This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



# Information Freshness-Aware Task Offloading in Air-Ground Integrated Edge Computing Systems

Xianfu Chen, *Member*, *IEEE*, Celimuge Wu, *Senior Member*, *IEEE*, Tao Chen, *Senior Member*, *IEEE*, Zhi Liu, *Senior Member*, *IEEE*, Honggang Zhang, *Senior Member*, *IEEE*, Mehdi Bennis, *Fellow*, *IEEE*, Hang Liu, *Senior Member*, *IEEE*, and Yusheng Ji, *Senior Member*, *IEEE*

***Abstract*—This paper investigates an air-ground integrated multi-access edge computing system, which is deployed by an infrastructure provider (InP). Under a business agreement with the InP, a third-party service provider provides computing services to the subscribed mobile users (MUs). MUs compete for the shared spectrum and computing resources over time to achieve their distinctive goals. From the perspective of an MU, we deliberately define the age of update to capture the staleness of information from refreshing computation outcomes. Given the system dynamics, we model the interactions among MUs as a stochastic game. In the Nash equilibrium without cooperation, each MU behaves in accordance with the local system states and conjectures. We can hence transform the stochastic game into a single-agent Markov decision process. As another major contribution, we develop an online deep reinforcement learning (RL) scheme that adopts two separate double deep Q-networks to approximate the Q-factor and the post-decision Q-factor, respectively. The deep RL scheme allows each MU to optimize the behaviours with unknown dynamic statistics. Numerical experiments show that our proposed scheme outperforms the baselines in terms of the average utility under various system conditions.**

***Index Terms*—Multi-access edge computing, unmanned aerial vehicle, stochastic games, age of update, multi-agent deep reinforcement learning.**

This paper was presented in part at the IEEE Global Communications Conference (GLOBECOM), Taipei, Taiwan, Dec. 2020.

This work was supported in part by the Academy of Finland under Grants 319759, 319758, and 317669, in part by the Zhejiang Lab Open Program under Grant 2021LC0AB06, in part by The Okawa Foundation for Information and Telecommunications, in part by G-7 Scholarship Foundation, in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grants 19H04092, 20H04174, JP18KK0279 and JP20H00592, in part by the Research Organization of Information and Systems (ROIS) NII Open Collaborative Research 2021(21FA02), in part by the National Natural Science Foundation of China under Grant 61731002, in part by the Zhejiang Key Research and Development Plan under Grant 2019C01002, in part by the Academy of Finland 6G Flagship, in part by the European coordinated CHIST-ERA LearningEdge and CONNECT, and in part by the University of Oulu Infotech NOOR project. *(Corresponding author: Zhi Liu.)*

X. Chen and T. Chen are with the VTT Technical Research Centre of Finland, Finland (e-mail: {xianfu.chen, tao.chen}@vtt.fi).

C. Wu and Z. Liu are with the Graduate School of Informatics and Engineering, University of Electro-Communications, Tokyo, Japan (e-mail: celimuge@uec.ac.jp, liu@ieee.org).

H. Zhang is with the College of Information Science and Electronic Engineering (ISEE), Zhejiang University, Hangzhou, China (e-mail: honggangzhang@zju.edu.cn).

M. Bennis is with the Centre for Wireless Communications, University of Oulu, Finland (e-mail: mehdi.bennis@oulu.fi).

H. Liu is with the Department of Electrical Engineering and Computer Science, the Catholic University of America, USA (e-mail: liuh@cua.edu).

Y. Ji is with the Information Systems Architecture Research Division, National Institute of Informatics, Tokyo, Japan (e-mail: kei@nii.ac.jp).

## I. Introduction

By provisioning data storage and computing power at the network edge, multi-access edge computing (MEC) is instrumental in reducing the computational burden of resource constrained device of a mobile user (MU) [1]. In MEC systems, an MU decides whether the computation tasks are processed locally or offloaded to the edge computing servers for remote execution. Computation offloading not only improves the Quality-of-Experience (QoE) and Quality-of-Service (QoS), but also augments the capability of MUs for running a variety of emerging services and applications (e.g., virtual/augmented reality and mission-critical controls) [2]. However, designing efficient offloading policies remains daunting [3]. The computation offloading performance is predominantly influenced by the wireless connectivity and the remote execution [4]. Specifically, an MU and an edge computing server are connected via the capacity-limited wireless links, where the offloading input data size, the frequency bandwidth and the transmit power have to be carefully optimized. At the edge servers, orchestrating computing resources and coordinating task executions are the key factors.

### A. Related Works and Motivation

Recent years have witnessed a large body of research on computation offloading in MEC, most of which belongs to one-shot optimization (e.g., [5]–[7] and the references therein). The one-shot optimization based designs do not account for the dependence of decision-makings on subsequent system dynamics, and hence cannot reach the optimal long-term computation offloading performance. An infinite time-horizon Markov decision process (MDP) framework has been adopted to investigate the problem of computation offloading, where 1) the Lyapunov optimization technique only constructs an approximately optimal solution [8], [9], and 2) the dynamic programming approach requires full statistical knowledge of system dynamics [10]. Machine learning (ML) has shown the potential to address the performance loss under incomplete statistics. In our prior work [11], we proposed the reinforcement learning (RL)-based schemes to solve the optimal computation offloading policy for a representative MU in an ultra-dense radio access network (RAN). In [12], He et al. studied the privacy vulnerability caused by the wireless communication feature of MEC-enabled Internet-of-Things (IoT), for which an effective computation offloading scheme based on the post-decision state learning was developed. Currently, how to integrate ML into the wireless networks is actively discussed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



by standards development organizations and industrial fora [13]–[17].

The uncertainties in wireless connectivity are the main obstacle for further improving the long-term computation offloading performance [18]. Specifically, a computation offloading policy has to adapt to the spatially and temporally varying channel conditions due to the mobility of MUs [19]. Because of the flexibility of deployment and the desired line-of-sight (LOS) connections, unmanned aerial vehicles (UAVs) are expected to play a significant role in enhancing the performance of the future wireless networks [20]–[23]. Incorporating UAVs to a ground MEC system has been shown to be substantial. In [24], Hu et al. derived an alternating algorithm to minimize the weighted sum energy consumption for an UAV-assisted MEC architecture, where an UAV acts as a computing server or as a relay to help MUs offload the computation tasks to the access point. In [25], Shang and Liu investigated the total energy consumption minimization problem in an air-ground integrated MEC system by jointly optimizing MU association, resource allocation and UAV three-dimensional placement. Despite the efforts focusing on technical implementations, the air-ground integrated MEC systems open up a sustainable business model in the mobile industry [26]. In [27], Asheralieva and Niyato presented a hierarchical game-theoretic and RL framework for computation offloading in a multi-service provider (SP) operated MEC network, where computing servers are installed at both ground base stations (BSs) and UAVs.

This paper is primarily concerned with an air-ground integrated MEC system deployed by an infrastructure provider (InP), where the UAVs serve as the flying computing servers. Multiple computation tasks can be executed in parallel by the created isolated virtual machines (VMs) at an UAV [28]. Such an architecture enables the third-party SPs to provide the ubiquitous computing services to the subscribed MUs with computation requests. The important aspects that need to be researched are as follows.

1) *Lack of a harmonized theoretical framework:* For the air-ground integrated MEC system under consideration, it becomes inevitable to optimize the computation offloading performance from both the technical and economic points of view. However, on the one hand, most of the existing works (e.g., [24] and [25]) are based on a finite time-horizon. To nearly attain the long-term performance, it is nevertheless expensive to repeat the computation offloading problem formulation in accordance with the system dynamics, arising from such as the UAV and MU mobilities, the random computation task arrivals, and the unpredictable available spectrum and computing resources. On the other hand, the economic issues of facilitating an air-ground integrated MEC system are overlooked (e.g., [27]). A long-term business agreement with the InP allows the SPs to steer the computation requests to the edge servers [26]. How to dynamically charge the computing services to the subscribed MUs for revenue maximization remains to be solved by an SP [29].
2) *Delay versus information freshness:* In contrast to the incurred queuing, transmission and computation delay, the QoE and QoS for many applications are restricted by the information freshness of the computation outcomes [30], which adds another dimension of challenge to the computation offloading problem in an air-ground integrated MEC system. Age of information (AoI) has been introduced to describe the timeliness of an update process [31], [32]. By definition, from the perspective of a destination, AoI refers to the time elapsed since the generation of the freshest received update [32], [33]. This paper employs a new metric, termed as age of update (AoU), to capture the fresh information for the air-ground integrated MEC system. More specifically, we define AoU as the amount of time after updating the outcome of the most recently scheduled computation task. Compared with AoI, AoU also takes into account the additional task computation delay. It should be noted that there are only a few related works studying the information freshness under the context of edge computing. In [34], Li et al. used a constrained MDP formulation to minimize the average age of processing for MEC-enabled real-time IoT application. In [35], Xu et al. developed an analytical framework for an IoT system with multiple sensors to analyze the effect of computing on the information freshness, which is in terms of peak AoI. The results of these works are limited to the ground MEC systems and are not applicable to the considered air-ground integrated MEC system.

### B. Contribution

Different from the literature, we concentrate in this paper on the problem of information freshness-aware task offloading in an air-ground integrated MEC system. Across the infinite time-horizon, a third-party SP serves the subscribed MUs, which compete for a limited number of channels and computing resources owned by the InP. Upon receiving the auction bids submitted by the non-cooperative MUs, the resource orchestrator (RO)[1] helps the SP manage the channel allocation through a Vickrey-Clarke-Groves (VCG)[2] pricing mechanism [36]. Consequently, each MU is able to not only locally process a computation task, but also offload a computation task to the ground MEC server or to the UAV for remote execution via the channel won from the auction. In summary, the main contributions from this paper are threefold.

- Taking into account the dynamics and the limited number of channels as well as computing resources in the air-ground integrated MEC system, we formulate the information freshness-aware task offloading across the infinite time-horizon as a stochastic game under the multi-agent MDP framework, in which each MU aims to selfishly maximize its own expected long-term payoff

[1]The role of RO can be mapped as a software-defined controller in network slicing as in [19], and is enhanced by the single ownership of the air-ground integrated MEC system by the InP.

[2]One major advantage of the VCG auction mechanism is that the dominant auction policy of an MU is to bid with the true valuation of the channels, while maintaining the individual rationality and the computational efficiency. In addition, the VCG auction mechanism outperforms the generalized second-price auction for revenue produced to the third-party SP [37].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



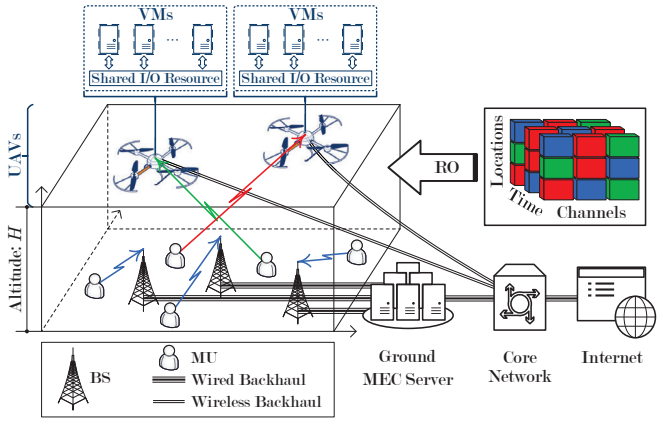


Fig. 1. Illustration of an air-ground integrated MEC system, where the UAVs are deployed as the flying servers. A third-party SP serves the subscribed MUs with sporadic computation requests. The RO is responsible for allocating a limited number of channels to the non-cooperative MUs across the decision epochs based on the submitted auction bids.

from the interactions with other MUs. To the best of our knowledge, there does not exist a comprehensive study for the problem targeted in this paper.

- To avoid any private information exchange among the non-cooperative MUs, we propose that each MU behaves independently with the local conjectures, each of which preserves the payment to the SP from the channel auction and the experienced computation service rate at the UAV. The stochastic game can hence be transformed into a single-agent MDP.
- Without a priori statistical knowledge of system dynamics and to deal with the huge local state space faced by each MU, we put forward a novel online deep RL scheme leveraging the double deep Q-network (DQN) [38]. Similar to a deep advantage actor-critic (A2C) architecture [39], the proposed deep RL scheme maintains for each MU two separate DQNs to approximate, respectively, the Q-factor and the post-decision Q-factor.

The remainder of this paper is organized as follows. In the next section, we describe the air-ground integrated MEC system and the assumptions used throughout this paper. In Section III, we formulate the information freshness-aware task offloading as a stochastic game among the non-cooperative MUs and discuss the general best-response solution. In Section IV, we elaborate how each MU plays the stochastic game with the local conjectures and propose an online deep RL scheme to address the optimal control policy. In Section V, we provide numerical experiments under various settings to compare the performance from our scheme with other baselines. Finally, we draw the conclusions in Section VI. For convenience, Table I summarizes the primary notations of this paper.

## II. System Descriptions and Assumptions

In this paper, we assume an InP deploys an air-ground integrated MEC system as shown in Fig. 1, where the ground MEC server and the UAVs jointly provide computing capability at the network edge. A set $\mathcal{B} = \{1, 2, \cdots, B\}$ of BSs in the RAN are connected via the wired backhaul to the resource-rich ground MEC server, while each UAV works as a parallel computing server. Based on a long-term business agreement with the InP, a third-party SP serves over the system a set $\mathcal{K}$ of subscribed MUs with sporadic computation requests. The UAVs fly in the air at a fixed altitude of $H$ (in meters) [3]. We choose a finite set $\mathcal{L}$ of locations (i.e., small two-dimensional non-overlapping areas) to denote both the service region covered by the RAN and the region of the UAVs mapped vertically from the air to the ground. A location can be characterized by uniform wireless communication conditions [19], [20]. Let $\mathcal{L}_b$ denote the locations covered by an BS $b \in \mathcal{B}$. For any two BSs $b$ and $b' \in \mathcal{B} \setminus \{b\}$, we assume that $\mathcal{L}_b \cap \mathcal{L}_{b'} = \emptyset$. Thus, $\mathcal{L} = \cup_{b\in\mathcal{B}}\mathcal{L}_b$. The geographical topology of the BSs is represented by a two-tuple graph $\langle \mathcal{B}, \mathcal{E} \rangle$, where $\mathcal{E} = \{e_{b,b'} : b, b' \in \mathcal{B}, b \neq b'\}$ with each $e_{b,b'}$ being equal to 1 if BSs $b$ and $b'$ are neighbours, and 0, otherwise. The infinite time-horizon is divided into discrete decision epochs, each of which is with equal duration $\delta$ (in seconds) and indexed by

TABLE I
Primary Notations Used in the Paper.

| Notation | Description |
|---|---|
| $B/\mathcal{B}$ | number/set of BSs |
| $\mathcal{L}_b$ | set of locations covered by BS $b$ |
| $\mathcal{K}$ | set of MUs |
| $\mathcal{C}$ | set of channels |
| $\boldsymbol{\beta}_k$, $\boldsymbol{\beta}_k^\jmath$ | auction bid of MU $k$ |
| $\boldsymbol{\varphi}_k$, $\boldsymbol{\varphi}_k^\jmath$ | channel allocation variable of MU $k$ |
| $\boldsymbol{\phi}$, $\boldsymbol{\phi}^k$ | auction winner determination vector |
| $\tau_k$, $\tau_k^\jmath$ | payment of MU $k$ |
| $X_k$, $X_k^\jmath$ | task offloading decision of MU $k$ |
| $R_k$, $R_k^\jmath$ | packet scheduling decision of MU $k$ |
| $G_{b,k}^\jmath / G_{(\mathrm{v}),k}^\jmath$ | channel power gain between MU $k$ and BS $b$/UAV |
| $W_{(\mathrm{m}),k}$, $W_{(\mathrm{m}),k}^\jmath$ | local CPU state of MU $k$ |
| $W_{(\mathrm{v}),k}$, $W_{(\mathrm{v}),k}^\jmath$ | remote processing state of MU $k$ |
| $D_k$, $D_k^\jmath$ | local transmitter state of MU $k$ |
| $\chi^\jmath$ | computation service rate |
| $F_k$, $F_k^\jmath$ | total local energy consumption of MU $k$ |
| $A_k$, $A_k^\jmath$ | AoU of MU $k$ |
| $\ell_k$ | payoff function of MU $k$ |
| $u_k$ | utility function of MU $k$ |
| $\mathbf{S}$, $\mathbf{S}^\jmath$ | global system state |
| $\mathbf{S}_k$, $\mathbf{S}_k^\jmath$ | local system state of MU $k$ |
| $\mathbf{O}_k$, $\mathbf{O}_k^\jmath$ | local conjecture of MU $k$ |
| $\widetilde{\mathbf{S}}_k$ | local post-decision state of MU $k$ |
| $\boldsymbol{\pi}$, $\boldsymbol{\pi}^*$ | joint control policy |
| $\boldsymbol{\pi}_k$, $\boldsymbol{\pi}_k^*$ | control policy of MU $k$ |
| $V_k$ | expected long-term payoff of MU $k$ |
| $Q_k$ | Q-factor of MU $k$ |
| $\widetilde{Q}_k$ | post-decision Q-factor of MU $k$ |
| $\boldsymbol{\theta}_k$, $\boldsymbol{\theta}_k^\jmath$, $\boldsymbol{\theta}_k^{\jmath,-}$ | parameters associated with the DQN-I of MU $k$ |
| $\tilde{\boldsymbol{\theta}}_k$, $\tilde{\boldsymbol{\theta}}_k^\jmath$ | parameters associated with the DQN-II of MU $k$ |
| $\mathcal{M}_k^\jmath$ | replay memory of MU $k$ |
| $\mathcal{Y}_k^\jmath$ | mini-batch of MU $k$ |

[3]This work assumes that the power of the UAVs is supplied by laser charging [40]. Hence the UAVs are able to operate for the long run. Under the RL framework [41], the proposed study in this paper can be straightforwardly applied to the episodic case in which an episode is defined as the maximum UAV operation time, if an UAV needs to land on the ground for battery recharging [20].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



an integer $j \in \mathbb{N}_+$. To ease the following analysis, we focus on the air-ground integrated MEC system with a single UAV without loss of generality. The results in this paper can be extended to the multi-UAV scenario by simply expanding the dimension of the task offloading decision-makings.

### A. VCG-based Channel Auction

In the service region, we assume that the UAV and the MUs move at the same speed following a Markov mobility model[4]. Let $L^j_{(\mathrm{v})} \in \mathcal{L}$ and $L^j_{(\mathrm{m}),k} \in \mathcal{L}$ denote, respectively, the mapped ground location of the UAV and the location of each MU $k \in \mathcal{K}$ during a decision epoch $j$. The computation task arrivals at the MUs are assumed to be independent and identically distributed sequences of Bernoulli random variables with a common parameter $\lambda \in [0, 1]$. More specifically, we denote by $\zeta^j_k \in \{0, 1\}$ the task arrival indicator for an MU $k$, that is, $\zeta^j_k = 1$ if a computation task is generated at MU $k$ at the beginning of a decision epoch $j$ and otherwise, $\zeta^j_k = 0$. Then, $\mathbb{P}(\zeta^j_k = 1) = 1 - \mathbb{P}(\zeta^j_k = 0) = \lambda$, $\forall k \in \mathcal{K}$, where $\mathbb{P}(\cdot)$ means the probability of the occurrence of an event. Each MU $k$ employs a pre-processing buffer to temporarily store a computation task. Since a newer computation task is always with fresher information (e.g., to share augmented vision between vehicles in accident warning, a task can be an image captured by a vehicle for object detection [45]), it is reasonable for an incoming task with newer arrival time to replace an old task in the pre-processing buffer. We assume that a computation task is composed of $D_{(\max)}$ input data packets and each data packet contains $\mu$ bits. We let $\vartheta$ represent the number of CPU cycles required to accomplish one bit of a computation task. A computation task can be either computed locally at the device of the MU or executed remotely (at the ground MEC server or the UAV). We let $X^j_k \in \mathcal{X} = \{0, 1, 2, 3\}$ denote the computation offloading decision of MU $k$ at each decision epoch $j$, where $X^j_k = 1$, $X^j_k = 2$ and $X^j_k = 3$ indicate that the task in the pre-processing buffer is scheduled to be processed by the local CPU, executed by the ground MEC server and offloaded to the UAV for execution, respectively, while $X^j_k = 0$ means that the task is not scheduled for computation. The RO assists the SP to manage a finite set $\mathcal{C}$ of non-overlapping orthogonal channels, each of which is with the same bandwidth $\eta$ (in Hz). In order to upload the input data packets of a scheduled computation task for remote execution, an MU competes with other MUs in the system for the limited channel access opportunities using an VCG auction mechanism.

Specifically, at the beginning of each decision epoch $j$, each MU $k \in \mathcal{K}$ submits to the RO an auction bid given by a vector $\boldsymbol{\beta}^j_k = (\nu^j_k, \mathbf{N}^j_k)$, where $\nu^j_k$ is the true valuation over $\mathbf{N}^j_k = (N^j_{(\mathrm{s}),k}, N^j_{(\mathrm{v}),k})$ with $N^j_{(\mathrm{s}),k}$ and $N^j_{(\mathrm{v}),k}$ being the numbers of demanded channels for transmitting the input data packets to the ground MEC server and the UAV. We will illustrate how an MU selects the optimal auction bid in Section IV-A. Let $\boldsymbol{\rho}^j_k = (\rho^j_{k,c} : c \in \mathcal{C})$ be the channel allocation vector for MU $k$ during epoch $j$, where $\rho^j_{k,c}$ equals 1 if a channel $c \in \mathcal{C}$ is allocated to MU $k$ during epoch $j$, and 0, otherwise. In order to guarantee no interference in data transmissions during the task offloading, we consider the constraints

$$\left(\sum_{k \in \mathcal{K}^j_{(\mathrm{s}),b}} \rho^j_{k,c}\right) \cdot \left(\sum_{k \in \mathcal{K}^j_{(\mathrm{s}),b'}} \rho^j_{k,c}\right) = 0, \quad \text{if } e_{b,b'} = 1, \forall e_{b,b'} \in \mathcal{E}, \forall c \in \mathcal{C}; \tag{1}$$

$$\left(\sum_{k \in \bigcup_{b \in \mathcal{B}} \mathcal{K}^j_{(\mathrm{s}),b}} \rho^j_{k,c}\right) \cdot \left(\sum_{k \in \mathcal{K}^j_{(\mathrm{v})}} \rho^j_{k,c}\right) = 0, \forall c \in \mathcal{C}; \tag{2}$$

$$\sum_{k \in \mathcal{K}^j_{(\mathrm{s}),b}} \rho^j_{k,c} \leq 1, \forall b \in \mathcal{B}, \forall c \in \mathcal{C}; \tag{3}$$

$$\sum_{k \in \mathcal{K}^j_{(\mathrm{v})}} \rho^j_{k,c} \leq 1, \forall c \in \mathcal{C}; \tag{4}$$

$$\sum_{c \in \mathcal{C}} \rho^j_{k,c} \leq 1, \forall k \in \mathcal{K}, \tag{5}$$

for the centralized channel allocation at the RO at each decision epoch $j$ to ensure that

1) a channel cannot be allocated simultaneously to the MUs covered by two adjacent BSs if the MUs transmit the input data packets to the ground MEC server (Constraint (1));
2) a channel cannot be shared between the data transmissions to the ground MEC server and the UAV (Constraint (2)); and
3) in the coverage of an BS or the ground service region of the UAV, a channel can be assigned to at most one MU, and an MU can be assigned at most one channel (Constraints (3), (4) and (5)),

where $\mathcal{K}^j_{(\mathrm{s}),b} = \{k : k \in \mathcal{K}, L^j_{(\mathrm{m}),k} \in \mathcal{L}_b, N^j_{(\mathrm{s}),k} > 0\}$, $\forall b \in \mathcal{B}$, while $\mathcal{K}^j_{(\mathrm{v})} = \{k : k \in \mathcal{K}, N^j_{(\mathrm{v}),k} > 0\}$. Obviously, we have the following

$$N^j_{(\mathrm{s}),k} + N^j_{(\mathrm{v}),k} \leq 1, \forall k \in \mathcal{K}, \forall j, \tag{6}$$

that constrains the selection of an auction bid.

We denote $\boldsymbol{\phi}^j = (\phi^j_k : k \in \mathcal{K})$ as the winner determination in the channel auction at a decision epoch $j$, where $\phi^j_k = 1$ if an MU $k \in \mathcal{K}$ wins the channel auction while $\phi^j_k = 0$ indicates that no channel is allocated to MU $k$ during the epoch. The RO calculates $\boldsymbol{\phi}^j$ according to

$$\begin{aligned} \boldsymbol{\phi}^j = \arg\max_{\boldsymbol{\phi}} & \sum_{k \in \mathcal{K}} \phi_k \cdot \nu^j_k \\ \text{s.t.} \quad & \text{constraints (1), (2), (3), (4) and (5);} \\ & \sum_{k \in \mathcal{K}^j_{(\mathrm{s}),b}} \varphi^j_k = \phi_k \cdot N^j_{(\mathrm{s}),k}, \forall b \in \mathcal{B}, \forall k \in \mathcal{K}; \\ & \sum_{k \in \mathcal{K}^j_{(\mathrm{v})}} \varphi^j_k = \phi_k \cdot N^j_{(\mathrm{v}),k}, \forall k \in \mathcal{K}, \end{aligned} \tag{7}$$

[4]Other mobility models [42], [43], including changing the flying altitude within the operating region [44], can also be applied with different time granularity and by the transition from two-dimensional mobility to three-dimensional mobility as the UAV moves from the surface into the bulk, but do not affect the proposed scheme in this paper. We leave the UAV trajectory optimization for part of our future investigation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



where $\boldsymbol{\phi} = (\phi_k \in \{0,1\} : k \in \mathcal{K})$ and $\varphi_k^j = \sum_{c\in\mathcal{C}} \rho_{k,c}^j$ is a channel allocation variable that equals 1 if MU $k$ is assigned a channel during the decision epoch, and 0, otherwise. Moreover, the payment for MU $k$ to the SP, which is calculated to be

$$\tau_k^j = \max_{\boldsymbol{\phi}_{-k}} \sum_{l\in\mathcal{K}\setminus\{k\}} \phi_l \cdot \nu_l^j - \sum_{l\in\mathcal{K}\setminus\{k\}} \phi_l^j \cdot \nu_l^j, \tag{8}$$

is incurred from accessing the allocated channel, where $-k$ denotes all the other MUs in $\mathcal{K}$ without the presence of MU $k$. For consistency, we may also rewrite $\varphi_k^j$ and $\tau_k^j$ as, respectively, $\varphi_k(\boldsymbol{\beta}^j)$ and $\tau_k(\boldsymbol{\beta}^j)$, where $\boldsymbol{\beta}^j = (\beta_k^j, \boldsymbol{\beta}_{-k}^j)$. The VCG auction process among the non-cooperative MUs is shown in Fig. 2 in Section IV-C.

### B. Computation and Communication Models

The UAV complements the ground MEC system with the computing resource from the air. By strategically offloading the computation tasks to the ground MEC server or the UAV for remote execution, the MUs can expect a significantly optimized computing experience. Let $T_k^j \in \mathbb{N}$ be the arrival epoch index of the computation task waiting in the pre-processing buffer of an MU $k \in \mathcal{K}$ at the beginning of a decision epoch $j$. By default, we set $T_k^j = 0$ if the pre-processing buffer is empty.

*1) Local Processing:* When a computation task is scheduled for processing locally at the device of an MU $k \in \mathcal{K}$ during a decision epoch $j$, i.e., $X_k^j = 1$, the number of required epochs can be calculated as $\Delta = \lceil (D_{(\max)} \cdot \mu \cdot \vartheta)/(\delta \cdot \varrho) \rceil$, where $\lceil \cdot \rceil$ means the ceiling function and we assume that the local CPU of an MU operates at frequency $\varrho$ (in Hz).

We describe by $W_{(\mathrm{m}),k}^j \in \{0, 1, \cdots, \Delta\}$ the local CPU state of each MU $k \in \mathcal{K}$ at the beginning of each decision epoch $j$, which is the number of remaining epochs required to accomplish the scheduled computation task. In particular, $W_{(\mathrm{m}),k}^j = 0$ indicates that the local CPU is idle and is available for a new task from epoch $j$. The energy (in Joules) consumed by local CPU during epoch $j$ is then given by

$$F_{(\mathrm{m}),k}^j = \tag{9}$$
$$\begin{cases} 0, & \text{for } W_{(\mathrm{m}),k}^j = 0; \\ \varsigma \cdot \left(D_{(\max)} \cdot \mu \cdot \vartheta - (\Delta - 1) \cdot \delta \cdot \varrho\right) \cdot (\varrho)^2, & \text{for } W_{(\mathrm{m}),k}^j = 1; \\ \varsigma \cdot \delta \cdot (\varrho)^3, & \text{for } W_{(\mathrm{m}),k}^j > 1, \end{cases}$$

where $\varsigma$ is the effective switched capacitance that depends on the chip architecture of the device of an MU [46].

*2) Remote Execution:* To upload the input data packets under remote execution, an MU has to be associated to the RAN (via one of the BSs depending on the geographical locations of the MU) or with the UAV until the task is finished. Let $I_k^j \in \mathcal{B} \cup \{B+1\}$ be the association state of each MU $k \in \mathcal{K}$ at the beginning of a decision epoch $j$, namely, $I_k^j = b \in \mathcal{B}$ if MU $k$ is associated with an BS $b$ and if MU $k$ is associated with the UAV, $I_k^j = B+1$. If no computation task is being scheduled during epoch $j$, the association state of MU $k$ is set according to

$$I_k^j = \begin{cases} I_k^{j-1}, \text{ for } I_k^{j-1} = B+1; \\ b, \quad \text{for } I_k^{j-1} \in \mathcal{B} \text{ and } L_{(\mathrm{m}),k}^j \in \mathcal{L}_b. \end{cases} \tag{10}$$

When $I_k^{j+1} \neq I_k^j, \forall j$, a handover is triggered [11]. We assume that the energy consumption during the occurrence of one handover is negligible for MU $k$ but the handover delay is $\xi$ (in seconds). The exact transmission time of MU $k$ during an epoch $j$ can be written as

$$\tilde{\delta}_k^j = \delta - \xi \cdot \mathbf{1}_{\left\{I_k^{j+1} \neq I_k^j\right\}}, \tag{11}$$

where the indicator function $\mathbf{1}_{\{i\}}$ equals 1 if the condition $i$ is met, and 0, otherwise. Let $D_k^j \in \mathcal{D} = \{0, 1, \cdots, D_{(\max)}\}$ denote the local transmitter state of MU $k$ at the beginning of each decision epoch $j$, which is defined as the number of input data packets left at the transmitter. Let $R_k^j$ be the number of input data packets that are scheduled for uploading during epoch $j$, the transmitter state of MU $k$ then evolves to

$$D_k^{j+1} = D_k^j - \varphi_k^j \cdot R_k^j. \tag{12}$$

During a decision epoch $j$, each MU $k$ experiences the average channel power gains $G_{b,k}^j = g_{(\mathrm{s})}(L_{(\mathrm{m}),k}^j)$ for the link to each BS $b$ and $G_{(\mathrm{v}),k}^j = g_{(\mathrm{v})}(L_{(\mathrm{m}),k}^j, L_{(\mathrm{v})}^j)$ for the link to the UAV. Notice that $0 \leq R_k^j \leq \min\{D_k^j, R_{(\max),k}^j\}$, where $R_{(\max),k}^j$ is jointly determined by the channel gain during a decision epoch $j$, the exact transmission time and the maximum transmit power $P_{(\max)}$ at the MUs.

At the beginning of a decision epoch $j$, if an MU $k \in \mathcal{K}$ schedules the computation task in the pre-processing buffer for execution at the ground MEC server, namely, $X_k^j = 2$. During the current and subsequent decision epochs, all the input data packets need to be uploaded via the allocated channels from the VCG auctions over the RAN. When $L_{(\mathrm{m}),k}^j \in \mathcal{L}_b$, $b \in \mathcal{B}$, the energy consumed for reliably transmitting $\varphi_k^j \cdot R_k^j$ input data packets of the scheduled computation task is calculated as

$$F_{(\mathrm{s}),k}^j = \frac{\tilde{\delta}_k^j \cdot \eta \cdot \sigma^2}{G_{b,k}^j} \cdot \left( 2^{\frac{\varphi_k^j \cdot \left(\mu \cdot R_k^j\right)}{\eta \cdot \tilde{\delta}_k^j}} - 1 \right), \tag{13}$$

where $\sigma^2$ is the noise power spectral density. In general, the ground MEC server is of rich computing resources and accordingly, the task execution delay can be ignored. Moreover, the time consumption (by an BS or the UAV) for sending the computation outcome back to the MU is negligible, due to the fact that the computation outcome is much smaller than the input data packets [47].

In this paper, we assume that once all the input data packets of a computation task are received up to a current decision epoch, the UAV starts to execute from the beginning of next epoch by creating VMs for the MUs. If an MU $k \in \mathcal{K}$ decides to upload the computation task to the UAV for execution (i.e.,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



$X_k^j = 3$), the energy consumption of transmitting $\varphi_k^j \cdot R_k^j$ input data packets to the UAV during an epoch $j$ turns to be

$$F_{(\mathrm{v}),k}^j = \frac{\tilde{\delta}_k^j \cdot \eta \cdot \sigma^2}{G_{(\mathrm{v}),k}^j} \cdot \left( 2^{\frac{\varphi_k^j \cdot \left(\mu \cdot R_k^j\right)}{\eta \cdot \tilde{\delta}_k^j}} - 1 \right). \tag{14}$$

Let $\breve{\mathcal{K}}_{(\mathrm{v})}^j$ represent the set of MUs, whose tasks are being simultaneously executed at the UAV during a decision epoch $j$. Sharing the same physical platform of an UAV for parallel execution among the MUs causes I/O interference, leading to computation service rate reduction for each VM [28]. Denote by $\chi_0$ the computation service rate (in bits per second) of an VM created by the UAV given that the task is executed in isolation, the degraded computation service rate of an MU $k \in \breve{\mathcal{K}}_{(\mathrm{v})}^j$ is modeled as $\chi^j = \chi_0 \cdot (1+\varepsilon)^{1-|\breve{\mathcal{K}}_{(\mathrm{v})}^j|}$, where $|\cdot|$ denotes the cardinality of a set and $\varepsilon \in \mathbb{R}_+$ is a factor specifying the percentage of reduction in the computation service rate of an VM when multiplexed with another VM at the UAV. We then update the remote processing state of MU $k$ by

$$W_{(\mathrm{v}),k}^{j+1} = \max\left\{ W_{(\mathrm{v}),k}^j - \chi^j \cdot \delta, 0 \right\}, \tag{15}$$

where $W_{(\mathrm{v}),k}^j$ quantifies the amount of input data bits remaining at the UAV at the beginning of an epoch $j$.

### C. AoU Evolution

For each MU $k \in \mathcal{K}$ in the air-ground integrated MEC system, we define the AoU as the difference between the current time and the arrival time of the computation task whose outcome has most recently been received. The AoU metric clearly depicts the information freshness during the task computing process from the perspective of an MU. Let $A_k^j$ denote the AoU of MU $k$ at each decision epoch $j$. In line with the discussions, an arriving computation task can be either computed at the device of MU $k$, or executed remotely at the ground MEC server or the UAV. Depending on whether or not the computation outcomes are received during an epoch $j$, the AoU evolution of each MU $k$ can be analysed in three cases.

1) When there is no computation outcome received at MU $k$ during decision epoch $j$, the AoU increases linearly according to $A_k^{j+1} = A_k^j + \delta$.
2) If MU $k$ receives only one computation outcome (from either local processing or remote execution) during decision epoch $j$, the AoU is then updated to be

$$A_k^{j+1} = \begin{cases} \left(j - T_{(\mathrm{m}),k}^j - \Delta + 1\right) \cdot \delta + \dfrac{D_{(\max)} \cdot \mu \cdot \vartheta}{\varrho}, \\ \qquad \text{for } W_{(\mathrm{m}),k}^j = 1, D_k^j = 0 \text{ and } W_{(\mathrm{v}),k}^j = 0; \\ \left(j - T_{(\mathrm{s}),k}^j + 1\right) \cdot \delta, \\ \qquad \text{for } W_{(\mathrm{m}),k}^j = 0, D_k^j > 0 \text{ and } W_{(\mathrm{v}),k}^j = 0; \\ \left(j - T_{(\mathrm{v}),k}^j\right) \cdot \delta + \dfrac{W_{(\mathrm{v}),k}^j}{\chi^j}, \\ \qquad \text{for } W_{(\mathrm{m}),k}^j = 0, D_k^j = 0 \text{ and } W_{(\mathrm{v}),k}^j > 0, \end{cases} \tag{16}$$

where $T_{(\mathrm{m}),k}^j$, $T_{(\mathrm{s}),k}^j$ and $T_{(\mathrm{v}),k}^j$ are, respectively, the arrival epoch indices of the tasks computed at the local CPU, the ground MEC server and the UAV.

3) The AoU evolution of MU $k$ can be expressed as

$$A_k^{j+1} = \begin{cases} \left(j - T_{(\mathrm{s}),k}^j + 1\right) \cdot \delta, \\ \qquad \text{for } D_k^j > 0, W_{(\mathrm{v}),k}^j = 0 \text{ and } T_{(\mathrm{s}),k}^j > T_{(\mathrm{m}),k}^j; \\ \left(j - T_{(\mathrm{v}),k}^j\right) \cdot \delta + \dfrac{W_{(\mathrm{v}),k}^j}{\chi^j}, \\ \qquad \text{for } D_k^j = 0, W_{(\mathrm{v}),k}^j > 0 \text{ and } T_{(\mathrm{v}),k}^j > T_{(\mathrm{m}),k}^j; \\ \left(j - T_{(\mathrm{m}),k}^j - \Delta + 1\right) \cdot \delta + \dfrac{D_{(\max)} \cdot \mu \cdot \vartheta}{\varrho}, \\ \qquad \text{otherwise}, \end{cases} \tag{17}$$

when two computation outcomes arrive during decision epoch $j$.

In this paper, the value of AoU is initialized to be $A_k^1 = 0$ and up-limited by $A_{(\max)}$ for each MU $k$. When $A_k^j = A_{(\max)}$, it means that the information from the computation outcomes is too stale for MU $k$.

## III. Game-Theoretic Problem Statement

In this section, we first formulate the problem of information freshness-aware task offloading across the infinite time-horizon from a game-theoretic perspective and then discuss the best-response solution.

### A. Stochastic Game Formulation

During each decision epoch $j$, the local system state of an MU $k \in \mathcal{K}$ can be described by $\mathbf{S}_k^j = (L_{(\mathrm{v})}^j, L_{(\mathrm{m}),k}^j, \mathbf{1}_{\{T_k^j > 0\}}, I_k^j, W_{(\mathrm{m}),k}^j, W_{(\mathrm{v}),k}^j, D_k^j, A_k^j) \in \mathcal{S}$, where $\mathcal{S}$ denotes a common local state space for all MUs in the considered air-ground integrated MEC system. Then $\mathbf{S}^j = (\mathbf{S}_k^j, \mathbf{S}_{-k}^j) \in \mathcal{S}^{|\mathcal{K}|}$ characterizes the global system state during decision epoch $j$. Let $\boldsymbol{\pi}_k = (\pi_{(\mathrm{c}),k}, \pi_{(\mathrm{t}),k}, \pi_{(\mathrm{p}),k})$ denote the stationary control policy of MU $k$, where $\pi_{(\mathrm{c}),k}$, $\pi_{(\mathrm{t}),k}$ and $\pi_{(\mathrm{p}),k}$ are the channel auction, the task offloading and the packet scheduling policies, respectively. It is worth noting that $\pi_{(\mathrm{p}),k}$ is MU-specified and dependent on $\mathbf{S}_k^j$ only. The joint control policy of all MUs can be given by $\boldsymbol{\pi} = (\boldsymbol{\pi}_k, \boldsymbol{\pi}_{-k})$. When deploying $\boldsymbol{\pi}_k$, MU $k$ observes $\mathbf{S}^j$ at the beginning of each decision epoch $j$ and accordingly, submits the channel auction bid as well as makes the decisions of computation task offloading and input data packet scheduling, that is, $\boldsymbol{\pi}_k(\mathbf{S}^j) = (\pi_{(\mathrm{c}),k}(\mathbf{S}^j), \pi_{(\mathrm{t}),k}(\mathbf{S}^j), \pi_{(\mathrm{p}),k}(\mathbf{S}_k^j)) = (\boldsymbol{\beta}_k^j, X_k^j, R_k^j)$. We define an immediate payoff function[5] for MU $k$ by

$$\ell_k\left(\mathbf{S}^j, \left(\varphi_k^j, X_k^j, R_k^j\right)\right) = u_k\left(\mathbf{S}^j, \left(\varphi_k^j, X_k^j, R_k^j\right)\right) - \tau_k^j, \tag{18}$$

[5]To stabilize the training of the proposed scheme in this paper, we choose an exponential function for the definition of a payoff utility, whose value does not dramatically diverge. Moreover, the exponential function has been well fitted to the generic quantitative relationship between the QoE and the QoS [48].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



in which the utility function $u_k(\mathbf{S}^j, (\varphi_k^j, X_k^j, R_k^j)) = \varpi_k \cdot \exp(-A_k^j) + \omega_k \cdot \exp(-F_k^j)$ measures the satisfaction of information freshness and total local energy consumption $F_k^j = F_{(\mathrm{m}),k}^j + F_{(\mathrm{s}),k}^j + F_{(\mathrm{v}),k}^j$ during each epoch $j$, while $\varpi_k \in \mathbb{R}_+$ and $\omega_k \in \mathbb{R}_+$ are the weighting constants.

It is easy to verify that the randomness hidden in a sequence of the global system state realizations over the infinite time-horizon $\{\mathbf{S}^j : j \in \mathbb{N}_+\}$ is Markovian with the controlled state transition probability given by (19) at the bottom of Page 8, where $\boldsymbol{\varphi}(\boldsymbol{\pi}_{(\mathrm{c})}(\mathbf{S}^j)) = (\varphi_k(\boldsymbol{\pi}_{(\mathrm{c})}(\mathbf{S}^j)), \boldsymbol{\varphi}_{-k}(\boldsymbol{\pi}_{(\mathrm{c})}(\mathbf{S}^j)))$ is the global channel allocation by the RO, while $\boldsymbol{\pi}_{(\mathrm{c})} = (\pi_{(\mathrm{c}),k}, \boldsymbol{\pi}_{(\mathrm{c}),-k})$, $\boldsymbol{\pi}_{(\mathrm{t})} = (\pi_{(\mathrm{t}),k}, \boldsymbol{\pi}_{(\mathrm{t}),-k})$ and $\boldsymbol{\pi}_{(\mathrm{p})} = (\pi_{(\mathrm{p}),k}, \boldsymbol{\pi}_{(\mathrm{p}),-k})$ are the joint channel auction, the joint task offloading and the joint packet scheduling policies, respectively. Given the control policy $\boldsymbol{\pi}_k$ by each MU $k \in \mathcal{K}$ and an initial global system state $\mathbf{S} = (\mathbf{S}_k = (L_{(\mathrm{v})}, L_{(\mathrm{m}),k}, \mathbf{1}_{\{T_k>0\}}, I_k, W_{(\mathrm{m}),k}, W_{(\mathrm{v}),k}, D_k, A_k) : k \in \mathcal{K}) \in \mathcal{S}^{|\mathcal{K}|}$, we express the expected long-term discounted payoff function of MU $k$ as below

$$V_k(\mathbf{S}, \boldsymbol{\pi}) = \tag{20}$$
$$(1-\gamma) \cdot \mathsf{E}_{\boldsymbol{\pi}}\left[\sum_{j=1}^{\infty} (\gamma)^{j-1} \cdot \ell_k\Big(\mathbf{S}^j, \Big(\varphi_k^j, X_k^j, R_k^j\Big)\Big) \,|\mathbf{S}^1 = \mathbf{S}\right],$$

where $\gamma \in [0, 1)$ is the discount factor and the expectation $\mathsf{E}_{\boldsymbol{\pi}}[\cdot]$ is taken over different decision-makings under different global system states following the joint control policy $\boldsymbol{\pi}$ across the discrete decision epochs. When $\gamma$ approaches 1, (20) well approximates the expected long-term un-discounted payoff[6] [49]. $V_k(\mathbf{S}, \boldsymbol{\pi})$ in (20) is also named as the state-value function of the global system state $\mathbf{S}$ under the joint control policy $\boldsymbol{\pi}$ [41].

Due to the limited number of channels managed by the RO, the shared I/O resource at the physical platform of the UAV and the dynamic characteristics of the air-ground integrated MEC system, we formulate the problem of information freshness-aware task offloading among the competing MUs over the infinite time-horizon as a non-cooperative stochastic game, in which $|\mathcal{K}|$ MUs are the players and there are a set $\mathcal{S}^{|\mathcal{K}|}$ of global system states and a collection of control policies $\{\boldsymbol{\pi}_k : \forall k \in \mathcal{K}\}$. The objective of each MU $k$ in the stochastic game is to device a best-response control policy $\boldsymbol{\pi}_k^* = (\pi_{(\mathrm{c}),k}^*, \pi_{(\mathrm{t}),k}^*, \pi_{(\mathrm{p}),k}^*)$ that maximizes its own $V_k(\mathbf{S}, \boldsymbol{\pi})$ for any given global system state $\mathbf{S} \in \mathcal{S}^{|\mathcal{K}|}$, which can be formulated as

$$\boldsymbol{\pi}_k^* = \arg\max_{\boldsymbol{\pi}_k} V_k(\mathbf{S}, \boldsymbol{\pi}), \forall \mathbf{S} \in \mathcal{S}^{|\mathcal{K}|}. \tag{21}$$

A Nash equilibrium (NE) describes the rational behaviours of the MUs in a stochastic game. Specifically, an NE is a tuple of control policies $\langle \boldsymbol{\pi}_k^* : k \in \mathcal{K} \rangle$, where each $\boldsymbol{\pi}_k^*$ of an MU $k$ is the best response to $\boldsymbol{\pi}_{-k}^*$. Theorem 1 ensures the existence of an NE in our formulated game.

[6]The non-cooperative interactions among MUs in the system result in that the control policies, $\boldsymbol{\pi}_k$, $\forall k \in \mathcal{K}$, are not unichain. Therefore, the Markovian system is non-ergodic, due to which we continue using (20) as the optimization objective for each MU.

*Theorem 1.* For the $|\mathcal{K}|$-player stochastic game with expected long-term discounted payoffs, there always exists an NE in stationary control policies [50].

For brevity, we define $V_k(\mathbf{S}) = V_k(\mathbf{S}, \boldsymbol{\pi}_k^*, \boldsymbol{\pi}_{-k}^*)$ as the optimal state-value function, $\forall k \in \mathcal{K}$, $\forall \mathbf{S} \in \mathcal{S}^{|\mathcal{K}|}$. From (20), we can easily observe that the expected long-term payoff of an MU $k \in \mathcal{K}$ depends on information of not only the global system states across the time-horizon but also the joint control policy $\boldsymbol{\pi}$. In other words, the decision-makings from all MUs are coupled in the stochastic game.

### B. Best-Response Approach

Suppose that in the formulated stochastic game, the global system state information over the infinite time-horizon is perfectly known to all MUs and all MUs behave following the NE control policy profile $\boldsymbol{\pi}^* = (\boldsymbol{\pi}_k^*, \boldsymbol{\pi}_{-k}^*)$, the best-response of each MU $k \in \mathcal{K}$ under a current global system state $\mathbf{S} \in \mathcal{S}^{|\mathcal{K}|}$ can then be given in the form of (22) (shown at the bottom of Page 8), where $\mathbf{S}' = (\mathbf{S}_k' = (L_{(\mathrm{v})}', L_{(\mathrm{m}),k}', \mathbf{1}_{\{T_k'>0\}}, I_k', W_{(\mathrm{m}),k}', W_{(\mathrm{v}),k}', D_k', A_k') : k \in \mathcal{K})$ is the consequent global system state. We note that in order to operate in the NE, all MUs must have a priori the statistical knowledge of global dynamics (i.e., (19)), which is prohibited for a non-cooperative system.

## IV. Deep RL with Local Conjectures

In this section, we shall elaborate on how the MUs play the non-cooperative stochastic game only with limited local information. Our aim is to develop an online deep RL scheme to approach the NE control policy with the local conjectures from the interactions among the competing MUs.

### A. Local Conjectures

During the competitive interactions in the stochastic game, it is challenging for each MU $k \in \mathcal{K}$ to obtain the private system state information at other MUs. Meanwhile, the coupling of the decision-makings by the non-cooperative MUs exists in the channel auction and the remote task execution at the UAV. From the viewpoint of an MU $k$, the payment $\tau_k^j$ to the SP in the channel auction and the computation service rate[7] $\chi^j$ at each decision epoch $j$ are realized under $\mathbf{S}_{-k}^j$. In our previous works [19], [51], an abstract game was constructed to approximate the stochastic game with a bounded performance regret under stationary control policies. However, the approximation bound highly depends on the abstraction mechanisms [52]. Instead, in this paper, we allow each MU $k$ to conjecture $\mathbf{S}^{j+1}$ during the next decision epoch $j+1$ as $\widehat{\mathbf{S}}_k^{j+1} = (\mathbf{S}_k^{j+1}, \mathbf{O}_k^{j+1})$, where $\mathbf{O}_k^{j+1} = (\tau_k^j, \chi^j) \in \mathcal{O}_k$ with $\mathcal{O}_k$ being the finite space[8]

[7]It is straightforward that during each epoch $j$, the computation service rate $\chi^j$ of an MU $k \in \breve{\mathcal{K}}_{(\mathrm{v})}^j$ can be estimated locally with $W_{(\mathrm{v}),k}^j$, $W_{(\mathrm{v}),k}^{j+1}$ and the time consumed by the respective VM at the UAV.

[8]From the assumptions made throughout the paper, the payments and the computation service rates take discrete values. Therefore, the space $\mathcal{O}_k$ is sufficiently large, but finite.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



of all possible local conjectures. Now we are able to transform (20) into

$$V_k\Big(\widehat{\mathbf{S}}_k, \boldsymbol{\pi}\Big) = \tag{23}$$
$$(1-\gamma)\cdot \mathsf{E}_{\boldsymbol{\pi}}\Bigg[\sum_{j=1}^{\infty}(\gamma)^{j-1}\cdot \ell_k\Big(\mathbf{S}^j, \Big(\varphi_k^j, X_k^j, R_k^j\Big)\Big)\,|\widehat{\mathbf{S}}_k^1 = \widehat{\mathbf{S}}_k\Bigg],$$

where $\widehat{\mathbf{S}}_k = (\mathbf{S}_k, \mathbf{O}_k) \in \widehat{\mathcal{S}}_k = \mathcal{S}\times\mathcal{O}_k$ with $\mathbf{O}_k$ being the initial local conjecture of $\mathbf{S}_{-k}$[9], while $\boldsymbol{\pi}$ hereinafter refers to the conjecture based joint control policy. Each MU $k$ then switches to maximize $V_k(\widehat{\mathbf{S}}_k, \boldsymbol{\pi}), \forall \widehat{\mathbf{S}}_k \in \widehat{\mathcal{S}}_k$, which is basically a single-agent MDP. With a slight abuse of notation, we let $V_k(\widehat{\mathbf{S}}_k) = V_k(\widehat{\mathbf{S}}_k, \boldsymbol{\pi}^*), \forall k \in \mathcal{K}$, where $\boldsymbol{\pi}^*$ is the best-response control policy profile of all MUs with local conjectures and the Bellman's optimality equation is given by (24), which is shown at the bottom of Page 9.

As addressed in Section II-A, each MU $k \in \mathcal{K}$ in the system observes local state $\widehat{\mathbf{S}}_k \in \widehat{\mathcal{S}}_k$ at the beginning of a current decision epoch and submits an optimal auction bid $\pi^*_{(\mathrm{c}),k}(\widehat{\mathbf{S}}_k) = (\nu_k, \mathbf{N}_k)$ to the RO, which includes a true valuation $\nu_k$ of occupying $\mathbf{N}_k = (N_{(\mathrm{s}),k}, N_{(\mathrm{v}),k})$ channels. We have Theorem 2 that provides the optimal configuration of $(\nu_k, \mathbf{N}_k)$.

*Theorem 2:* When all MUs in the system follow the best-response control policy profile $\boldsymbol{\pi}^*$ based on the local conjectures, each MU $k \in \mathcal{K}$ declares at the beginning of a current decision epoch to the RO the channel demands as

$$N_{(\mathrm{s}),k} = N_{(\mathrm{v}),k} = 0, \text{ if } z_k = 0, \tag{25}$$

or

$$\text{if } z_k = 1, \text{ then } \begin{cases} N_{(\mathrm{s}),k} = 1, \text{ for } \pi^*_{(\mathrm{t}),k}\Big(\widehat{\mathbf{S}}_k\Big) = 2; \\ N_{(\mathrm{v}),k} = 1, \text{ for } \pi^*_{(\mathrm{t}),k}\Big(\widehat{\mathbf{S}}_k\Big) = 3, \end{cases} \tag{26}$$

[9]The conjecture $\mathbf{O}_k = \mathbf{O}_k^1$ of each MU $k \in \mathcal{K}$ at the beginning of decision epoch $j = 1$ can be initialized to be, for example, $(0, 0)$ as in numerical simulations.

together with the true valuation being specified as

$$\nu_k = u_k\Big(\mathbf{S}, \Big(z_k, \pi^*_{(\mathrm{t}),k}\Big(\widehat{\mathbf{S}}_k\Big), \pi^*_{(\mathrm{p}),k}(\mathbf{S}_k)\Big)\Big) + \frac{\gamma}{1-\gamma}\cdot \tag{27}$$
$$\sum_{\widehat{\mathbf{S}}'_k\in\widehat{\mathcal{S}}_k} \mathbb{P}\Big(\widehat{\mathbf{S}}'_k|\widehat{\mathbf{S}}_k, \Big(z_k, \pi^*_{(\mathrm{t}),k}\Big(\widehat{\mathbf{S}}_k\Big), \pi^*_{(\mathrm{p}),k}(\mathbf{S}_k)\Big)\Big)\cdot V_k\Big(\widehat{\mathbf{S}}'_k\Big),$$

where $z_k \in \{0, 1\}$ is the preference of winning one channel from the VCG auction centralized at the RO and satisfies

$$z_k = \tag{28}$$
$$\arg\max_{z\in\{0,1\}}\Bigg\{(1-\gamma)\cdot \ell_k\Big(\mathbf{S}, \Big(z, \pi^*_{(\mathrm{t}),k}\Big(\widehat{\mathbf{S}}_k\Big), \pi^*_{(\mathrm{p}),k}(\mathbf{S}_k)\Big)\Big) + \gamma\cdot$$
$$\sum_{\widehat{\mathbf{S}}'_k\in\widehat{\mathcal{S}}_k} \mathbb{P}\Big(\widehat{\mathbf{S}}'_k|\widehat{\mathbf{S}}_k, \Big(z, \pi^*_{(\mathrm{t}),k}\Big(\widehat{\mathbf{S}}_k\Big), \pi^*_{(\mathrm{p}),k}(\mathbf{S}_k)\Big)\Big)\cdot V_k\Big(\widehat{\mathbf{S}}'_k\Big)\Bigg\}.$$

*Proof:* The conjecture based best-response control policy $\boldsymbol{\pi}^*_k$ of each MU $k \in \mathcal{K}$ in the air-ground integrated MEC system consists of the channel auction policy $\pi^*_{(\mathrm{c}),k}$, the task offloading policy $\pi^*_{(\mathrm{t}),k}$ and the packet scheduling policy $\pi^*_{(\mathrm{p}),k}$. We hence restructure (24) as (29) (shown at the bottom of Page 9), $\forall \widehat{\mathbf{S}}_k \in \widehat{\mathcal{S}}_k$, where $\boldsymbol{\beta}_k = \pi_{(\mathrm{c}),k}(\widehat{\mathbf{S}}_k)$. From the rules of winner determination in (7) as well as payment calculation in (8), the optimal channel auction policy for MU $k$ is to bid truthfully across the decision epochs according to (25), (26) and (27). □

Without knowing the statistical dynamic characteristics of the local states and the structure of the payment function in VCG auction, it yet remains challenging for an MU to come up with an optimal bid configured by (25), (26) and (27) at the beginning of each decision epoch.

### *B. Post-Decision Q-Factor*

In order to remove the obstacle for the calculations of an optimal auction bid at the beginning of each decision epoch, we introduce a local post-decision state (as in [51]) for the MUs in the considered air-ground integrated MEC system. At each current decision epoch, the local post-decision state of an MU $k \in \mathcal{K}$ is defined as $\widetilde{\mathbf{S}}_k = (L_{(\mathrm{v})}, L_{(\mathrm{m}),k}, \mathbf{1}_{\{T_k>0\}}, I_k, W_{(\mathrm{m}),k}, W_{(\mathrm{v}),k}, \widetilde{D}_k, A_k, \mathbf{O}_k) \in \widehat{\mathcal{S}}_k$ by letting $\widetilde{D}_k = D_k - \varphi_k(\boldsymbol{\beta})\cdot R_k$, where $\boldsymbol{\beta} = (\boldsymbol{\beta}_k, \boldsymbol{\beta}_{-k})$.

$$\mathbb{P}\big(\mathbf{S}^{j+1}|\mathbf{S}^j, \big(\boldsymbol{\varphi}\big(\boldsymbol{\pi}_{(\mathrm{c})}\big(\mathbf{S}^j\big)\big), \boldsymbol{\pi}_{(\mathrm{t})}\big(\mathbf{S}^j\big), \boldsymbol{\pi}_{(\mathrm{p})}\big(\mathbf{S}^j\big)\big)\big) =$$
$$\mathbb{P}\Big(L^{j+1}_{(\mathrm{v})}|L^j_{(\mathrm{v})}\Big)\cdot\prod_{k\in\mathcal{K}}\mathbb{P}\Big(L^{j+1}_{(\mathrm{m}),k}|L^j_{(\mathrm{m}),k}\Big)\cdot\mathbb{P}\Big(\Big(\mathbf{1}_{\{T_k^{j+1}>0\}}, I_k^{j+1}, W^{j+1}_{(\mathrm{m}),k}\Big)\,|\,\Big(\mathbf{1}_{\{T_k^j>0\}}, I_k^j, W^j_{(\mathrm{m}),k}\Big), \pi_{(\mathrm{t}),k}\big(\mathbf{S}^j\big)\Big)\cdot$$
$$\mathbb{P}\Big(\Big(W^{j+1}_{(\mathrm{v}),k}, D_k^{j+1}, A_k^{j+1}\Big)\,|\,\Big(W^j_{(\mathrm{v}),k}, D_k^j, A_k^j\Big), \Big(\varphi_k\big(\boldsymbol{\pi}_{(\mathrm{c})}\big(\mathbf{S}^j\big)\big), \pi_{(\mathrm{t}),k}\big(\mathbf{S}^j\big), \pi_{(\mathrm{p}),k}\Big(\mathbf{S}_k^j\Big)\Big)\Big) \tag{19}$$

$$V_k(\mathbf{S}) = \max_{\boldsymbol{\pi}_k(\mathbf{S})}\Bigg\{(1-\gamma)\cdot\ell_k\Big(\mathbf{S}, \Big(\varphi_k\Big(\pi_{(\mathrm{c}),k}(\mathbf{S}), \boldsymbol{\pi}^*_{(\mathrm{c}),-k}(\mathbf{S})\Big), \pi_{(\mathrm{t}),k}(\mathbf{S}), \pi_{(\mathrm{p}),k}(\mathbf{S}_k)\Big)\Big) +$$
$$\gamma\cdot\sum_{\mathbf{S}'\in\mathcal{S}^{|\mathcal{K}|}}\mathbb{P}\Big(\mathbf{S}'|\mathbf{S}, \Big(\boldsymbol{\varphi}\Big(\pi_{(\mathrm{c}),k}(\mathbf{S}), \boldsymbol{\pi}^*_{(\mathrm{c}),-k}(\mathbf{S})\Big), \Big(\pi_{(\mathrm{t}),k}(\mathbf{S}), \boldsymbol{\pi}^*_{(\mathrm{t}),-k}(\mathbf{S})\Big), \Big(\pi_{(\mathrm{p}),k}(\mathbf{S}_k), \boldsymbol{\pi}^*_{(\mathrm{p}),-k}(\mathbf{S}_{-k})\Big)\Big)\Big)\cdot V_k(\mathbf{S}')\Bigg\} \tag{22}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



The local post-decision state herein can be interpreted as a local intermediate state right after the input data packet transmissions but before the transition into the next local state. The probability of the transition from $\widehat{\mathbf{S}}_k$ to $\widehat{\mathbf{S}}'_k$ under a conjecture based joint control policy $\boldsymbol{\pi}$ can be expressed as

$$\mathbb{P}\Big(\widehat{\mathbf{S}}'_k|\widehat{\mathbf{S}}_k,(\varphi_k(\boldsymbol{\beta}),X_k,R_k)\Big)= \tag{30}$$
$$\mathbb{P}\Big(\widetilde{\mathbf{S}}_k|\widehat{\mathbf{S}}_k,(\varphi_k(\boldsymbol{\beta}),X_k,R_k)\Big)\cdot\mathbb{P}\Big(\widehat{\mathbf{S}}'_k|\widetilde{\mathbf{S}}_k,(\varphi_k(\boldsymbol{\beta}),X_k,R_k)\Big),$$

where it admits $\mathbb{P}(\widetilde{\mathbf{S}}_k|\widehat{\mathbf{S}}_k,(\varphi_k(\boldsymbol{\beta}),X_k,R_k))=1$.

For each MU $k\in\mathcal{K}$ in the system, we define the right-hand-side of (24) as a Q-factor, which is a mapping[10] $Q_k:\widehat{\mathcal{S}}_k\times\{0,1\}\times\mathcal{X}\times\mathcal{D}\to\mathbb{R}$, namely,

$$Q_k\Big(\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k)\Big)=(1-\gamma)\cdot\ell_k(\mathbf{S},(\varphi_k,X_k,R_k))+$$
$$\gamma\cdot\sum_{\widehat{\mathbf{S}}'_k\in\widehat{\mathcal{S}}_k}\mathbb{P}\Big(\widehat{\mathbf{S}}'_k|\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k)\Big)\cdot V_k\Big(\widehat{\mathbf{S}}'_k\Big), \tag{31}$$

where $\varphi_k$, $X_k$ and $R_k$ correspond to, respectively, the channel allocation, the computation task offloading and the input data packet scheduling decisions under the current local state $\widehat{\mathbf{S}}_k$. For notational simplicity, the channel allocation function $\varphi_k(\boldsymbol{\beta}_k,\boldsymbol{\pi}^*_{(\mathrm{c}),-k}(\widehat{\mathbf{S}}_{-k}))$ of the auction bidding variable $\boldsymbol{\beta}_k$ is equivalently substituted by $\varphi_k$. By strictly following (30) and (31), we further define a post-decision Q-factor by

$$\widetilde{Q}_k\Big(\widetilde{\mathbf{S}}_k,(\varphi_k,X_k,R_k)\Big)=$$
$$\gamma\cdot\sum_{\widehat{\mathbf{S}}'_k\in\widehat{\mathcal{S}}_k}\mathbb{P}\Big(\widehat{\mathbf{S}}'_k|\widetilde{\mathbf{S}}_k,(\varphi_k,X_k,R_k)\Big)\cdot V_k\Big(\widehat{\mathbf{S}}'_k\Big), \tag{32}$$

which indicates another mapping for MU $k$, that is, $\widetilde{Q}_k:\widehat{\mathcal{S}}_k\times\{0,1\}\times\mathcal{X}\times\mathcal{D}\to\mathbb{R}$.

By substituting (32) back into (27), we arrive at the true valuation of each MU $k\in\mathcal{K}$,

$$\nu_k=u_k\Big(\mathbf{S}^j,\Big(z_k,\pi^*_{(\mathrm{t}),k}\Big(\widehat{\mathbf{S}}_k\Big),\pi^*_{(\mathrm{p}),k}(\mathbf{S}_k)\Big)\Big)$$
$$+\frac{1}{1-\gamma}\cdot\widetilde{Q}_k\Big(\widetilde{\mathbf{S}}_k,\Big(z_k,\pi^*_{(\mathrm{t}),k}\Big(\widehat{\mathbf{S}}_k\Big),\pi^*_{(\mathrm{p}),k}(\mathbf{S}_k)\Big)\Big), \tag{33}$$

[10]To keep what follows uniform, we do not exclude the infeasible decision-makings under a local state for an MU.

where the preference $z_k$ can be then derived from $z_k=\arg\max_{z\in\{0,1\}}Q_k(\widehat{\mathbf{S}}_k,(z,\pi^*_{(\mathrm{t}),k}(\widehat{\mathbf{S}}_k),\pi^*_{(\mathrm{p}),k}(\mathbf{S}_k)))$ instead of originally from (28). In the following subsection, we propose a novel deep RL scheme to learn the Q-factor and the post-decision Q-factor for each MU $k$.

### C. Proposed Deep RL Scheme

With the previously defined Q-factor as in (31), the optimal state-value function for each MU $k\in\mathcal{K}$ in the system can be in turn obtained from

$$V_k\Big(\widehat{\mathbf{S}}_k\Big)=\max_{\varphi_k,X_k,R_k}Q_k\Big(\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k)\Big), \tag{34}$$

$\forall\widehat{\mathbf{S}}_k\in\widehat{\mathcal{S}}_k$. The conventional model-free Q-learning algorithm can be applied to learn both the Q-factor and the post-decision Q-factor [12]. During the learning process, MU $k$ first acquires $\widehat{\mathbf{S}}_k=\widehat{\mathbf{S}}^j_k$, $(\varphi_k,X_k,R_k)=(\varphi^j_k,X^j_k,R^j_k)$, $\ell_k(\mathbf{S},(\varphi_k,X_k,R_k))$ during a current decision epoch $j$ as well as $\widehat{\mathbf{S}}'_k=\widehat{\mathbf{S}}^{j+1}_k$ at the beginning of next decision epoch $j+1$, and then proceeds to update the Q-factor and the post-decision Q-factor in an iterative manner using, respectively, (35) and (36) at the bottom of Page 10, where $\alpha^j\in[0,1)$ denotes the learning rate, while $\varphi'_k$, $X'_k$ and $R'_k$ are the feasible channel allocation, the computation task offloading and the input data packet scheduling decisions under $\widehat{\mathbf{S}}'_k$, respectively. It has been well established that if: 1) the global system state transition probability under $\boldsymbol{\pi}^*$ is time-invariant; 2) $\sum_{j=1}^{\infty}\alpha^j$ is infinite and $\sum_{j=1}^{\infty}(\alpha^j)^2$ is finite; and 3) $\widehat{\mathcal{S}}_k\times\{0,1\}\times\mathcal{X}\times\mathcal{D}$ is exhaustively explored, the learning process surely converges [12].

It is not difficult to find that for the air-ground integrated MEC system investigated in this paper, the space $\widehat{\mathcal{S}}_k$ of local states faced by each MU $k\in\mathcal{K}$ is extremely large. The tabular nature in representing the Q-factor and the post-decision Q-factor values makes the learning rule as in (35) and (36) impractical. Inspired by the recent advances in neural networks [53] and the widespread success of a deep neural network, we propose to adopt two separate deep Q-networks (DQNs), namely, DQN-I and DQN-II, to reproduce the Q-factor and

$$V_k\Big(\widehat{\mathbf{S}}_k\Big)=\max_{\boldsymbol{\pi}_k(\widehat{\mathbf{S}}_k)}\Bigg\{(1-\gamma)\cdot\ell_k\Big(\mathbf{S},\varphi_k\Big(\pi_{(\mathrm{c}),k}\Big(\widehat{\mathbf{S}}_k\Big),\boldsymbol{\pi}^*_{(\mathrm{c}),-k}\Big(\widehat{\mathbf{S}}_{-k}\Big)\Big),\pi_{(\mathrm{t}),k}\Big(\widehat{\mathbf{S}}_k\Big),\pi_{(\mathrm{p}),k}(\mathbf{S}_k)\Big)+$$
$$\gamma\cdot\sum_{\widehat{\mathbf{S}}'_k\in\widehat{\mathcal{S}}_k}\mathbb{P}\Big(\widehat{\mathbf{S}}'_k|\widehat{\mathbf{S}}_k,\Big(\varphi_k\Big(\pi_{(\mathrm{c}),k}\Big(\widehat{\mathbf{S}}_k\Big),\boldsymbol{\pi}^*_{(\mathrm{c}),-k}\Big(\widehat{\mathbf{S}}_{-k}\Big)\Big),\pi_{(\mathrm{t}),k}\Big(\widehat{\mathbf{S}}_k\Big),\pi_{(\mathrm{p}),k}(\mathbf{S}_k)\Big)\Big)\cdot V_k\Big(\widehat{\mathbf{S}}'_k\Big)\Bigg\} \tag{24}$$

$$\pi^*_{(\mathrm{c}),k}\Big(\widehat{\mathbf{S}}_k\Big)=\arg\max_{\boldsymbol{\beta}_k}\Bigg\{u_k\Big(\mathbf{S},\Big(\varphi_k\Big(\boldsymbol{\beta}_k,\boldsymbol{\pi}^*_{(\mathrm{c}),-k}\Big(\widehat{\mathbf{S}}_{-k}\Big)\Big),\pi^*_{(\mathrm{t}),k}\Big(\widehat{\mathbf{S}}_k\Big),\pi^*_{(\mathrm{p}),k}(\mathbf{S}_k)\Big)\Big)-\tau_k\Big(\boldsymbol{\beta}_k,\boldsymbol{\pi}^*_{(\mathrm{c}),-k}\Big(\widehat{\mathbf{S}}_{-k}\Big)\Big)+$$
$$\frac{\gamma}{1-\gamma}\cdot\sum_{\widehat{\mathbf{S}}'_k\in\widehat{\mathcal{S}}_k}\mathbb{P}\Big(\widehat{\mathbf{S}}'_k|\widehat{\mathbf{S}}_k,\Big(\varphi_k\Big(\boldsymbol{\beta}_k,\boldsymbol{\pi}^*_{(\mathrm{c}),-k}\Big(\widehat{\mathbf{S}}_{-k}\Big)\Big),\pi^*_{(\mathrm{t}),k}\Big(\widehat{\mathbf{S}}_k\Big),\pi^*_{(\mathrm{p}),k}(\mathbf{S}_k)\Big)\Big)\cdot V_k\Big(\widehat{\mathbf{S}}'_k\Big)\Bigg\} \tag{29}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



the post-decision Q-factor of an MU. More specifically, for each MU $k$, we model the Q-factor in (31) by

$$Q_k\Big(\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k)\Big)\approx Q_k\Big(\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k);\boldsymbol{\theta}_k\Big),\quad(37)$$

$\forall(\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k))\in\widehat{\mathcal{S}}_k\times\{0,1\}\times\mathcal{X}\times\mathcal{D}$, and the post-decision Q-factor in (32) by

$$\widetilde{Q}_k\Big(\widetilde{\mathbf{S}}_k,(\varphi_k,X_k,R_k)\Big)\approx\widetilde{Q}_k\Big(\widetilde{\mathbf{S}}_k,(\varphi_k,X_k,R_k);\tilde{\boldsymbol{\theta}}_k\Big),\quad(38)$$

$\forall(\widetilde{\mathbf{S}}_k,(\varphi_k,X_k,R_k))\in\widehat{\mathcal{S}}_k\times\{0,1\}\times\mathcal{X}\times\mathcal{D}$, where $\boldsymbol{\theta}_k$ and $\tilde{\boldsymbol{\theta}}_k$ denote, respectively, the vectors of parameters that are associated with DQN-I of the Q-factor and DQN-II of the post-decision Q-factor. Similar to the A2C architecture [39], DQN-I with $\boldsymbol{\theta}_k$ of MU $k$ in the proposed deep RL scheme estimates the Q-factor values while DQN-II with $\tilde{\boldsymbol{\theta}}_k$ approximates the best-response control policy suggested by DQN-I [54], [55]. MU $k$ learns $\boldsymbol{\theta}_k$ and $\tilde{\boldsymbol{\theta}}_k$, rather than the Q-factor and the post-decision Q-factor values according to (35) and (36). The implementation of the proposed deep RL scheme is illustrated in Fig. 2.

During the deep RL process, each MU $k\in\mathcal{K}$ in the system is equipped with a finite replay memory $\mathcal{M}_k^j=\{\mathbf{y}_k^{j-M+1},\cdots,\mathbf{y}_k^j\}$ to store the most recent $M$ historical experiences up to a decision epoch $j$, where an experience $\mathbf{y}_k^{j-m+1}$ ($1\le m\le M$) given by (39) (shown at the bottom of Page 11) happens at the transition between two consecutive decision epochs $j-m$ and $j-m+1$.

*1) DQN-I Training:* Each MU $k\in\mathcal{K}$ maintains an DQN-I as well as a target DQN-I, which are $Q_k(\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k);\boldsymbol{\theta}_k^j)$ and $Q_k(\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k);\boldsymbol{\theta}_k^{j,-})$ with $\boldsymbol{\theta}_k^j$ and $\boldsymbol{\theta}_k^{j,-}$ being the associated vectors of parameters at each decision epoch $j$ and from a previous decision epoch before epoch $j$, respectively. To perform experience replay [56], MU $k$ randomly samples a mini-batch $\mathcal{Y}_k^j\subseteq\mathcal{M}_k^j$ from the replay memory $\mathcal{M}_k^j$ at each decision epoch $j$ to train DQN-I. The training objective is to update the parameters $\boldsymbol{\theta}_k^j$ of DQN-I in the direction of minimizing the loss function as in (40).

*2) DQN-II Training:* At each decision epoch $j$, we designate $\tilde{\boldsymbol{\theta}}_k^j$ as the parameters associated with DQN-II of each MU $k\in\mathcal{K}$ in the system. Holding $\boldsymbol{\theta}_k^j$ from DQN-I fixed, MU $k$ adjusts $\tilde{\boldsymbol{\theta}}_k^j$ to minimize the loss function given by (41).

In Algorithm 1, we briefly summarize the procedure of the proposed online deep RL scheme implemented by each MU $k\in\mathcal{K}$ in the air-ground integrated MEC system.

**Algorithm 1** Online Deep RL Scheme for Learning Q-Factor and Post-Decision Q-Factor of Each MU $k\in\mathcal{K}$

1: **initialize** the replay memory $\mathcal{M}_k^j$, the mini-batch $\mathcal{Y}_k^j$, the local state $\widehat{\mathbf{S}}_k^j$, and an DQN-I, a target DQN-I as well as an DQN-II with parameters $\boldsymbol{\theta}_k^j$, $\boldsymbol{\theta}_k^{j,-}$ and $\tilde{\boldsymbol{\theta}}_k^j$, for $j=1$.
2: **repeat**
3: At the beginning of decision epoch $j$, MU $k$ first takes the observation of $\widehat{\mathbf{S}}_k^j$ as an input to DQN-I with parameters $\boldsymbol{\theta}_k^j$, and then selects $(z_k^j,X_k^j,R_k^j)$ randomly with probability $\epsilon$ or $(z_k^j,X_k^j,R_k^j)$ that maximizes $Q_k(\widehat{\mathbf{S}}_k^j,(z_k^j,X_k^j,R_k^j);\boldsymbol{\theta}_k^j)$ with probability $1-\epsilon$.
4: MU $k$ computes the bid $\boldsymbol{\beta}_k^j$ according to (33), (25) and (26) with the approximated Q-factor in (37) and post-decision Q-factor in (38), and sends it to the RO.
5: With the bids from all MUs, the RO determines the auction winners $\boldsymbol{\phi}^j$ and the channel allocation $\boldsymbol{\rho}_k^j$ according to (7), and calculates the payment $\tau_k^j$ according to (8).
6: After channel allocation, MU $k$ performs the decisions $X_k^j$ and $\varphi_k^j\cdot R_k^j$.
7: MU $k$ achieves a payoff $\ell_k(\mathbf{S}_k^j,(\varphi_k^j,X_k^j,R_k^j))$ and observes $\widehat{\mathbf{S}}_k^{j+1}$ at the next epoch $j+1$.
8: MU $k$ fills the replay memory $\mathcal{M}_k^j$ with the latest experience $\mathbf{y}_k^j$.
9: With a randomly sampled mini-batch $\mathcal{Y}_k^j$ from $\mathcal{M}_k^j$, MU $k$ adjusts $\boldsymbol{\theta}_k^j$ of DQN-I and $\tilde{\boldsymbol{\theta}}_k^j$ of DQN-II by minimizing the loss functions in (40) and (41), respectively.
10: MU $k$ regularly resets the target DQN-I with $\boldsymbol{\theta}_k^{j+1,-}=\boldsymbol{\theta}_k^j$, and otherwise $\boldsymbol{\theta}_k^{j+1,-}=\boldsymbol{\theta}_k^{j,-}$.
11: The decision epoch index is updated by $j\leftarrow j+1$.
12: **until** A predefined stopping condition is satisfied.

## V. Numerical Experiments

In order to quantitatively evaluate the performance gained from the proposed deep RL scheme, we conduct numerical experiments based on TensorFlow [57].

### A. Parameter Settings

We set up an experimental scenario of the RAN covering a $0.4\times0.4$ Km$^2$ square area, where there are $B=4$ BSs and $|\mathcal{K}|=20$ MUs. The BSs are placed at equal distance apart, and the square area is divided into $|\mathcal{L}|=1600$ locations with each representing a small area of $10\times10$ m$^2$. The flying altitude of the UAV is kept to $H=100$ meters. For each

$$Q_k^{j+1}\Big(\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k)\Big)=Q_k^j\Big(\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k)\Big)+$$
$$\alpha^j\cdot\Bigg((1-\gamma)\cdot\ell_k(\mathbf{S},(\varphi_k,X_k,R_k))+\gamma\cdot\max_{\varphi_k',X_k',R_k'}Q_k^j\Big(\widehat{\mathbf{S}}_k',(\varphi_k',X_k',R_k')\Big)-Q_k^j\Big(\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k)\Big)\Bigg)\quad(35)$$
$$\widetilde{Q}_k^{j+1}\Big(\widetilde{\mathbf{S}}_k,(\varphi_k,X_k,R_k)\Big)=\widetilde{Q}_k^j\Big(\widetilde{\mathbf{S}}_k,(\varphi_k,X_k,R_k)\Big)+$$
$$\alpha^j\cdot\Bigg(\gamma\cdot\max_{\varphi_k',X_k',R_k'}Q_k^j\Big(\widehat{\mathbf{S}}_k',(\varphi_k',X_k',R_k')\Big)-\widetilde{Q}_k^j\Big(\widetilde{\mathbf{S}}_k,(\varphi_k,X_k,R_k)\Big)\Bigg)\quad(36)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



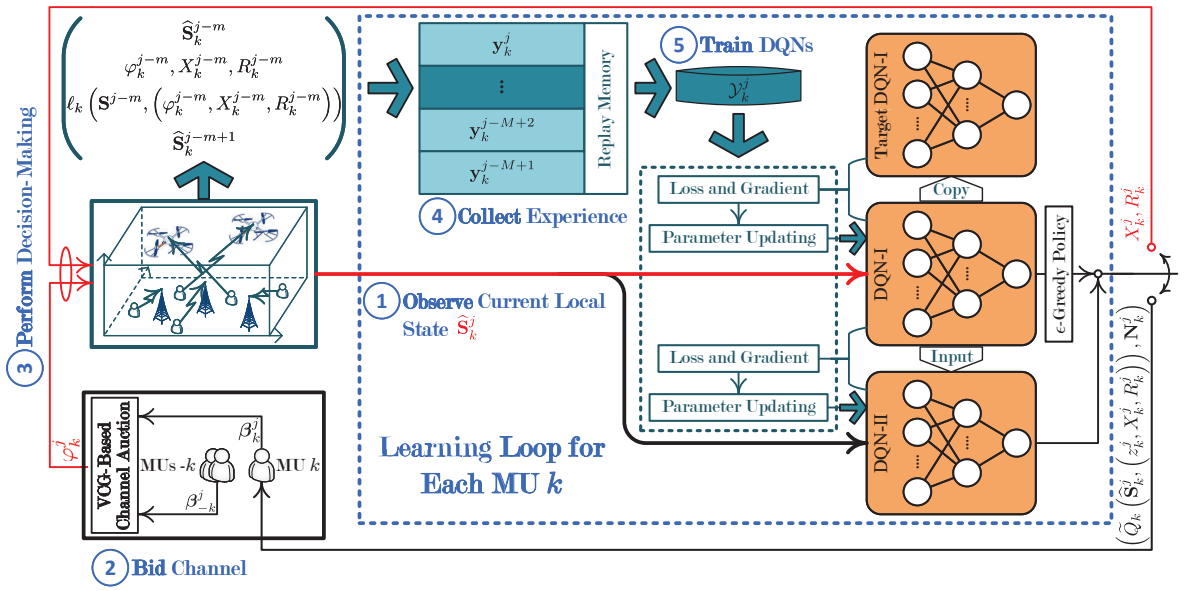


Fig. 2. Implementation of the proposed deep RL scheme to approximate the Q-factor and the post-decision Q-factor of each MU $k \in \mathcal{K}$ in the system.

MU $k \in \mathcal{K}$ in the system, $G^j_{b,k}$ and $G^j_{(\mathrm{v}),k}$, $\forall b \in \mathcal{B}$ and $\forall j$, follow the channel model in [19] and the LOS model in [58], respectively. The state transition probability matrices underlying the Markov mobilities of the UAV and all MUs are independently and randomly generated. We design the DQN-I and the DQN-II of an MU to be with two hidden layers, each of which contains 32 neurons. ReLU is selected as the activation function [59] and Adam as the optimizer [60]. Other parameter values are listed in Table II.

For the performance comparisons, we develop the following four baseline schemes as well.

1) *Local Processing (Baseline 1)* – Each MU processes the arriving computation tasks only at the local device, and hence no channel auction is involved.
2) *Ground MEC Server Execution (Baseline 2)* – Each MU always offloads the computations to the ground MEC server for execution.
3) *UAV Execution (Baseline 3)* – All computation tasks from the pre-processing buffer of each MU are executed by the VMs at the UAV.
4) *Greedy Processing (Baseline 4)* – Whenever possible, a buffered computation task is processed locally or executed remotely via the better link of the two between the MU and the server as well as the UAV.

TABLE II
PARAMETER VALUES IN EXPERIMENTS.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $D_{(\max)}$ | 10 | $\mu$ | 500 Kbits |
| $\vartheta$ | 1300 | $A_{(\max)}$ | 30 seconds |
| $\eta$ | 1 MHz | $\sigma^2$ | $-144$ dBm/Hz |
| $\delta$ | 1 second | $P_{(\max)}$ | 3 Watt |
| $\varpi_k$ | 10, $\forall k$ | $\omega_k$ | 2, $\forall k$ |
| $\varrho$ | 1 GHz | $\xi$ | $10^{-2}$ seconds |
| $\chi_0$ | $2 \cdot 10^7$ bits/second | $\varepsilon$ | 0.2 |
| $\varsigma$ | $10^{-27}$ | $M$ | 5000 |

Implementing Baselines 2, 3 and 4 during each decision epoch, an MU defines the valuation of winning the channel auction as the utility that can be potentially achieved from transmitting a maximum number of input data packets.

$$\mathbf{y}_k^{j-m+1} = \left(\widehat{\mathbf{S}}_k^{j-m}, \left(\varphi_k^{j-m}, X_k^{j-m}, R_k^{j-m}\right), \ell_k\left(\mathbf{S}^{j-m}, \left(\varphi_k^{j-m}, X_k^{j-m}, R_k^{j-m}\right)\right), \widehat{\mathbf{S}}_k^{j-m+1}\right) \tag{39}$$

$$\mathsf{LOSS}_{(\text{DQN-I}),k}\left(\boldsymbol{\theta}_k^j\right) = \mathsf{E}_{\left\{\left(\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k),\ell_k(\mathbf{S},(\varphi_k,X_k,R_k)),\widehat{\mathbf{S}}'_k\right)\in\mathcal{Y}_k^j\right\}}\left[\left(\begin{array}{c}(1-\gamma)\cdot\ell_k(\mathbf{S},(\varphi_k,X_k,R_k))+\\ \gamma\cdot Q_k\left(\widehat{\mathbf{S}}'_k,\underset{\varphi'_k,X'_k,R'_k}{\arg\max}Q_k\left(\widehat{\mathbf{S}}'_k,(\varphi'_k,X'_k,R'_k);\boldsymbol{\theta}_k^j\right);\boldsymbol{\theta}_k^{j,-}\right)-\\ Q_k\left(\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k);\boldsymbol{\theta}_k^j\right)\end{array}\right)^2\right] \tag{40}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



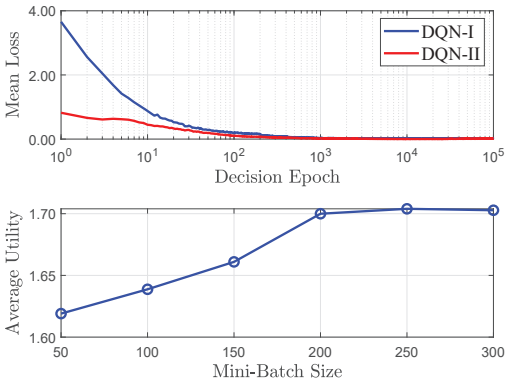


Fig. 3. Illustration of convergence speed of the proposed deep RL scheme in terms of mean losses (namely, $(1/|\mathcal{K}|)\cdot\sum_{k\in\mathcal{K}}\mathsf{LOSS}_{(\text{DQN-I}),k}(\boldsymbol{\theta}_k^j)$ and $(1/|\mathcal{K}|)\cdot\sum_{k\in\mathcal{K}}\mathsf{LOSS}_{(\text{DQN-II}),k}(\tilde{\boldsymbol{\theta}}_k^j)$) versus decision epoch $j$ (upper) and average utility performance per MU across the learning procedure versus batch sizes (lower).

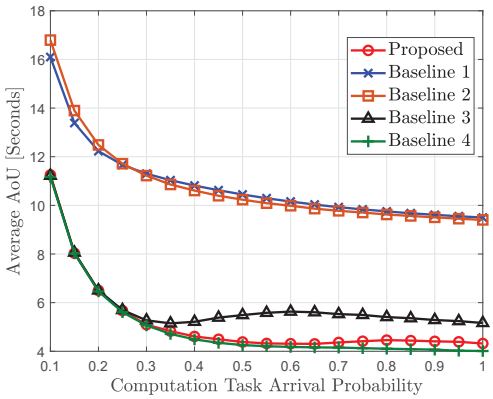


Fig. 4. Average AoU performance per MU across the learning procedure versus computation task arrival probability.

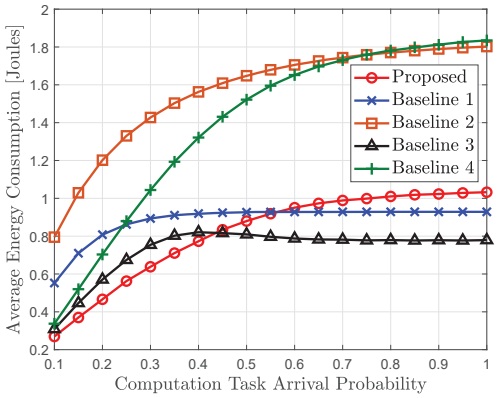


Fig. 5. Average energy consumption per MU across the learning procedure versus computation task arrival probability.

### B. Experiment Results

*1) Experiment 1 – Convergence Performance:* The goal of the first experiment is to validate if the air-ground integrated MEC system remains stable when implementing the proposed online deep RL scheme for information freshness-aware task offloading. We fix the computation task arrival probability and the number of channels to be $\lambda = 0.3$ and $|\mathcal{C}| = 18$, respectively. For each MU $k \in \mathcal{K}$, we set the mini-batch size as $|\mathcal{Y}_k^j| = 200$, $\forall j$. We plot the variations in the mean losses $(1/|\mathcal{K}|)\cdot\sum_{k\in\mathcal{K}}\mathsf{LOSS}_{(\text{DQN-I}),k}(\boldsymbol{\theta}_k^j)$ and $(1/|\mathcal{K}|)\cdot\sum_{k\in\mathcal{K}}\mathsf{LOSS}_{(\text{DQN-II}),k}(\tilde{\boldsymbol{\theta}}_k^j)$ over all the MUs versus the decision epochs in the upper subplot in Fig. 3, which shows that the proposed scheme converges within $10^4$ epochs. In the lower subplot in Fig. 3, we plot the average utility performance per MU with various mini-batch sizes under the given replay memory capacity. It is obvious from (40) and (41) that for each MU, a larger mini-batch size results in a more stable gradient estimate, i.e., a smaller variance, hence a better average utility performance across the learning procedure. When the mini-batch size exceeds 200, the average utility performance improvement saturates. In Experiments 2 and 3, we hence continue to use a mini-batch of size 200 for all MUs to strike a balance between the performance improvement and the mini-batch sampling complexity.

*2) Experiment 2 – Performance under Different Computation Task Arrival Probabilities:* In this experiment, we aim to demonstrate the average performance per MU per decision epoch in terms of the average AoU, the average energy consumption and the average utility under different computation task arrival probabilities. We assume there are $|\mathcal{C}| = 16$ channels in the system, which can be utilized among the non-cooperative MUs to access the computing service provided by the third-party SP. The simulated results are exhibited in Figs. 4, 5 and 6. Fig. 4 illustrates the average AoU per MU. Fig. 5 illustrates the average energy consumption per MU. Fig. 6 illustrates the average utility per MU.

Each plot compares the performance of the proposed deep RL scheme with the four baseline task offloading schemes. From Fig. 6, it can be observed that the proposed scheme achieves the best performance in average utility per MU. Fig. 4 shows that the comparable average AoU performance can be realized between the proposed scheme and Baseline 4. As the computation task arrival probability increases, each MU consumes more energy for task processing in order to maintain the information freshness, as can be seen from Fig. 5. Note that

$$\mathsf{LOSS}_{(\text{DQN-II}),k}\left(\tilde{\boldsymbol{\theta}}_k^j\right) = \mathsf{E}_{\left\{\left(\widehat{\mathbf{S}}_k,(\varphi_k,X_k,R_k),\ell_k(\mathbf{S},(\varphi_k,X_k,R_k)),\widehat{\mathbf{S}}'_k\right)\in\mathcal{Y}_k^j\right\}}\left[\left(\gamma\cdot\max_{\varphi'_k,X'_k,R'_k}Q_k\left(\widehat{\mathbf{S}}'_k,(\varphi'_k,X'_k,R'_k);\boldsymbol{\theta}_k^j\right)-\widetilde{Q}_k\left(\widetilde{\mathbf{S}}_k,(\varphi_k,X_k,R_k);\tilde{\boldsymbol{\theta}}_k^j\right)\right)^2\right] \quad (41)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



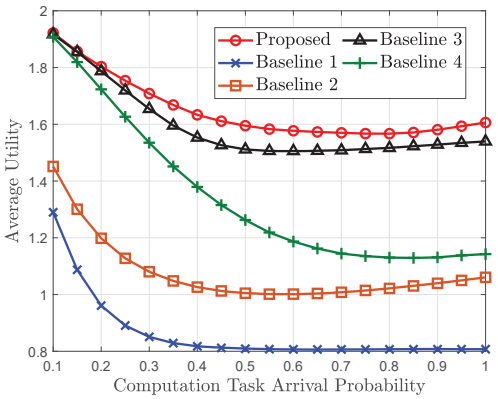


Fig. 6. Average utility performance per MU across the learning procedure versus computation task arrival probability.

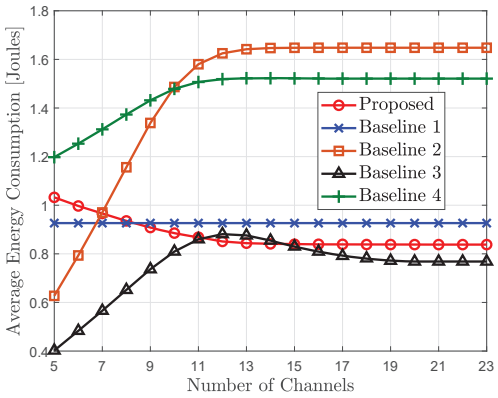


Fig. 8. Average energy consumption per MU across the learning procedure versus number of channels.

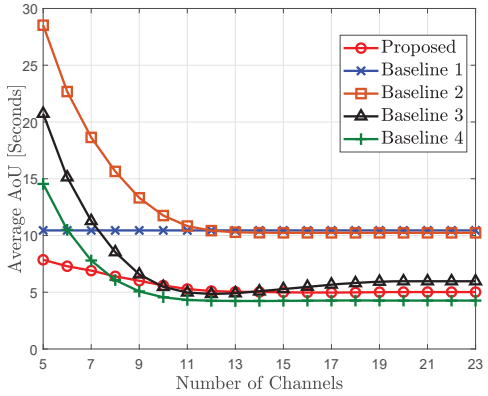


Fig. 7. Average AoU performance per MU across the learning procedure versus number of channels.

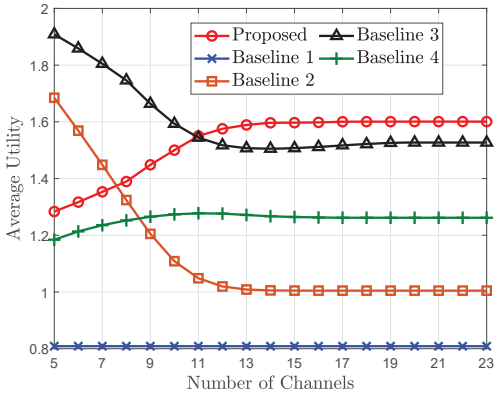


Fig. 9. Average utility performance per MU across the learning procedure versus number of channels.

when implementing Baseline 3, the average energy consumption per MU first increases and then slightly decreases, which is due to the fact that the maximum transmit power at the device of each MU and the constrained computation service rate of an VM at the UAV limit the transmissions of input data packets during a decision epoch. Similar observations can be made from the curves of the proposed scheme and Baseline 3 in Figs. 4, 7 and 8. Furthermore, Baselines 1, 2 and 4 show monotonic performance in the average AoU and the average energy consumption, as can be expected. With the chosen weighting constant values, the AoU increasingly dominates the utility function value as the energy consumption increases, which conforms the average utility performance trends of the proposed scheme as well as Baselines 2, 3 and 4.

*3) Experiment 3 – Performance with Changing Number of Channels:* The last experiment simulates the average performance per MU per decision epoch from the proposed online deep RL scheme and the four baselines versus the numbers of channels. In experiment, the computation task arrival probability is selected as $\lambda = 0.5$. The average AoU, average energy consumption and average utility per MU across the entire learning procedure are depicted in Figs. 7, 8 and 9, respectively. It can be easily observed from Fig. 7 that as the number of available channels increases, the average AoU decreases. The more channels available in the system, the more likely an MU is able to obtain one channel from the auction. Therefore, with Baselines 2, 3 and 4, the MU consumes more energy to offload more input data packets for remote execution, while with the proposed deep RL scheme, there are more opportunities for the MU to have a computation task executed remotely with less energy consumption compared with the local processing, as shown in Fig. 8. Though the average AoU from the proposed scheme is smaller than that from Baseline 3, the weight choices in utility function make Baseline 3 outperforming the proposed scheme in average utility when the number of channels is small, as explained in Experiment 2. With Baseline 1, the average performance does not change since all MUs do not participate the channel auction. Last but not least, both Experiments 2 and 3 tell that the proposed deep RL scheme achieves promising average utility performance while keeping the information fresh for the MUs.

## VI. Conclusions

In this paper, the purpose is to optimize the information freshness-aware task offloading in an air-ground integrated MEC system. We formulate the interactions among the non-

cooperative MUs across the infinite time-horizon as a stochastic game. To approach the NE, each MU forms conjectures of the system states at other competing MUs with the local payment and computation service rate observations, which enables the transformation of the stochastic game into a single-agent MDP. We then derive an online deep RL scheme that maintains two separate DQNs for each MU to approximate the Q-factor and the post-decision Q-factor. Implementing the proposed deep RL scheme, each MU makes the decisions of channel auction, computation task offloading and input data packet scheduling with only the local information. Numerical experiments confirm that compared with the four baselines, our scheme achieves a better tradeoff between the AoU and the energy consumption for all MUs in the system.

## REFERENCES


[1] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Q4 2017.

[2] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, Q2 2020.

[3] X. Chen, C. Wu, Z. Liu, N. Zhang, and Y. Ji, "Computation offloading in beyond 5G networks: A distributed learning framework and applications," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 56–62, Apr. 2021.

[4] M. S. Elbamby, C. Perfecto, C.-F. Liu, J. Park, S. Samarakoon, X. Chen, and M. Bennis, "Wireless edge computing with latency and reliability guarantees," *Proc. IEEE*, vol. 107, no. 8, pp. 1717–1737, Aug. 2019.

[5] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[6] B. Li, F. Si, W. Zhao, and H. Zhang, "Wireless powered mobile edge computing with NOMA and user cooperation," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1957–1961, Feb. 2021.

[7] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Risk-aware data offloading in multi-server multi-access edge computing environment," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1405–1418, Jun. 2020.

[8] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[9] C.-F. Liu, M. Bennis, and H. V. Poor, "Latency and reliability-aware task offloading and resource allocation for mobile edge computing," in *Proc. IEEE GLOBECOM WKSHP*, Singapore, Dec. 2017.

[10] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Proc. IEEE ISIT*, Barcelona, Spain, Jul. 2016.

[11] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4005–4018, Jun. 2019.

[12] X. He, R. Jin, and H. Dai, "Deep PDS-Learning for Privacy-Aware Offloading in MEC-Enabled IoT," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4547–4555, Jun. 2019.

[13] 3GPP TR 22.874, "Technical Specification Group Services and System Aspects; Study on traffic characteristics and performance requirements for AI/ML model transfer in 5GS (Release 18)," v0.2.0, Nov. 2020.

[14] "Focus Group on Machine Learning for Future Networks including 5G," https://www.itu.int/en/ITU-T/focusgroups/ml5g/Pages/default.aspx.

[15] "Experiential Network Intelligence (ENI)," https://www.etsi.org/technologies/experiential-networked-intelligence.

[16] "Artificial Intelligence," ISO/IEC JTC 1/SC 42, https://www.iso.org/committee/6794475.html.

[17] "Artificial Intelligence makes Smart BPM Smarter," TM Forum Catalyst Project, https://www.tmforum.org/catalysts/smart-bpm/.

[18] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3072–3108, Q4 2019.

[19] X. Chen, Z. Zhao, C. Wu, M. Bennis, H. Liu, Y. Ji, and H. Zhang, "Multi-tenant cross-slice resource orchestration: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2377–2392, Oct. 2019.

[20] M. A. Abd-Elmagid, A. Ferdowsi, H. S. Dhillon, and W. Saad, "Deep reinforcement learning for minimizing age-of-information in UAV-assisted networks," in *Proc. IEEE GLOBECOM*, Waikoloa, HI, Dec. 2019.

[21] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.

[22] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, Q3 2019.

[23] R. M. de Amorim, J. Wigard, I. Z. Kovacs, T. B. Sorensen, and P. E. Mogensen, "Enabling cellular communication for aerial vehicles: Providing reliability for future applications," *IEEE Veh. Technol. Mag.*, vol. 15, no. 2, pp. 129–135, Jun. 2020

[24] X. Hu, K.-K. Wong, K. Yang, and Z. Zheng, "UAV-assisted relaying and edge computing: Scheduling and trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4738–4752, Oct. 2019.

[25] B. Shang and L. Liu, "Mobile edge computing in the sky: Energy optimization for air-ground integrated networks," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7443–7456, Aug. 2020.

[26] "Multi-access edge computing (MEC); Phase 2: Use cases and requirements," Oct. 2018, ETSI GS MEC 002 V2.1.1. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/02.01.01_60/gs_MEC002v020101p.pdf [Accessed: 9 Mar. 2021].

[27] A. Asheralieva and D. Niyato, "Hierarchical game-theoretic and reinforcement learning framework for computational offloading in UAV-enabled mobile edge computing networks with multiple service providers," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8753–8769, Oct. 2019.

[28] Z. Liang, Y. Liu, T.-M. Lok, and K. Huang, "Multiuser computation offloading and downloading for edge computing with virtualization," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4298–4311, Sep. 2019.

[29] X. Wang and L. Duan, "Economic analysis of unmanned aerial vehicle (UAV) provided mobile services," *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 1804–1816, May 2021.

[30] Q. Kuang, J. Gong, X. Chen, and X. Ma, "Analysis on computation-intensive status update in mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4353–4366, Apr. 2020.

[31] X. Chen, C. Wu, T. Chen, H. Zhang, Z. Liu, Y. Zhang, and M. Bennis, "Age of information-aware radio resource management in vehicular networks: A proactive deep reinforcement learning perspective," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2268–2281, Apr. 2020.

[32] R. D. Yates, "The age of information in networks: Moments, distributions, and sampling," *IEEE Trans. Inf. Theory*, vol. 66, no. 9, pp. 5712–5728, Sep. 2020.

[33] Y. Dong, Z. Chen, S. Liu, P. Fan, and K. B. Letaief, "Age-upon-decisions minimizing scheduling in Internet of things: To be random or to be deterministic?," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1081–1097, Feb. 2020.

[34] R. Li, Q. Ma, J. Gong, Z. Zhou, and X. Chen, "Age of processing: Age-driven status sampling and processing offloading for edge computing-enabled real-time IoT applications," *IEEE Internet Things J.*, vol. 8, no. 19, pp. 14471–14484, Oct. 2021.

[35] C. Xu, H. H. Yang, X. Wang, and T. Q. S. Quek, "Optimizing information freshness in computing-enabled IoT networks," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 971–985, Feb. 2020.

[36] Z. Ji and K. J. R. Liu, "Dynamic spectrum sharing: A game theoretical overview," *IEEE Commun. Mag.*, vol. 45, no. 5, pp. 88–94, May 2007.

[37] B. Edelman, M. Ostrovsky, and M. Schwarz, "Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords," *Am. Econ. Rev.*, vol. 97, no. 1, pp. 242–259, Mar. 2007.

[38] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI*, Phoenix, AZ, Feb. 2016.

[39] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. ICML*, New York City, NY, Jun. 2016.

[40] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, "Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7957–7969, Aug. 2019.

[41] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications




[42] Y. Wan, K. Namuduri, Y. Zhou, and S. Fu, "A smooth-turn mobility model for airborne networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 7, pp. 3359–3370, Sep. 2013.

[43] X. Xi, X. Cao, P. Yang, Z. Xiao, and D. Wu, "Efficient and fair network selection for integrated cellular and drone-cell networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 923–937, Jan. 2019.

[44] R. Amer, W. Saad, and N. Marchetti, "Mobility in the sky: Performance and mobility analysis for cellular-connected UAVs," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 3229–3246, May 2020.

[45] P. Zhou, T. Braud, A. Zavodovski, Z. Liu, X. Chen, P. Hui, and J. Kangasharju, "Edge-facilitated augmented vision in vehicle-to-everything networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12187–12201, Oct. 2020.

[46] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *J. VLSI Signal Process. Syst.*, vol. 13, no. 2–3, pp. 203–221, Aug. 1996.

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

[48] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *IEEE Netw.*, vol. 24, no. 2, pp. 36–41, Mar./Apr. 2010.

[49] D. Adelman and A. J. Mersereau, "Relaxations of weakly coupled stochastic dynamic programs," *Oper. Res.*, vol. 56, no. 3, pp. 712–727, Jan. 2008.

[50] A. M. Fink, "Equilibrium in a stochastic $n$-person game," *J. Sci. Hiroshima Univ. Ser. A-I*, vol. 28, pp. 89–93, 1964.

[51] X. Chen, Z. Han, H. Zhang, G. Xue, Y. Xiao, and M. Bennis, "Wireless resource scheduling in virtualized radio access networks using stochastic learning," *IEEE Trans. Mobile Comput.*, vol. 17, no. 4, pp. 961–974, Apr. 2018.

[52] C. Kroer and T. Sandholm, "Imperfect-recall abstractions with bounds in games," in *Proc. ACM EC*, Maastricht, the Netherlands, Jul. 2016.

[53] Apple, "The future is here: iPhone X", 2017. [Online]. Available: https://www.apple.com/newsroom/2017/09/the-future-is-here-iphone-x/ [Accessed: 16 Oct. 2021].

[54] J. A. Ramírez-Hernández and E. Fernandez, "Optimization of preventive maintenance scheduling in semiconductor manufacturing models using a simulation-based approximate dynamic programming approach," in *IEEE CDC*, Atlanta, GA, Dec. 2010.

[55] Y. Wang and D. R. Jiang, "Structured actor-critic for managing public health points-of-dispensing," 2021. [Online]. Available: https://arxiv.org/pdf/1806.02490.pdf [Accessed: 16 Oct. 2021].

[56] L.-J. Lin, "Reinforcement learning for robots using neural networks," Carnegie Mellon University, 1992.

[57] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *Proc. OSDI*, Savannah, GA, Nov. 2016.

[58] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4983–4996, Dec. 2016.

[59] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. ICML*, Haifa, Israel, Jun. 2010.

[60] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. ICLR*, San Diego, CA, May 2015.


**Xianfu Chen** (Member, IEEE) received his Ph.D. degree with honors in Signal and Information Processing, from the Department of Information Science and Electronic Engineering (ISEE) at Zhejiang University, Hangzhou, China, in March 2012. Since April 2012, he has been with the VTT Technical Research Centre of Finland, Oulu, Finland, where he is currently a Senior Scientist. His research interests cover various aspects of wireless communications and networking, with emphasis on human-level and artificial intelligence for resource awareness in next-generation communication networks. He is the recipient of the 2021 IEEE Communications Society Outstanding Paper Award and the 2021 IEEE Internet of Things Journal Best Paper Award. He received the Exemplary Reviewer Certificate of IEEE Transactions on Communications in 2021. He serves as an Editor for IEEE Transactions on Cognitive Communications and Networking as well as Microwave and Wireless Communications, an Academic Editor for Wireless Communications and Mobile Computing, and an Associate Editor for China Communications. He has been a Guest Editor for several international journals, including IEEE Wireless Communications Magazine. He is a Vice Chair of the IEEE Special Interest Group on Big Data with Computational Intelligence and the IEEE Special Interest Group on AI Empowered Internet of Vehicles.

**Celimuge Wu** (Senior Member, IEEE) received his PhD degree from The University of Electro-Communications, Japan in 2010. He has been an Associate Professor of The University of Electro-Communications since 2015. His research interests include vehicular networks, edge computing, IoT, intelligent transport systems, and application of machine learning in wireless networking and computing. He serves as an Associate Editor of IEEE Transactions on Network Science and Engineering, IEEE Transactions on Green Communications and Networking, IEEE Open Journal of the Computer Society, Wireless Networks, and IEICE Transactions on Communications. He also has been a Guest Editor of IEEE Transaction on Intelligent Transportation Systems, IEEE Transactions on Emerging Topics in Computational Intelligence, IEEE Computational Intelligence Magazine, etc. He is the chair of IEEE TCGCC Special Interest Group on Green Internet of Vehicles and IEEE TCBD Special Interest Group on Big Data with Computational Intelligence. He is a recipient of the 2021 IEEE Communications Society Outstanding Paper Award, the 2021 IEEE Internet of Things Journal Best Paper Award, and the IEEE Computer Society 2019 Best Paper Award Runner-Up.

**Tao Chen** (Senior Member, IEEE) received his B.E. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 1996, and Ph.D. degree from University of Trento, Trento, Italy, in 2007, both in telecommunications engineering. He is currently a Senior Scientist with the VTT Technical Research Center of Finland, an Adjunct Professor at the University of Jyväskylä, Finland, and an Honorary Professor at the University of Kent, UK. He was the coordinator of the 5G PPP CO-HERENT project, which studied programmability in RAN, and the technical manager of the 5G PPP trial project 5G-DRIVE. His research interests include artificial intelligence and programmability in mobile networks, new spectrum access, and resource management in heterogeneous networks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2021.3126075, IEEE Journal on Selected Areas in Communications



**Zhi Liu** (Senior Member, IEEE) received the B.E. degree from the University of Science and Technology of China, Hefei, China, and the Ph.D. degree in informatics in National Institute of Informatics. He is currently an Associate Professor at The University of Electro-Communications, Japan. His research interest includes video network transmission and mobile edge computing. He is now an editorial board member of Springer wireless networks and IEEE Open Journal of the Computer Society. He was the recipient of the IEEE StreamComm 2011 best student paper award, the 2015 IEICE Young Researcher Award and the ICOIN 2018 best paper award.

**Honggang Zhang** (Senior Member, IEEE) is a Full Professor with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. He was an Honorary Visiting Professor with the University of York, York, U.K., and an International Chair Professor of Excellence with the Université Européenne de Bretagne and Supélec, France. He has coauthored and edited two books: Cognitive Communications-Distributed Artificial Intelligence (DAI), Regulatory Policy and Economics, Implementation (John Wiley & Sons) and Green Communications: Theoretical Fundamentals, Algorithms and Applications (CRC Press), respectively. His research interests include cognitive radio and networks, green communications, mobile computing, machine learning, artificial intelligence, and Internet of Intelligence (IoI). He was the leading Guest Editor of the IEEE Communications Magazine special issues on Green Communications. He served as the Series Editor for the IEEE Communications Magazine for its Green Communications and Computing Networks Series from 2015 to 2018 and the Chair of the Technical Committee on Cognitive Networks of the IEEE Communications Society from 2011 to 2012. He is an Associate Editor-in-Chief of China Communications.

**Mehdi Bennis** (Fellow, IEEE) is a Professor at the Centre for Wireless Communications, University of Oulu, Finland. His main research interests are in radio resource management, heterogeneous networks, game theory and machine learning in 5G networks and beyond. He has co-authored one book and published more than 200 research papers in international conferences, journals and book chapters. He has been the recipient of several awards, including the 2015 Fred W. Ellersick Prize from the IEEE Communications Society, the 2016 Best Tutorial Prize from the IEEE Communications Society, the 2017 EURASIP Best paper Award for the Journal of Wireless Communications and Networks, the all-University of Oulu award for research, the 2019 IEEE ComSoc Radio Communications Committee Early Achievement Award, the 2021 IEEE Communications Society Outstanding Paper Award, and the 2021 IEEE Internet of Things Journal Best Paper Award. He is an Editor of IEEE Transactions on Communications.

**Hang Liu** (Senior Member, IEEE) joined the Catholic University of America in 2013, where he currently is a Professor with the Department of Electrical Engineering and Computer Science. Prior to joining the Catholic University of America, he had more than 10 years of research experience in networking industry and worked in senior research and management positions at several companies. He has published more than 100 papers in leading journals and conferences, and received two best paper awards and one best student paper award. He is the inventor/co-inventor of over 50 granted US patents. He has also made many contributions to the IEEE 802 wireless standards and 3GPP standards, and was the editor of the IEEE 802.11aa standard and the rapporteur of a 3GPP work item. He received his Ph.D. degree in Electrical Engineering from the University of Pennsylvania. His research interests include wireless communications and networking, millimeter wave communications, dynamic spectrum management, mobile computing, Internet of Things, future Internet architecture and protocols, mobile content distribution, video streaming, and network security.

**Yusheng Ji** (Senior Member, IEEE) received the B.E., M.E., and D.E. degrees in electrical engineering from the University of Tokyo. She joined the National Center for Science Information Systems (NACSIS), Japan, in 1990. She is currently a Professor at the National Institute of Informatics (NII), and the Graduate University for Advanced Studies, SOKENDAI, Japan. Her research interests include network architecture, resource management in wireless networks, and mobile computing. She has served as Editor of IEEE Transactions of Vehicular Technology, Symposium Co-Chair of IEEE GLOBECOM in 2012 and 2014, Track Chair of IEEE VTC 2016 Fall and 2017 Fall, General Co-chair of ICT-DM 2018, MSN2020, and Symposium Co-Chair of IEEE ICC 2020. She is an Associate Editor of IEEE Vehicular Technology Magazine, Area Chair of IEEE INFOCOM, and TPC member of IEEE ICC, GLOBECOM, WCNC, etc.