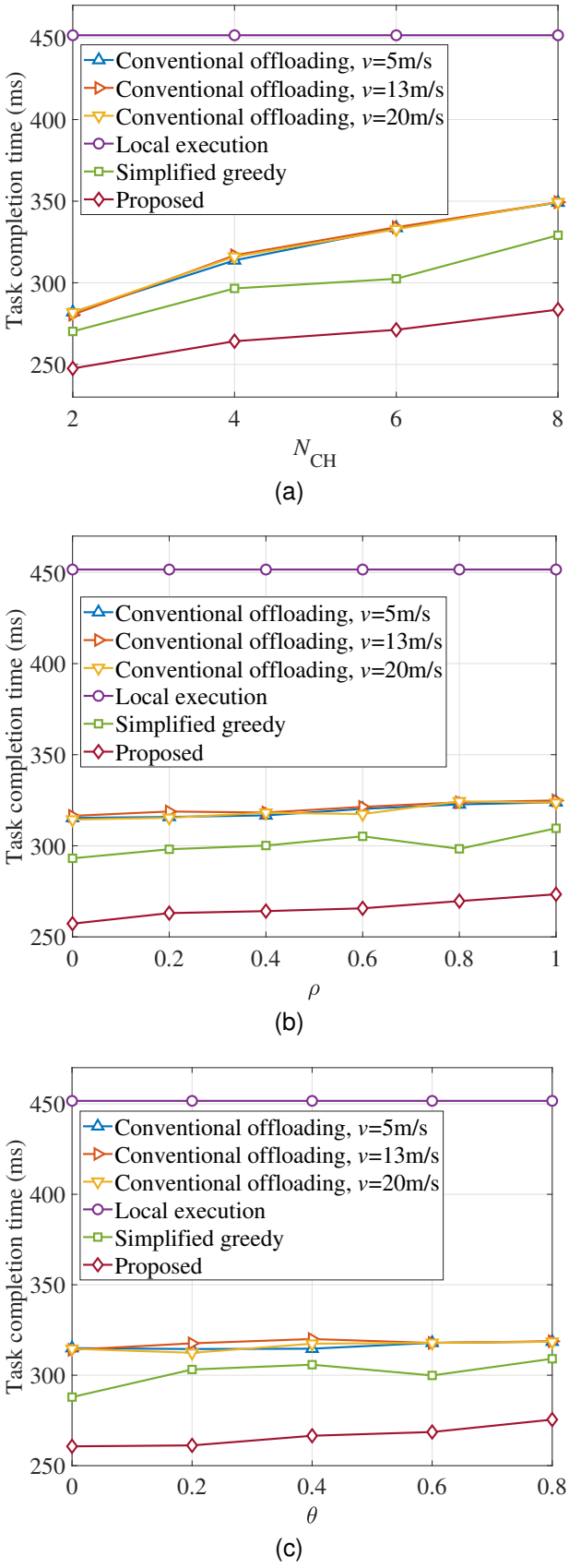


Fig. 4. Task completion time comparison among conventional offloading, local execution, simplified greedy, and the proposed schemes. (a) $\rho = 0.1$ and $\theta = 0.1$, (b) $N_{\mathrm{CH}} = 4$ and $\theta = 0.1$, (c) $N_{\mathrm{CH}} = 4$ and $\rho = 0.1$.

completion time than conventional schemes. For example, for $N_{\mathrm{CH}} = 4$, compared to conventional offloading, local execution, and simplified greedy, the completion time of the proposed scheme is reduced by approximately 16%, 41%, and 11%, respectively. We also plot the task completion time curves for varying migration factors ρ in Fig. 4(b). It is shown that upon increasing the migration ratio, a slightly increased task completion time is incurred. For varying values of ρ, the proposed scheme also has a much lower completion time than conventional offloading, local execution, and simplified greedy, with average reduction ratios of 17%, 41%, and 12%, respectively. Moreover, in Fig. 4(c), we plot the task completion time curves of these schemes for different moving factors θ. As seen from this figure, the moving factor has a slight effect on the task completion time of all MEC-based offloading schemes. In these schemes, the proposed scheme has the lowest task completion time, whose average reduction ratios over conventional offloading, local execution, and simplified greedy are 16%, 41%, and 11%, respectively. In addition, Fig. 4 shows that, since velocity control is not considered in the conventional offloading scheme, the conventional scheme has almost the same average task completion time for different velocities. To summarize, as opposed to conventional schemes, the proposed scheme can effectively reduce the service delay of MEC-based offloading for the hybrid satellite-terrestrial-network-enabled robot.

Fig. 5 portrays the task completion time versus the size of the data generated by the robot. The data size per offloading interval is randomly selected from the set $\{100, 250, 400, 550, 700\}(\mathrm{KB}) + \Delta D$, where the incremental parameter ΔD belongs to $\{1, 3, 5, 7, 9\}$ (MB). It is shown that for different data sizes, the proposed scheme has the lowest task completion time. Furthermore, compared to conventional schemes, when the data size increases, a much higher time consumption reduction can be achieved by the proposed scheme.

To show the effect of mobility control on service delay, we now investigate task completion time versus moving time in Fig. 6. It is noted that, when the marker size increases, the values of the parameters involving N_{CH}, ρ, and θ are increased. First, in Fig. 6(a), we compare the task completion time versus moving time performance for conventional offloading, local execution, simplified greedy, and the proposed schemes with different values of N_{CH}. It is shown that since a constant velocity is assumed in conventional offloading and local execution, the effect of the velocity control on the completion time performance cannot be clearly observed. Although the simplified greedy scheme has a lower moving time than the proposed scheme, it has a higher time consumption than the proposed scheme. Based on the joint optimization of task offloading and velocity control, the proposed scheme obtained the lowest task completion time at a moderate moving time. Fig. 6(a) also implies that the proposed scheme is sensitive to N_{CH}, since communication state-based velocity control is employed. Second, we portray a scatterplot to compare the task completion time versus moving time performance for different migration ratios in Fig. 6(b). Similar to 6(a), Fig. 6(b) shows that, compared to conventional schemes, the

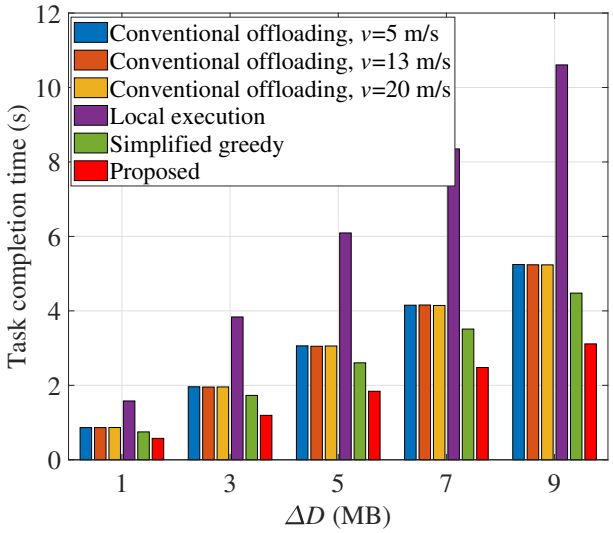
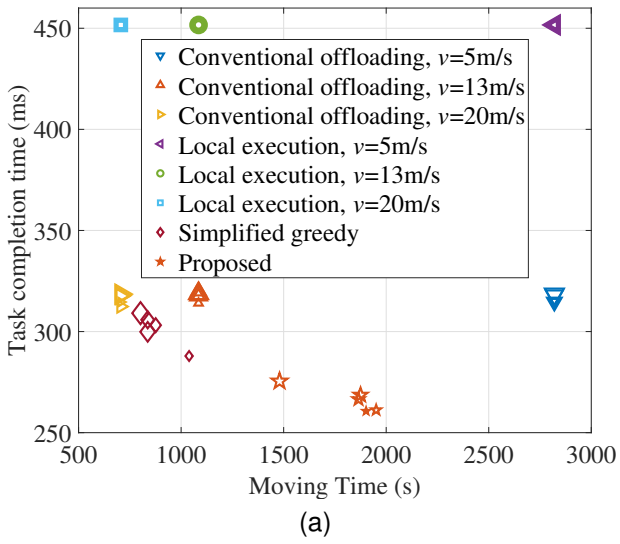


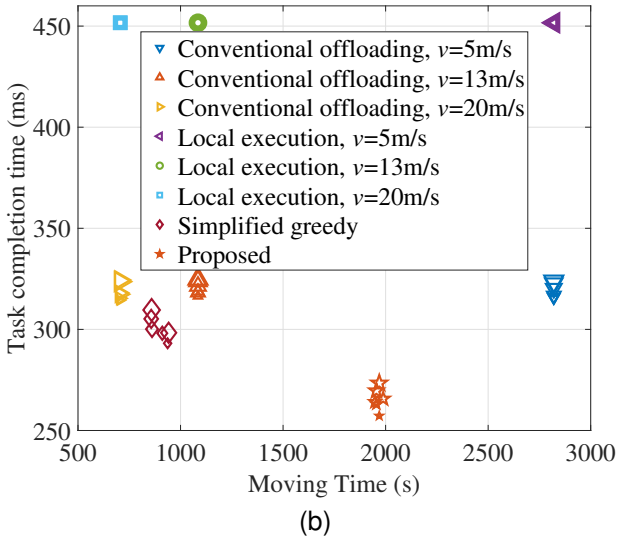
Fig. 5. Task completion time comparison among conventional offloading, local execution, simplified greedy, and the proposed schemes for different data sizes, where $N_{\mathrm{CH}} = 4$, $\rho = 0.1$, and $\theta = 0.1$.

proposed scheme can achieve the lowest time consumption at a moderate moving time. It is also shown that compared to the communication state, the migration ratio has a smaller effect on the moving time (corresponding to velocity control) in the proposed scheme. Finally, in Fig. 6(c), we plot the task completion time scatters versus moving time for different moving factors. As shown in Fig. 6(c), the proposed scheme can also achieve the lowest time consumption at a moderate moving time. Moreover, with an increased moving factor, the proposed scheme can efficiently reduce the moving time while maintaining the lowest service delay among these schemes. It is also shown in Fig. 6 that all conventional schemes are insensitive to the moving time, and the proposed scheme benefits from velocity control.

Furthermore, to verify the efficiency of the proposed scheme, two possible cases for the joint optimization of velocity control and task offloading are considered. In Case I, according to the availability of all wireless channels in each AP, the mobile robot directly makes a velocity control decision. For the channel states $\mu_n = 1$ and $\mu_n = 0$, we have $v_{\mathrm{goal},n} = v_{\min}$ and $v_{\mathrm{goal},n} = v_{\max}$, respectively. In Case II, based on the availability of all wireless channels per AP, the mobile robot directly makes a task offloading decision. For the channel states $\mu_n = 1$ and $\mu_n = 0$, we have $\mathbf{1}_\alpha(m,t) = 1$ with a given MEC server and $\mathbf{1}_\alpha(m,t) = 0$ with the local computation, respectively. Our proposed scheme corresponds to Case III. Different from the fixed velocity control in Case I and the fixed offloading decision in Case II, a flexible joint optimization is designed in the proposed scheme. Thus, Cases I and II can also be regarded as two special cases of Case III. In the following figures, the task completion time performance is comprehensively compared for these three cases.

In Fig. 7(a), we first compare the task completion times of these three cases for different values of N_{CH}, where $\rho = 0.1$ and $\theta = 0.1$. We also plot the time consumption curves for different migration ratios in Fig. 7(b), where $N_{\mathrm{CH}} = 4$ and $\theta = 0.1$. Moreover, we plot the time consumption curves for





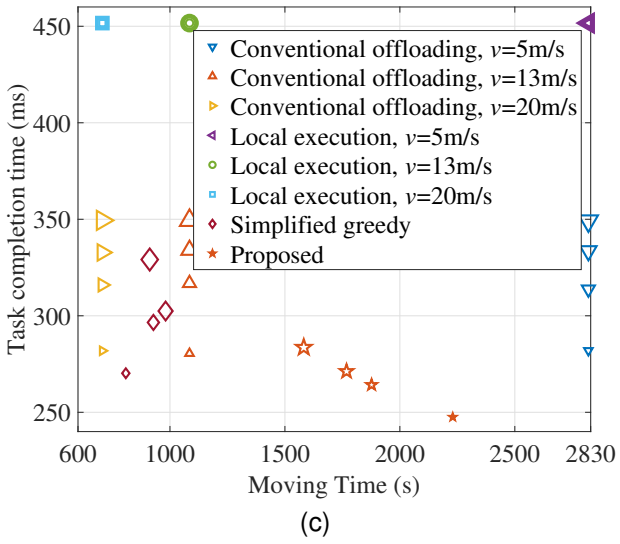
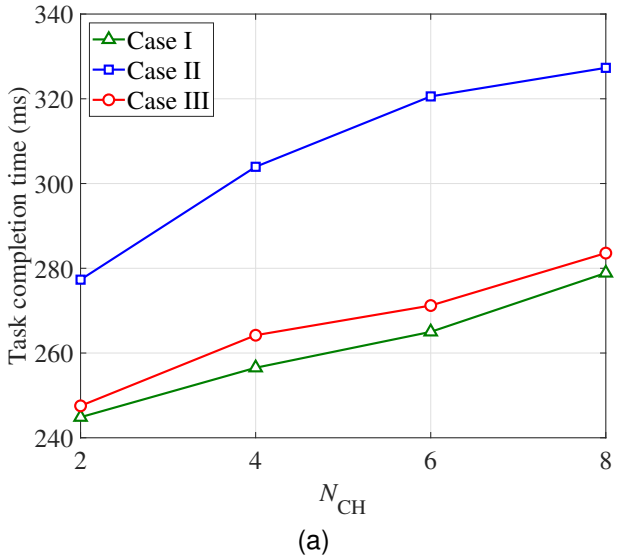
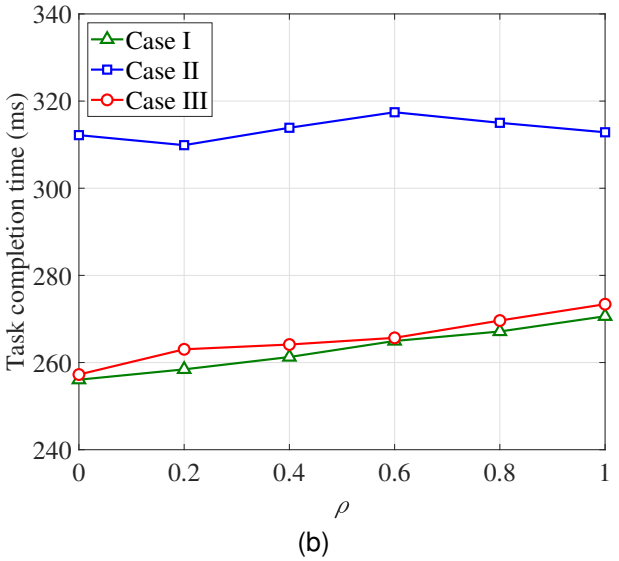


Fig. 6. Task completion time versus moving time among conventional offloading, local execution, simplified greedy, and the proposed schemes. (a) $N_{\mathrm{CH}} = 2, 4, 6, 8$, $\rho = 0.1$, and $\theta = 0.1$. When the marker size increases, the value of N_{CH} increases from 2 to 8. (b) $N_{\mathrm{CH}} = 4$, $\rho = 0, 0.2, 0.4, 0.6, 0.8, 1$, and $\theta = 0.1$. When the marker size increases, the value of ρ increases from 0 to 1. (c) $N_{\mathrm{CH}} = 4$, $\rho = 0.1$, and $\theta = 0, 0.2, 0.4, 0.6, 0.8$. When the marker size increases, the value of θ increases from 0 to 0.8.





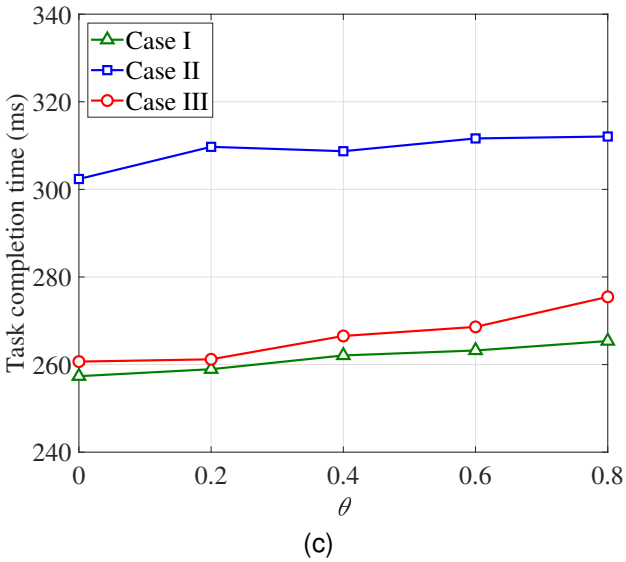


Fig. 7. Task completion time comparison of three cases for joint velocity control and task offloading. (a) $\rho = 0.1$ and $\theta = 0.1$. (b) $N_{\mathrm{CH}} = 4$ and $\theta = 0.1$. (c) $N_{\mathrm{CH}} = 4$ and $\rho = 0.1$.

different moving factors in Fig. 7(c), where $N_{\mathrm{CH}} = 4$ and $\rho = 0.1$. Observe from Fig. 7 that as the parameter values involving N_{CH}, ρ, and θ increase, all task completion times increase. Furthermore, Case II has a much higher service delay than Cases I and III, and Case III has a slightly higher service delay than Case I.

From Fig. 8, the task completion time versus moving time performance is compared for Cases I, II, and III. The parameters are set as follows: 1) $N_{\mathrm{CH}} = 2, 4, 6, 8$, $\rho = 0.1$, and $\theta = 0.1$ in Fig. 8(a); 2) $N_{\mathrm{CH}} = 4$, $\rho = 0, 0.2, 0.4, 0.6, 0.8, 1$, and $\theta = 0.1$ in Fig. 8(b); and 3) $N_{\mathrm{CH}} = 4$, $\rho = 0.1$, and $\theta = 0, 0.2, 0.4, 0.6, 0.8$ in Fig. 8(c). As shown in Fig. 8(a), although Case II has a lower moving time than Cases I and III, it has a much higher time completion time than Cases I and III. Furthermore, Case III can obtain a lower moving time than Case I at the price of a slightly increased task completion time. In addition, Figs. 8(b) and 8(c) show that due to the fixed number of APs with unavailable wireless communication, the velocity control of Case III lacks flexibility. As a result, Case III cannot adaptively reduce the moving time. In contrast, Case III can dramatically reduce the moving time, especially for a large value of the moving factor shown in Fig. 8(c), while maintaining a close task completion time to Case I.

In summary, the proposed scheme can create an elegant balance between service delay reduction and moving time reduction, and thus, it is a better choice for task offloading and velocity control compared to conventional schemes.

V. Conclusion

In this paper, a joint optimization problem of velocity control and task offloading has been proposed in a hybrid satellite-terrestrial network with multiple MEC servers. To reduce the service delay for a mobile robot caused by increased local computations and frequent service migrations, the effect of wireless communication availability and velocity control on task offloading has been studied. The analytical results of convergence rate and sample complexity of the improved Q-learning algorithm have been obtained. Simulation results have shown that, unlike conventional counterparts, based on velocity control, the proposed scheme can obtain an effective offloading performance improvement in terms of the task completion time. It was found that for mobile robots, mobility control is beneficial for providing high-quality service offloading in complex network environments. However, further study is required to determine its effectiveness in the scenario of multiple mobile robots for multiple cooperative missions. While this paper has only considered the decision-making of offloading and velocity control for one robot in the multi-robot environment, other forms of dynamics, such as time-varying bandwidth, dynamic computational resources, and complex trajectory planning, have considerable impact on MEC-based offloading. These issues will constitute the direction of our future work.

Appendix A
Proof of Theorem 1

We first assume that the mobile robot has a constant velocity, that is, $\beta_n = 0$, while the constraint $T_{\mathrm{goal},n} \leqslant k_n T_{\mathrm{move}}$ is

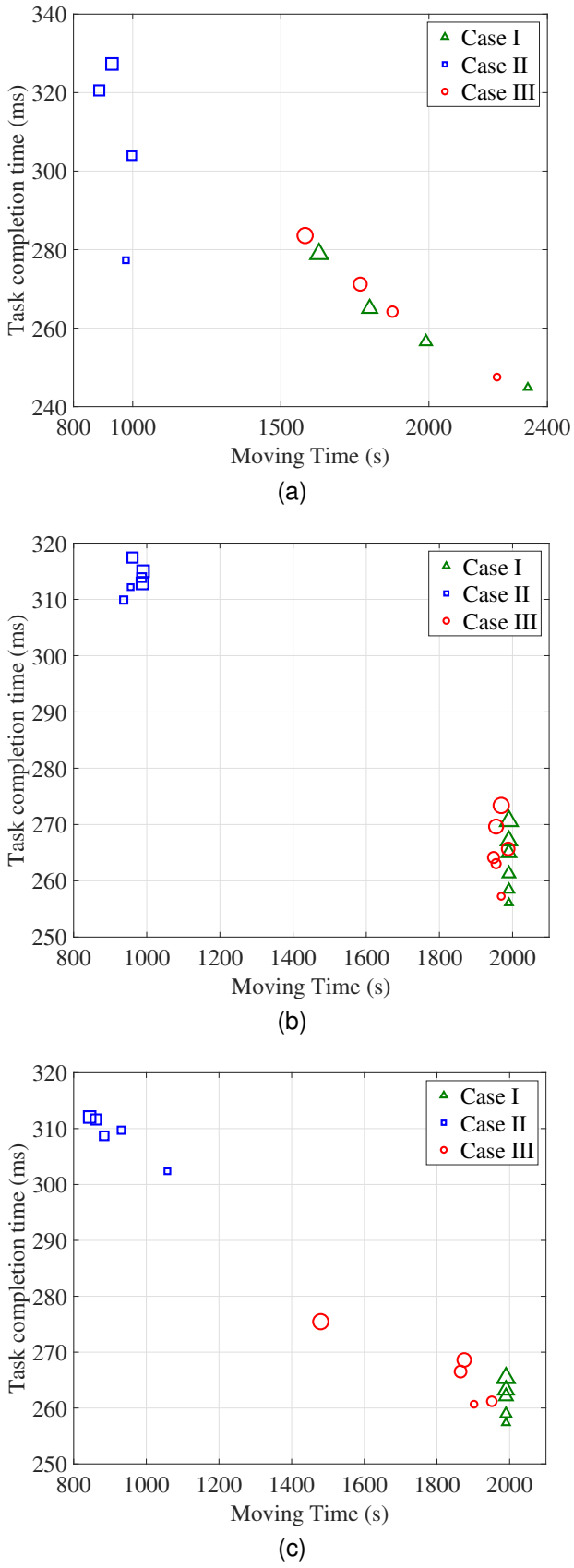


Fig. 8. Task completion time versus moving time among three cases for joint velocity control and task offloading. (a) $N_{\mathrm{CH}} = 2, 4, 6, 8$, $\rho = 0.1$, and $\theta = 0.1$. When the marker size increases, the value of N_{CH} increases from 2 to 8. (b) $N_{\mathrm{CH}} = 4$, $\rho = 0, 0.2, 0.4, 0.6, 0.8, 1$, and $\theta = 0.1$. When the marker size increases, the value of ρ increases from 0 to 1. (c) $N_{\mathrm{CH}} = 4$, $\rho = 0.1$, and $\theta = 0, 0.2, 0.4, 0.6, 0.8$. When the marker size increases, the value of θ increases from 0 to 0.8.

satisfied. Thus, (11) is simplified as

$$\min_{\mathbf{1}_\alpha(m,t)} T_{\text{mean}} \tag{42a}$$
$$\text{s.t.} \quad T_n(t) \leqslant T_{n,\max}(t) \tag{42b}$$
$$\sum_{m=1}^{N} \mathbf{1}_\alpha(m,t) \leqslant 1 \tag{42c}$$
$$\mathbf{1}_\alpha(m,t) \in \{0,1\} \tag{42d}$$

According to [51], the generalized assignment problem (GAP) can be formulated as

$$\min_{x_{ij}} \sum_i \sum_j c_{ij} x_{ij} \tag{43a}$$
$$\text{s.t.} \quad \sum_j a_{ij} x_{ij} \leqslant b_i \tag{43b}$$
$$\sum_i x_{ij} = 1 \tag{43c}$$
$$x_{ij} \in \{0,1\} \tag{43d}$$

where c_{ij} is the cost of assigning Task j to Agent i, a_{ij} denotes the required capacity when Task j is assigned to Agent i, and b_i is the available capacity of Agent i. Based on [51], [52], GAP is NP-hard.

If we set $x_{ij} = \mathbf{1}_\alpha(m,t)$, $b_i = \sum_{l=1}^{L_n} T_{n,\max}(l)$, $c_{ij} = T_n(t)/(\mathbf{1}_\alpha(m,t)\sum_{n=1}^{N} L_n)$, $\sum_{m=1}^{N} \mathbf{1}_\alpha(m,t) = 1$, and $a_{ij} = T_n(t)/\mathbf{1}_\alpha(m,t)$, GAP is a special case of (42). Thus, (42) is NP-hard. Furthermore, since (42) is a special problem of (11), according to [53], the optimization problem (11) is NP-hard.

Appendix B
Proof of Theorem 2

A constant-moving robot is first assumed, that is, $\beta_n = 0$, and the constraint $T_{\text{goal},n} \leqslant k_n T_{\text{move}}$ should be satisfied. Then, the coverage region decomposition-based optimization problem (12) can be formulated as

$$\max_{\alpha_m(l)} -\frac{1}{L_n}\sum_{l=1}^{L_n} T_n(l) \tag{44}$$
$$\text{s.t.} \quad \text{(42b)} - \text{(42d)}$$

The 0-1 knapsack problem is expressed as follows. In a knapsack, its weight capacity is W. For a set of items, item n has a weight of w_n and a value of v_n. The objective is maximizing the summed value of items that can be packed in the knapsack, while maintaining the summed weight of items less than or equal to the weight capacity W. Thus, the 0-1 knapsack problem is formulated as

$$\max_{\mathbf{O}} \sum_{n\in\mathbf{O}} v_n \tag{45a}$$
$$\text{s.t.} \quad \mathbf{O} \subseteq \mathbf{I} \tag{45b}$$
$$\sum_{n\in\mathbf{O}} w_n \leqslant W \tag{45c}$$

where $\mathbf{O}$ denotes the set of items that should be packed. According to [19], [52], [54], the 0-1 knapsack problem is NP-hard.

If we set $v_n = -T_n/L_n$, $\mathbf{I} = \mathcal{L}_n$, $w_n = \alpha_m(l)$, and $W = 1$, the 0-1 knapsack problem becomes a special case of (44) for $T_n(t) \leqslant T_{n,\max}(t)$. Thus, the problem (44) is also NP-hard. Moreover, since (44) is a special case of (12), according to [53], problem (12) is also NP-hard.

References

[1] P. Park, S. Coleri Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 978–1013, 2nd Quart. 2018.

[2] Next G Alliance, "Next G alliance report: 6G applications and use cases," Alliance for Telecommunications Industry Solutions, Washington, DC, Next G Alliance Report, 2022.

[3] X.-H. You, C.-X. Wang, J. Huang, *et al.*, "Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts," *Sci. China Inf. Sci.*, vol. 64, no. 1, pp. 1–74, Jan. 2021.

[4] T. Ji, E. Ayday, E. Yilmaz, and P. Li, "Robust fingerprinting of genomic databases," *Bioinformatics*, vol. 38, no. Suppl 1, pp. i143–i152, Jun. 2022.

[5] T. Ji, P. Li, E. Yilmaz, E. Ayday, Y. F. Ye, and J. Sun, "Differentially private binary- and matrix-valued data query: An XOR mechanism," *Proc. VLDB Endow.*, vol. 14, no. 5, pp. 849–862, Mar. 2021.

[6] 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.

[7] 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.

[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. 2017.

[9] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live service migration in mobile edge clouds," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 140–147, Feb. 2018.

[10] C. Liu, W. Feng, X. Tao, and N. Ge, "MEC-empowered non-terrestrial network for 6G wide-area time-sensitive Internet of Things," *Engineering*, vol. 8, pp. 96–107, Jan. 2022.

[11] C.-X. Wang, Z. Lv, X. Gao, X. You, Y. Hao, and H. Haas, "Pervasive wireless channel modeling theory and applications to 6G GBSMs for all frequency bands and all scenarios," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9159–9173, Sep. 2022.

[12] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5G wireless communications: A deep learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 227–236, Jan. 2020.

[13] P. Li, C. Zhang, and Y. Fang, "The capacity of wireless Ad Hoc networks using directional antennas," *IEEE Trans. Mobile Comput.*, vol. 10, no. 10, pp. 1374–1387, Oct. 2011.

[14] Y. Guo, L. Yu, Q. Wang, T. Ji, Y. Fang, J. Wei-Kocsis, and P. Li, "Weak signal detection in 5G+ systems: A distributed deep learning framework," in *Proc. Int. Symp. Mobile Ad Hoc Networking Comput. (MobiHoc)*, Shanghai, China, Jul. 2021, pp. 201–210.

[15] G. Wang, W. Xiang, and J. Yuan, "Outage performance for compute-and-forward in generalized multi-way relay channels," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2099–2102, Dec. 2012.

[16] C. Liu, F. Tang, Y. Hu, K. Li, Z. Tang, and K. Li, "Distributed task migration optimization in MEC by extending multi-agent deep reinforcement learning approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1603–1614, Jul. 2021.

[17] 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. 12 240–12 251, Oct. 2020.

[18] 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, Dec. 2019, pp. 1–6.

[19] W. Zhan, C. Luo, G. Min, C. Wang, Q. Zhu, and H. Duan, "Mobility-aware multi-user offloading optimization for mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3341–3356, Mar. 2020.

[20] D. Zhu, T. Li, H. Tian, Y. Yang, Y. Liu, H. Liu, L. Geng, and J. Sun, "Speed-aware and customized task offloading and resource allocation in mobile edge computing," *IEEE Commun. Lett.*, vol. 25, no. 8, pp. 2683–2687, Aug. 2021.

[21] S. Wang, Y. Guo, N. Zhang, P. Yang, A. Zhou, and X. Shen, "Delay-aware microservice coordination in mobile edge computing: A reinforcement learning approach," *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 939–951, Mar. 2021.

[22] 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. 2019.

[23] Z. Tang, H. Zhou, T. Ma, K. Yu, and X. S. Shen, "Leveraging LEO assisted cloud-edge collaboration for energy efficient computation offloading," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Madrid, Spain, Dec. 2021, pp. 1–6.

[24] L. Liu, J. Zhang, X. Zhang, P. Wang, Y. Wang, and L. Ouyang, "Design and analysis of cooperative multicast-unicast transmission scheme in hybrid satellite-terrestrial networks," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*, Chengdu, China, Dec. 2018, pp. 309–314.

[25] Y. Wang, J. Zhang, X. Zhang, P. Wang, and L. Liu, "A computation offloading strategy in satellite terrestrial networks with double edge computing," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*, Chengdu, China, Dec. 2018, pp. 450–455.

[26] G. Cui, X. Li, L. Xu, and W. Wang, "Latency and energy optimization for MEC enhanced SAT-IoT networks," *IEEE Access*, vol. 8, pp. 55 915–55 926, 2020.

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

[28] 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.

[29] A. Varasteh, S. A. Amiri, and C. Mas-Machuca, "HOMA: Online in-flight service provisioning with dynamic bipartite matching," *IEEE Trans. Netw. Service Manag.*, pp. 1–1, 2022.

[30] 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.

[31] M. Li, J. Gao, L. Zhao, and X. Shen, "Adaptive computing scheduling for edge-assisted autonomous driving," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5318–5331, Jun. 2021.

[32] K. Sasaki, N. Suzuki, S. Makido, and A. Nakao, "Vehicle control system coordinated between cloud and mobile edge computing," in *Proc. Annu. Conf. Soc. Instrum. Control Eng. Japan (SICE)*, Tsukuba, Japan, 2016, pp. 1122–1127.

[33] 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, Dec. 2020, pp. 1–7.

[34] X. Huang, L. He, X. Chen, L. Wang, and F. Li, "Revenue and energy efficiency-driven delay-constrained computing task offloading and resource allocation in a vehicular edge computing network: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8852–8868, Jun. 2022.

[35] X. Huang, L. He, and W. Zhang, "Vehicle speed aware computing task offloading and resource allocation based on multi-agent reinforcement learning in a vehicular edge computing network," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Virtual, Beijing, China, Oct. 2020, pp. 1–8.

[36] Y. Wu, K. Ni, C. Zhang, L. P. Qian, and D. H. K. Tsang, "NOMA-assisted multi-access mobile edge computing: A joint optimization of computation offloading and time allocation," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12 244–12 258, Dec. 2018.

[37] Y. Wu, Y. Song, T. Wang, L. Qian, and T. Q. S. Quek, "Non-orthogonal multiple access assisted federated learning via wireless power transfer: A cost-efficient approach," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2853–2869, Apr. 2022.

[38] 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.

[39] 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.

[40] R. Schoenen, G. Bulu, A. Mirtaheri, and H. Yanikomeroglu, "Green communications by demand shaping and user-in-the-loop tariff-based control," in *Proc. IEEE Online Conf. Green Commun. (GreenCom)*, New York, NY, Sep. 2011, pp. 64–69.

[41] R. Schoenen, "On increasing the spectral efficiency more than 100% by user-in-the-control-loop," in *Proc. Asia-Pac. Conf. Commun. (APCC)*, Auckland, New Zealand, Oct. 2010, pp. 159–164.

[42] X. Li, S. Chen, Y. Zhou, J. Chen, and G. Feng, "Intelligent service migration based on hidden state inference for mobile edge computing," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 8, no. 1, pp. 380–393, Mar. 2022.

[43] R. Yang, H. He, and W. Zhang, "Multitier service migration framework based on mobility prediction in mobile edge computing," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–13, Apr. 2021.

[44] Q. Wang, Y. Guo, X. Wang, T. Ji, L. Yu, and P. Li, "AI at the edge: Blockchain-empowered secure multiparty learning with heterogeneous models," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9600–9610, Oct. 2020.

[45] X. Chen, J. Ji, C. Luo, W. Liao, and P. Li, "When machine learning meets blockchain: A decentralized, privacy-preserving and secure design," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Seattle, WA, Dec. 2018, pp. 1178–1187.

[46] F. S. Melo, "Convergence of *Q*-learning: A simple proof," *Institute Of Systems and Robotics, Tech. Rep*, pp. 1–4, 2001.

[47] H. Xiong, L. Zhao, Y. Liang, and W. Zhang, "Finite-time analysis for double *Q*-learning," in *Proc. Adv. Neural Inf. Proces. Syst. (NeurIPS)*, vol. 33, Virtual, Online, Dec. 2020, pp. 16 628–16 638.

[48] D. Lee and N. He, "Periodic *Q*-learning," in *Proc. 2nd Conf. Learn. Dyn. Control*, ser. Proceedings of Machine Learning Research (PMLR), vol. 120, Virtual, Online, Jun. 2020, pp. 582–598.

[49] G. Li, C. Cai, Y. Chen, Y. Gu, Y. Wei, and Y. Chi, "Is *Q*-learning minimax optimal? a tight sample complexity analysis," *arXiv preprint arXiv:2102.06548*, Nov. 2021. [Online]. Available: https://arxiv.53yu.com/abs/2102.06548

[50] F. Maan and N. Mazhar, "MANET routing protocols vs mobility models: A performance evaluation," in *Proc. Int. Conf. Ubiquitous Future Networks (ICUFN)*, Dalian, China, 2011, pp. 179–184.

[51] D. G. Cattrysse and L. N. Van Wassenhove, "A survey of algorithms for the generalized assignment problem," *Eur. J. Oper. Res.*, vol. 60, no. 3, pp. 260–272, Aug. 1992.

[52] K. Bernhard and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 3rd ed. Berlin, Germany: Springer, 2008.

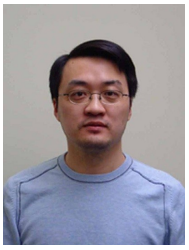
[53] M. R. Garey and D. S. Johnson, "'Strong' NP-completeness results: Motivation, examples, and implications," *J. ACM*, vol. 25, no. 3, pp. 499–508, Jul. 1978.

[54] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, Eds. Boston, MA: Springer, 1972, pp. 85–103.

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 in the department of Electrical and Computer Engineering at the University of Delaware from 2014 to 2016. He has been a lecturer of the School of Electronics and Information Engineering at Tiangong University from 2017 to 2020. He is now a post doctor with the Department of Electronic Engineering, Tsinghua University, from 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 of CHINA COMMUNICATIONS and an Editor of IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING.

Yanmin Wang received the B.S. degree from Shandong University, China, in 2008, and the Ph.D. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2013. She is currently an Associate Professor at the School of Information Engineering, Minzu University of China. Her research interests include distributed antenna systems, satellite networks, and coordinated satellite-UAV-terrestrial networks.



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. Dr. Ge is a senior member of the China Institute of Communications and the Chinese Institute of Electronics.

Cheng-Xiang Wang (Fellow, IEEE) received the B.Sc. and M.Eng. degrees in communication and information systems from Shandong University, Jinan, China, in 1997 and 2000, respectively, and the Ph.D. degree in wireless communications from Aalborg University, Aalborg, Denmark, in 2004.

He was a Research Assistant with the Hamburg University of Technology, Hamburg, Germany, from 2000 to 2001, a Visiting Researcher with Siemens AG Mobile Phones, Munich, Germany, in 2004, and a Research Fellow with the University of Agder, Grimstad, Norway, from 2001 to 2005. He has been with Heriot-Watt University, Edinburgh, U.K., since 2005, where he was promoted to a Professor in 2011. In 2018, he joined Southeast University, Nanjing, China, as a Professor. He is also a part-time Professor with Purple Mountain Laboratories, Nanjing. He has authored 4 books, 3 book chapters, and more than 500 papers in refereed journals and conference proceedings, including 27 highly cited papers. He has also delivered 24 invited keynote speeches/talks and 15 tutorials in international conferences. His current research interests include wireless channel measurements and modeling, 6G wireless communication networks, and electromagnetic information theory.

Dr. Wang is a Member of the Academia Europaea (The Academy of Europe), a Member of the European Academy of Sciences and Arts (EASA), a Fellow of the Royal Society of Edinburgh (FRSE), IEEE, IET, and China Institute of Communications (CIC), an IEEE Communications Society Distinguished Lecturer in 2019 and 2020, a Highly-Cited Researcher recognized by Clarivate Analytics in 2017-2020, and one of the most cited Chinese Researchers recognized by Elsevier in 2021. He is currently an Executive Editorial Committee Member of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He has served as an Editor for over ten international journals, including the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, from 2007 to 2009, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, from 2011 to 2017, and the IEEE TRANSACTIONS ON COMMUNICATIONS, from 2015 to 2017. He was a Guest Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, Special Issue on Vehicular Communications and Networks (Lead Guest Editor), Special Issue on Spectrum and Energy Efficient Design of Wireless Communication Networks, and Special Issue on Airborne Communication Networks. He was also a Guest Editor for the IEEE TRANSACTIONS ON BIG DATA, Special Issue on Wireless Big Data, and is a Guest Editor for the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, Special Issue on Intelligent Resource Management for 5G and Beyond. He has served as a TPC Member, a TPC Chair, and a General Chair for more than 80 international conferences. He received 15 Best Paper Awards from IEEE GLOBECOM 2010, IEEE ICCT 2011, ITST 2012, IEEE VTC-Spring 2013, IWCMC 2015, IWCMC 2016, IEEE/CIC ICCC 2016, WPMC 2016, WOCC 2019, IWCMC 2020, WCSP 2020, CSPS2021, WCSP 2021, and IEEE/CIC ICCC 2022.