Blockchain-Empowered Resource Allocation in Multi-UAV-Enabled 5G-RAN: A Multi-agent Deep Reinforcement Learning Approach

Abegaz Mohammed Seid, *Member, IEEE*, Aiman Erbad, *Senior Member, IEEE*, Hayla Nahom Abishu, *Graduate Student Member, IEEE*, Abdullatif Albaseer, *Member, IEEE*, Mohammed Abdallah, *Senior Member, IEEE,* and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—In 5G and B5G networks, real-time and secure resource allocation with the common telecom infrastructure is challenging. This problem may be more severe when mobile users are growing and connectivity is interrupted by natural disasters or other emergencies. To address the resource allocation problem, the network slicing technique has been applied to assign virtualized resources to multiple network slices, guaranteeing the 5G-RAN quality of service. Moreover, to tackle connectivity interruptions during emergencies, UAVs have been deployed as airborne base stations, providing various services to ground networks. However, this increases the complexity of resource allocation in the shared infrastructure of 5G-RAN. Therefore, this paper proposes a dynamic resource allocation framework that synergies blockchain and multi-agent deep reinforcement learning for multi-UAV-enabled 5G-RAN to allocate resources to smart mobile user equipment (SMUE) with optimal costs. The blockchain ensures the security of virtual resource transactions between SMUEs, infrastructure providers (InPs), and virtual network operators (VNOs). We formulate a virtualized resource allocation problem as a hierarchical Stackelberg game containing InPs, VNOs, and SMUEs, and then transform it into a stochastic game model. Then, we adopt a Multi-agent Deep Deterministic Policy Gradient (MADDPG) algorithm to solve the formulated problem and obtain the optimal resource allocation policies that maximize the utility function. The simulation results show that the MADDPG method outperforms the state-of-the-art methods in terms of utility optimization and quality of service satisfaction.**

***Index Terms*—Blockchain, 5G-RAN, Multi-agent DRL, Stackelberg game, Network slicing, Virtualization**

I. Introduction

IN the fifth-generation and beyond (B5G) cellular technology, ubiquitous gigantic services, and latency-critical applications such as intelligent transportation systems, virtual reality, augmented reality, healthcare, smart grids, industry 4.0, and others are enabled for all connected mobile users and machines. In 5G and B5G, resource management is the most critical aspect for radio access networks (RANs), since these networks have inadequate and geographically dispersed resources to deliver low latency, high data rates, high reliability, and low energy consumption. Accordingly, resource allocation must meet an unusual level of heterogeneity and the three main 5G deployment requirements identified by 3GPP: massive machine-type communication (mMTC), ultra-reliable low latency communication (URLLC), and enhanced mobile broadband (eMBB) [1], [2]. In this regard, wireless network virtualization (WNV) has been recognized as a promising technology for 5G and B5G cellular networks to handle the resource scarcity in these applications. With WNV, the physical wireless resources can be abstracted and sliced into multiple virtual wireless networks with specific corresponding functionalities that various parties can share by detaching from one another [3], [4]. The objective is to enhance resource utilization by virtualizing infrastructure providers' (InPs) physical resources into multiple virtual networks (slices) and allocating these virtualized resources dynamically to virtual network operators (VNOs). Furthermore, it aims to provide differentiated services that ensure service level agreements for each type of service and improve the flexibility and adaptability of network management [5].

However, the increased traffic at edge computing levels makes such resource allocation challenging. This brings out the need for a general solution to minimize the excessive workload at the edge network and allocate virtualized resources into different slices [6]. In addition, the 5G and B5G networks may be affected by various types of disasters, including earthquakes, volcano eruptions, landslides, and man-made disasters. These disasters disrupt network infrastructure functionality and can even cause real-time applications and other operations in 5G and B5G networks to fail. In response, the unmanned aerial vehicle (UAV) technology has been deployed as an aerial base station (ABS) to improve network availability and handle emergency communication (post-disaster) in the 5G and B5G networks. The deployment of a UAV is flexible, low-cost, supports mobility, and is more accessible than traditional base stations (BSs) [7]–[10]. Due to these merits, UAVs have been widely deployed in 5G and B5G networks as ABS to provide

Abegaz Mohammed Seid, Aiman Erbad, Abdullatif Albaseer, and Mohammed Abdallah are with the Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar. (*corresponding author: Aiman Erbad).*

Hayla Nahom Abishu is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731-CHINA.

M. Guizani is with Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE.

This work was made possible by NPRP-Standard (NPRP-S) Thirteen (13th) Cycle grant NPRP13S-0205-200265 from the Qatar National Research Fund. The findings achieved herein are solely the responsibility of the authors.

Manuscript received —.

various services such as extending the network, allocating resources, and offloading intensive tasks from smart mobile user equipments (SMUEs).

In the 5G and B5G network environments, the traffic is unpredictable and time-varying. Hence, reinforcement learning (RL) is becoming a prominent tool for handling real-time dynamic decision-making problems in 5G and B5G or upcoming 6G networks [11]–[17]. Another issue in the 5G and B5G networks is that security management is more complicated since various ultra-dense heterogeneous devices are connected using new fixed and mobile BSs. Recently, blockchain technology became popular by ensuring security requirements and changing the network architecture [18], [19]. Blockchain is a distributed, immutable, and transparent digital ledger used to ensure the security transactions between network nodes [20]–[22]. Therefore, integrating machine learning (ML) and blockchain into 5G and B5G networks can provide a new paradigm for managing mobile networks and services, enabling applications such as autonomous resource sharing, ubiquitous computing, reliable content-based storage, and intelligent data management [23]. Further, it ensures the security, efficiency, and reliability of resource allocation, sharing, and trading in B5G networks [24]. In [25]–[30], the authors presented blockchain-based machine learning frameworks in 5G and B5G networks for crowdsourcing, spectrum sharing, network slicing architecture, cache content distribution, and secure and intelligent data analytics. Moreover, the UAV-enabled 5G-RAN still faces several difficulties. First, it is challenging to implement an efficient resource allocation mechanism in an emergency on the ground network due to the dynamic UAV-enabled 5G-RAN environment. When SMUEs seek various resources, the MVNOs' servers may get overloaded, or certain SMUEs lack network coverage due to their mobile nature. The second issue that requires thorough investigation is how to motivate VNOs to allocate resources to SMUEs at an optimal price. Third, 5G-RAN security is a crucial concern. The 5G-RAN in aerial-to-ground (ATG) network scenarios requires a secure and intelligent system to provide the anticipated resource allocation service. It is, therefore, necessary to design a framework to optimize resource allocation to maximize utility while minimizing computation costs.

Motivated by the above remarks, we design a new resource allocation framework focusing on UAV-enabled 5G-RAN in emergency communication. We use virtualization technology to partition physical resources and control their usage. We deployed a multi-UAV network that enabled a 5G-RAN to restore the network connections and provide different services to SMUEs. To maintain the security of resource allocation transactions in the ATG virtualized network, we adopt consortium blockchain. Lastly, we formulate the optimization problem of resource allocation in a hierarchical three-stage multi-leader multi-follower (MLMF) Stackelberg game model, where resources are allocated to followers based on the leaders' optimal price and followers' demand. The main contributions of this paper are summarized as follows:

1) We propose a new multi-agent deep reinforcement learning (MADRL)-based dynamic resource allocation framework in a consortium blockchain-empowered multi-UAV-enabled 5G-RAN to allocate different resources in a virtualized network to maximize utility in different layers. In the proposed framework, UAV networks aid the 5G-RAN in providing various services during emergencies or when the network is congested.
2) We formulate the optimization problem into a hierarchical non-cooperative MLMF Stackelberg game model to control the selfishness between leaders and followers and make decisions independently in different layers. In this game, the interaction between InPs and VNOs is an upper-level game where InPs are the leaders that set prices depending on previous pricing strategies of other InPs, and VNOs are the followers who adjust their demands based on the prices set by the InPs. The interaction between VNOs and SMUEs is a lower-level game, where the VNOs are the leaders that determine the prices to lease resources to SMUEs based on the previous pricing strategies of other VNOs and prices set by InPs. The SMUEs determine their demands based on the prices set by VNOs. We transform this optimization problem into a stochastic game model since it is difficult to solve directly with a Stackelberg game.
3) We apply a model-free MADRL approach (multi-agent deep deterministic policy gradient (MADDPG)) to obtain optimal prices and optimal decisions to allocate resources. The agents maximize the long-term rewards in order to maximize the utility function.
4) We conduct extensive simulations to evaluate the performance of the proposed scheme. The simulation results show that the proposed MADDPG algorithm outperforms the baseline algorithms, i.e., multi-agent deep Q-network (MADQN), deep deterministic policy gradient (DDPG), asynchronous advantage actor-critic (A3C), and greedy.

The rest of the paper is organized as follows: Section II introduces the related work. Section III presents the proposed framework and system model. Section IV presents the utility function. Section V discusses the problem formulation and Stackelberg game analysis. The proposed solution is presented in Section VI. Section VII presents the simulation results and analysis. Finally, we conclude this work and future work in Section VIII.

II. Related work

In the last decade, plenty of research has been done on resource allocation problems such as radio resources, computation resources, and transmission power in virtual network technologies. Various optimization techniques were adopted [31]–[33]. Yuan *et al*. [33] presented the UAV network slicing architecture in a 5G network environment based on the 5G services; UAVs can be mapped with specific quality of service (QoS) services depending on them. In [34], the authors studied a resource allocation scheme in air slicing in UAV-assisted cellular vehicle-to-everything communication to maximize bandwidth efficiency. They have adopted the long short-term memory algorithm to control/forecast the vehicle and UAV mobilities. In [5], the authors proposed a dynamic virtual

resource allocation in 5G and B5G based on RAN slicing to ensure the QoS of mobile user services. They formulated a problem using a constrained Markov decision process (MDP) and adopted approximate dynamic programming to maximize the overall-sum transmission rate. To empower the mobile edge computing (MEC) services, [35] combined the virtual network function (VNF) with MEC and proposed a clustered-based adaptive VNF resource allocation scheme to minimize the average end-to-end delay for users in the network. They were grouping the users into the context-aware mechanism. In the UAV-enabled B5G network, researchers studied intelligent virtual resource allocation to enhance coverage and flexible services for profit maximization [36]. Previous studies proposed various schemes to handle different optimization problems on resource allocation. Under a limited spectrum resource, [37] proposed a deep Q-network (DQN) based resource allocation in a 5G network to improve the QoS. In [38], authors proposed a deep reinforcement learning (DRL)-based software-defined satellite-terrestrial network framework for bandwidth and computation resource allocation. They formulated a problem using MDP and solved it using the DQN approach to improve the network performance. Authors in [39] proposed a novel intelligent hierarchical framework for blockchain-enabled spectrum trading for network slicing in 5G-RAN to maximize the utility function. To maximize the incentive, they utilize a three-stage Stackelberg game model for simultaneously handling optimal pricing and demand prediction while deploying the Stackelberg MARL technique to achieve Stackelberg equilibrium.

In [40] and [41], the authors proposed a blockchain-enabled UAV network virtualization architecture. UAVs can assist the Industrial Internet of Things (IIoT) and vehicular networks by ensuring security and privacy, allocating resources, and enhancing terrestrial network coverage. Authors in [42] applied blockchain in UAV-assisted mobile networks for computation resource allocation between UAVs and edge computation stations (ECSs) to secure the transaction. The Stackelberg game model formulates the interaction between UAVs and ECSs to obtain optimal price and resource demands. Yao *et al.* [43], studied a three-stage Stackelberg game-based capacity allocation and caching placement for a multi-UAV collaborative edge caching system. In [44], the authors proposed a hierarchical dynamic pricing framework in the cloud-edge-client collaboration for IoT systems to maximize service providers' utility and improve the service quality for users. Alia *et al.* [45] proposed a dynamic resource allocation and pricing with the combination of blockchain-as-a-service and UAV-enabled MEC networks in IoT systems. In this work, the authors formulated the problem using the stochastic Stackelberg game model and transformed it into a partially observable MDP (POMDP). The authors solved this problem using deep Q-learning (DQL) for BSs with Bayesian deep learning for peers to minimize costs and maximize rewards.

Recently, researchers have focused on combining DRL, game theory, and blockchain technologies to solve multiple optimization problems in a 5G and B5G network [46], [47]. The Stackelberg game is an emerging tool to model the interaction between leaders and followers. It is widely applied in blockchain technology that enables MEC networks for optimization problems such as computation offloading, resource trading, and energy transfer to maximize or minimize utility between leaders and followers. The authors in [47] studied multi-agent-based resource trading in a blockchain-based industrial IoT scenario. They adopted the Stackelberg game to optimize the exchange of resources and prices between leaders and followers.

TABLE I: Comparison of our work with resource allocation and pricing-related works in 5G-RAN.

Features	[38]	[39]	[40], [41]	[42]	[44]	[45]	[46]	[47]	Our work
Multi-tier multiple resource allocation in ATG	×	×	×	×	×	×	✓	×	✓
Blockchain-empowered resource allocation and pricing	×	✓	✓	✓	✓	✓	✓	✓	✓
Multi-agent based resource allocation in 5G/MEC	✓	✓	×	×	×	✓	×	✓	✓
Hierarchical three stage MLMF Stackelberg game model	×	✓	×	×	✓	×	✓	✓	✓
Virtualized ATG network/5G-RAN	✓	✓	×	✓	×	×	✓	×	✓
Blockchain empowered ATG network	×	×	✓	✓	×	×	✓	×	✓
Synergy of blockchain and MADRL in multi-UAV-enabled MEC network/5G-RAN	×	×	×	×	×	✓	×	×	✓

Prior research has focused on computation offloading and resource allocation in MEC/IoT networks without considering virtualized ATG networks. In this paper, we design a virtualized ATG network where the InPs provide various resources to VNOs. VNOs can be UAV virtual network operators (UVNOs), which lease the resources of UAVs from InPs, or mobile virtual network operators (MVNOs), which lease the resources of BSs' from InPs. Then, VNOs allocate these resources to SMUEs in a different virtualized network. We deploy a consortium blockchain and MADRL to improve the network performance of the 5G-RAN and the security of the ATG network. The hierarchical MLMF three-stage Stackelberg game model is utilized to formulate the interaction between InPs, VNOs, and SMUEs to obtain optimal resource allocation decisions and pricing policies. Through interaction with the environment and network entities, the agents learn optimal decisions and pricing policies to maximize their utilities while satisfying their resource constraints and the QoS of SMUEs. We adopted a multi-agent deep deterministic policy gradient (MADDPG) to accelerate the convergence and enhance the robustness of the proposed framework. We compared our work to the existing literature in Table I.

III. Proposed Framework and System model

As shown in Fig. 1, the system model is composed of five main components: physical network, virtual network, blockchain network, DRL model, and UAV network. We assume that the substrate physical infrastructure is a cellular

network, which is abstracted and partitioned into different virtual networks known as slices, where each can be managed independently by either UVNOs or MVNOs. In our scenario, we deployed a clustered UAV network, which is partitioned into different network slices, and each has its own UVNO. The UVNOs and MVNOs provide differentiated services to SMUEs connected to their particular slices. InPs are resource providers who provide virtualized resources to VNOs. VNOs can act as both resource providers and requesters, leasing resources from InPs and subleasing them to SMUEs. The SMUE are resource-constrained devices that request resources from VNOs. These entities can tell the software-defined networking (SDN) controller about their status and state. The SDN then uses this information for proper management and admission control of the ATG network. The SDN controller assigns the slices either to the BSs or UCHs based on total resource demands. The network slicing broker (NSB) assists the resource providers and resource requestor in making a deal in resource allocation between them and selects the proper resource provider to allocate resources to requesters based on optimal price, QoS satisfaction, and resource constraints and demands [48]. The resources allocated within regular blocks are transmission power, computation, and bandwidth. The centralized DRL training is deployed on the SDN controller which observes the ATG network status either under emergency or network traffic for autonomous and decentralized resource management. The agents make a decision independently from local observation and other agents' experience and take the best action to satisfy their needs. It can be coupled with high SMUEs and UAV mobility, effective control, and secure resource allocation in the various virtual operators. We employ a decentralized blockchain network for resource allocation among virtual operators. Each node that participates in the blockchain network has its public keys, private keys, and transaction databases linked with NSB for authentication and other issues.

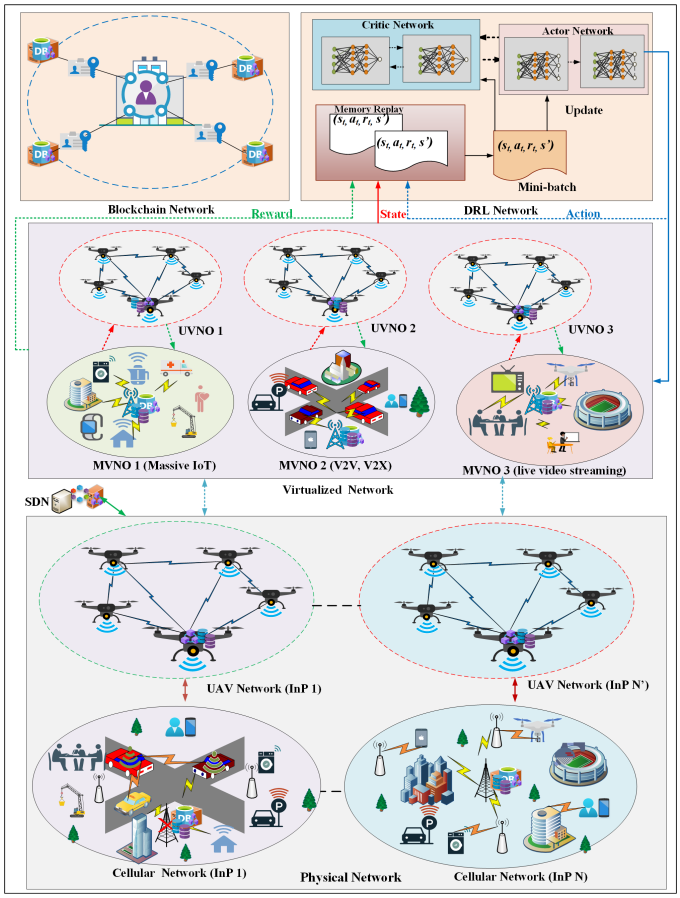


Fig. 1: Virtualized Multi-UAV-enabled 5G-RAN system model.

A. System Model

1) Virtualization Model: This section describes the virtualized ATG system in the B5G emergency scenario. Initially, we assume there is an emergency in a particular area of the physical network environment. The physical network owned by InPs consists of a UAV network and a terrestrial network being abstracted and partitioned into multiple virtual networks (slices) using the WNV technique. Let $\mathcal{Z} = \{1, 2, 3, \ldots, Z\}$ be the set of slices, and the InPs allocate resources to these virtual networks with heterogeneous QoS requirements and lease them to VNOs. The VNOs then allocate these resources and compute the tasks offloaded from the SMUE at optimal cost. Here, to maximize their utility, the VNOs adjust the unit price of each resource type using the proposed game model and sign a contract with the SMUEs. The SMUEs then transfer the digital coins to the VNOs' wallets for the resources allocated to them. Each slice supports the B5G network requirements for accommodating heterogeneous services such as eMBB, uRLLC, and mMTC application services. When a network slice becomes congested or a disaster occurs, UAV networks may allocate resources to SMUEs in the slice. In each slice, the controller assigns the SMUEs into UCHs or BSs of InPs depending on resource demands, emergency level, and the slices' traffic level. In the ATG virtualized network, a slice may not satisfy its SMUEs' resource demands and QoS requirements due to inadequate resources borrowed from other slices. Thus, the UAV or terrestrial network should have idle or available resources that can be leased or assigned to SMUEs. This ensures efficient virtual resource allocation and meets the QoS requirements for the customized slices.

2) Network Model: We consider a wireless ATG network infrastructure sliced into three different virtual networks depending on the B5G scenario, as shown in Fig. 1. Several SMUEs are randomly distributed in their coverage areas and connected to UCHs or BSs. Let $u \in \mathcal{U} = \{1, 2, 3, \ldots, U\}$, and $k \in \mathcal{K} = \{1, 2, 3, \ldots, k\}$ represent a set of UCHs and BSs with a set of MEC servers, respectively. In the proposed framework, the physical network deploys three UCHs and three BSs; the UCHs' resources leased and controlled by the UVNOs can be allocated to SMUEs in an emergency scenario. Let $n \in \mathcal{N} = \{1, 2, 3, \ldots, N\}$ represents a set of InPs, each owning active BS K and UCH U. The InPs lease their slices of resources to VNOs, where VNOs consist of MVNOs ($\mathcal{V} = \{1, 2, 3, \ldots, V\}$) and UVNOs ($\mathcal{V}' = \{1, 2, 3, \ldots, V'\}$). The VNOs provide edge services to SMUEs with the minimum

thresholds of QoS requirements. Furthermore, we consider several edge service requests from SMUEs and the network that have been affected by a disaster. Therefore, various computation tasks are offloaded to MEC servers operating under VNOs, where each computational node operates with limited bandwidth and computation capacity. Let us define $\mathcal{J} = \{1, 2, 3, \ldots, J\}$ and $\mathcal{I} = \{1, 2, 3, \ldots, I\}$ represent the sets of VNOs and SMUEs, respectively, where $\mathcal{J} = \mathcal{V}' \cup \mathcal{V}$. Each SMUE can only be associated with either UCH or BS in a particular time slot t. To avoid intra-cell interference in the ATG network, the orthogonal frequency division multiple access (OFDMA) is used. The coordinates of the SMUE i, UCH u, and BS k are denoted by (x_{iz}, y_{iz}), (x_{uz}, y_{uz}, H_u), and (x_{kz}, y_{kz}), respectively. The horizontal distance between SMUE i and UCH u in slice z, $\mathcal{M} = (x_{uz} - x_{iz})^2 + (y_{uz} - y_{iz})^2$ is expressed as;

$$d_{iu}(t) = \sqrt{\mathcal{M} + H_u^2}, \forall i \in \mathcal{I}, u \in \mathcal{U}. \tag{1}$$

Let the binary variable $\xi_{iuz}(t) \in \{0, 1\}$ indicate the connection status between UCH u and SMUE i. If $\xi_{iuz}(t) = 1$, the SMUE i connects with the UCH u at slice z; otherwise, $\xi_{iuz}(t) = 0$. When the SMUE i is associated with UCH u to offload tasks and access resources from UCH u, it must be within UCH u coverage range. Let $\mathcal{U}_{iz}$ denote the set of UCHs that cover SMUE i, is expressed as $\mathcal{U}_{iz} = \{u | d_{iu}(t) \leq H_u tan\theta_u\}, \forall i \in \mathcal{I}$.

Similarly, the horizontal distance between SMUE i and BS k in slice z is given as;

$$d_{ik}(t) = \sqrt{(x_{kz} - x_{iz})^2 + (y_{kz} - y_{iz})^2}, \forall i \in \mathcal{I}, k \in \mathcal{K}. \tag{2}$$

Each SMUE in the most adjacent UCH u must receive the signal-to-interference-plus-noise ratio (SINR) φ_{iu} and SMUE adjacent to BS k receives the SINR φ_{ik}. The SINR SMUE receives in the adjacent UCH/BS must be greater than or equal to the minimum threshold SINR φ_{thr} from the associated computational node to meet its minimum QoS, which is expressed as;

$$\varphi_{iu} \geq \varphi_{thr}, \varphi_{ik} \geq \varphi_{thr}, \forall i \in \mathcal{I}. \tag{3}$$

Air-to-Ground Channel Model: The ATG communication channel depends on the altitude, angle of elevation, and type of propagation environment [49]. The ATG communication links can be modeled by a probabilistic path loss model containing the line of sight (LoS) and non-LoS (NLoS) in different probabilities of occurrence. The LoS and NLoS path loss between UCH u and SMUE i expressions are similar to [50]. The transmission rate of SMUE i associated with u UCH at slice z is expressed as;

$$\iota_{iu}(t) = \frac{B}{I} log_2\big(1 + \frac{P_i}{\sigma^2 10^{\bar{\mathrm{L}}_{iu}(t)}}\big), \forall i \in \mathcal{I}, \forall u \in \mathcal{U}, \tag{4}$$

where $\bar{\mathrm{L}}_{iu}(t)$, B, P_i, and σ^2 denote the average path loss between UCH u and SMUE i [50], the channel bandwidth, the transmission power, and the noise power, respectively.

3) Communication Model: In the ATG network infrastructure, when the SMUEs offload their tasks to the MEC servers, the network will suffer from bandwidth inefficiency and communication costs. In this work, we consider three communication scenarios discussed as follows:

Scenario-1: When the ground BS $k \in \mathcal{K}$ is not overloaded/malfunctioning and the resource is available, the SMUE $i \in \mathcal{I}$ can obtain resources from MEC servers on the BSs through a wireless channel. Let us define $\alpha_{ikz}^{i \to k}(t) \in \{0, 1\}$ and $\xi_{ikz}(t) \in \{0, 1\}$ as a computation offloading variable that indicates the SMUE either computes tasks locally or offloads to edge computing node k and the connection status between SMUE i and BS k, respectively. Then, spectrum efficiency for SMUE i will be expressed as:

$$\Upsilon_{ikz}(t) = log_2\big(1 + \frac{\rho_i |G_{ikz}|^2}{\sigma_i^2}\big), \forall i \in \mathcal{I}, k \in \mathcal{K}, \tag{5}$$

where ρ_i, $|G_{ikz}|^2$, σ_i^2 denote the transmission power of SMUE i, channel gain between SMUE i and computation node k, and the power of the Gaussian noise at SMUE i, respectively. Therefore, the data rate for SMUE i is calculated as:

$$R_{ik}(t) = \xi_{ikz}(t) \alpha_{ikz}^{i \to k}(t) s_{ik} B \Upsilon_{ikz}(t), \forall i \in \mathcal{I}, k \in \mathcal{K}, \tag{6}$$

where $s_{ik} \in [0, 1]$ is a fraction of radio resource allocated to SMUE i by BS k. When the spectrum efficiency satisfies the requested demand, SMUE i can offload tasks to the MEC server on BS. The task transmission delay from SMUE i to BS k depends on the data rate of SMUE i, which is expressed as:

$$T_{ik}^{off}(t) = \frac{\alpha_{ikz}^{i \to k}(t) D_i}{R_{ik}(t)}, \forall i \in \mathcal{I}k, \tag{7}$$

where $\mathcal{I}k$ is a set of SMUE served by BS k and D_i is the input data size.

Scenario-2: In scenario-1, the SMUE i offloads tasks to BS k in slice z. However, in this scenario, when the BS k in slice z is overloaded, has limited coverage, or malfunctions, the SMUE i offloads tasks to the UCH u which are then computed by the MEC server on the UCH u. We define $\alpha_{iuz}^{i \to u}(t) \in \{0, 1\}$ as a decision variable that determines whether the SMUE i offloads computation task Λ_i to computational node UCH u or computes it locally.

$$\alpha_{iuz}^{i \to u}(t) = \begin{cases} 1, & \text{SMUE } i \text{ offloads } \Lambda_i \text{ to UCH } u \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Therefore, the offloading delay between SMUE i and UCH u is expressed as:

$$T_{iu}^{off}(t) = \frac{\sum_{i \in \mathcal{I}_x} \alpha_{iuz}^{i \to u}(t) D_i}{R_{iu}(t)}, \forall i \in \mathcal{I}, \forall u \in \mathcal{U}, \tag{9}$$

where $R_{iu}(t)$ is the data link capacity between SMUE i and UCH u.

Scenario-3: When the UCH's resources are insufficient to perform the offloaded task, it relays the request to the central controller/SDN via a wireless backhaul link. We define

$\alpha_{iuz}^{u\to SDN}$ as a decision variable that implies that the task is computed on UCH u or relayed to SDN.

$$\alpha_{iuz}^{u\to SDN}(t)=\begin{cases}1, & \text{if } \Lambda_i \text{ of SMUE } i \text{ is relayed}\\ & \text{from UCH } u \text{ to SDN}\\ 0, & \text{otherwise}\end{cases} \tag{10}$$

Therefore, the offloading delay between UCH u and SDN is expressed as:

$$T_{u,SDN}^{off}=\frac{\sum_{i\in\mathcal{I}_x}\alpha_{iuz}^{u\to SDN}(t)D_{iu}}{R_{u,SDN}},\forall i\in\mathcal{I},\forall u\in\mathcal{N}, \tag{11}$$

where $R_{u,SDN}$ is data link capacity between UCH u and SDN.

4) Computation Model: In our scenario, we consider that each SMUE i has a computation task $\Lambda_i(t)\equiv(D_i,C_i,\Gamma_i)$, where D_i is the size of input data including program coding and other parameters, C_i denotes total CPU cycle required to execute the task Λ_i, and Γ_i represents the maximum time delay of the task Λ_i.

A) Local Computation at SMUE: In the local computational mode, the task of SMUE i is executed locally at time slot t. The local execution delay is expressed as,

$$T_{iz}(t)=\frac{D_i}{f_{iz}}, \tag{12}$$

where f_{iz} denotes the computation capacity of SMUE i. To complete the tasks of SMUE i depends on $T_{iz}(t)\le\Gamma_i$. The local energy consumption of task Λ_i is expressed as;

$$P_{iz}(t)=\kappa_i(f_i)^{\upsilon}T_{iz}(t), \tag{13}$$

where $\kappa_i\ge 0$ denotes CPU switch capacitance of SMUE i and υ denotes constant value. In this paper, we set $\upsilon=3$ and $\kappa_i=10^{-27}$.

Although, when $T_{iz}>\Gamma_i$, $C_i>f_{iz}$, or $P_{iz}>\bar{P}_{iz}$, the SMUE i does not have enough computing or power resources to accomplish the tasks within a given deadline. Here, $\bar{P}_{iz}$ denotes the power availability of SMUE i. To handle this issue, we then define $\beta_{iz}(t)\in\{0,1\}$ as the SMUE status indicator variable at time slot t, expressed as;

$$\beta_{iz}(t)=\begin{cases}1, \text{if} \quad T_{iz}(t)<\Gamma_i, C_i<f_{iz}, P_{iz}<\bar{P}_{iz}\\ 0, \text{otherwise.}\end{cases} \tag{14}$$

Based on eqn. (14), the local execution time delay of task Λ_i is $T_{iz}(t)$ if $\beta_{iz}(t)=1$, and $\alpha_{ikz}^{i\to k}(t)=0$ and $\alpha_{iuz}(t)=0$; the local execution delay is $T_{iz}(t)+\tau_{iz}$ when $\beta_{iz}(t)=0,\alpha_{ikz}^{i\to k}(t)$ and $\alpha_{iuz}(t)=0$, where τ_i denotes average waiting time of task Λ_i or the local execution time delay = 0 when $\beta_{iz}(t)=0$, $\alpha_{iuz}(t)$ or $\alpha_{iuz}(t)=1$.

Let us denote ω^t and ω^e as the weighted parameters of time and energy, respectively. The local computing cost of SMUE i to perform the task Λ_i is expressed as:

$$\mathcal{Z}_{iz}^{loc}=\omega^t T_{iz}(t)+\omega^e P_{iz}(t). \tag{15}$$

B) Offloading SMUE Tasks: Based on the decision variable $\alpha_{ikz}^{i\to k}(t)\in\{0,1\}$ the SMUE i decides whether or not to compute task Λ_i on BS k at time slot t. If $\alpha_{ikz}^{i\to k}(t)=1$, implies the SMUE i offloads task to BS k, otherwise $\alpha_{ikz}^{i\to k}(t)=0$. The computation resource allocation F_{ik} of BS k at time slot t can be expressed as:

$$F_{ik}(t)=F_k\frac{C_i}{\sum_{i\in\mathcal{I}_k}C_i},\forall i\in\mathcal{I}k,k\in\mathcal{K}. \tag{16}$$

Therefore, the allocated computation resources from the MEC server should satisfy the following expression.

$$\sum_{i\in\mathcal{I}_k}\alpha_{ikz}^{i\to k}(t)\beta_{iz}F_{ik}(t)\le F_k. \tag{17}$$

The transmission delay of SMUE i while processing task Λ_i at time slot t is expressed as: $T_{ik}^{tr}(t)=\frac{D_i}{R_{ik}(t)},\forall i\in\mathcal{I}$. The execution time delay of the offloaded task is given as: $T_{ik}^{ex}(t)=\frac{C_iD_i}{F_{ik}(t)},\forall i\in\mathcal{I}$. The overall time delay of task Λ_i that execute at MEC server k is expressed as; $T_{ik}^{tot}=T_{ik}^{tr}(t)+T_{ik}^{ex}(t)$. The SMUE i consumes $P_{ik}^{tr}(t)$ energy to offload the tasks to the BS k at time slot t, which is expressed as: $P_{ik}^{tr}(t)=P_{ik}T_{ik}^{tr}(t)$, where P_{ik} denotes the transmission power of SMUE i. The execution energy consumption of offloaded task on BS k at time slot t is given as: $P_{ik}(t)=\kappa_k(f_{ik}^2)$, where $\kappa_k\ge 0$ denotes CPU switch capacitance of MEC server k. However, if $C_i>F_{ik}$ or $T_{ik}^{tot}(t)>\Gamma_i$, the BS k is overloaded. Therefore, the BS k offloads tasks to UCH u[1]/SDN or the SMUE i associated with the UCH u at time slot t.

When $\xi_{iuz}(t)=1,\forall u\in\mathcal{J}$, the SMUE i offloads task Λ_i to UCH u at time slot t. In addition, let $\upsilon_u(t)\in\{0,1\}$ indicate the resources of UCH u at the time t, where $\upsilon_u(t)=1$ has sufficient resources and can provide different services to SMUEs; otherwise, $\upsilon_u(t)$=0. We define $\alpha_{iuz}^{i\to u}\in\{0,1\}$ offloading decision variables whether the SMUE i offloads tasks to UCH u at time slot t, expressed as;

$$\alpha_{iuz}^{i\to u}(t)=\begin{cases}1, & \text{SMUE } i \text{ offloads } \Lambda_i \text{ to UCH } u\\ 0, & \text{Otherwise.}\end{cases} \tag{18}$$

The computation resource allocation F_{iu} at UCH u is expressed as;

$$F_{iu}=F_u\frac{C_i}{\sum\limits_{i\in\mathcal{I}_u}C_i},\forall i\in\mathcal{I},u\in\mathcal{U}, \tag{19}$$

where F_u represents the computation capacity of UCH u. The resources on UCH u are the overall resources of each UAV in the cluster. The computation capacity of UCH is expressed as $F_u=\sum\limits_{v\in V}f_v$. Therefore, the total computation resource must satisfy the following expression;

$$\sum_{i\in\mathcal{I}_u}\alpha_{iuz}(t)F_{iu}(t)\alpha_{iuz}^{i\to u}(t)\le F_u^{max}. \tag{20}$$

The transmission time delay of SMUE i when it offloads task Λ_i to UCH u at time slot t is expressed as; $T_{iu}^{tr}=\frac{\alpha_{iuz}D_i}{R_{iu}(t)},\forall i\in\mathcal{I}$. The offloaded task Λ_i execution time delay is expressed as; $T_{iu}^{ex}(t)=\alpha_{iuz}(t)\frac{C_iD_i}{F_{iu}}$. The total time delay to accomplish the offloaded task Λ_i at UCH u is expressed as: $T_{iu}^{tot}(t)=T_{iu}^{tr}(t)+T_{iu}^{ex}(t)$.

[1]Due to space limitation we focused on SMUE i offloads to UCH u, then task compute on UCH or SDN.

However, $F_{iu} < C_i | T_{iu}^{tot} > \Gamma_i, \upsilon_u(t) = 0$, the UCH u relays the offloaded task to the SDN. The total computational time delay of offloaded task Λ_i from SMUE i to UCH u and relays to SDN, which is expressed as:

$$T_{iu}^{SDN}(t) = T_{iu}^{tr} + T_u^{SDN} + T_{iu,SDN}^{ex}. \tag{21}$$

Task Λ_i of SMUE i will be executed either locally or offloaded to the nearest computational node (either the UCH u or BS k) but will be executed at only one location at a time slot t. The union set of the UCHs and BSs are defined as computational nodes, which is defined as $\hat{j} \in \hat{\mathcal{J}} = \{1, 2, \ldots, \hat{J}\}$ and $\hat{j} \in \{k \cup u\}$. The association of SMUE i with the computational node $\hat{j}$ should fulfill the following;

$$\left(1 - \alpha_{i\hat{j}z}(t) + \alpha_{i\hat{j}z}\left(\alpha_{i\hat{j}z}^{i\to k} + \alpha_{i\hat{j}z}^{k\to u} + \alpha_{i\hat{j}z}^{i\to u} + \alpha_{i\hat{j}z}^{u\to SDN}\right)\right) = 1, \tag{22}$$

5) Energy Model: In this model, we address the energy consumption cost [2] incurred when the SMUE i offloads a task to the MEC servers in different phases, such as transmission and execution during UCH u flight and hovering. Initially, we assume that each SMUE i and UCH u utilizes discrete transmit power control [50]. The power selection and offloading tasks depend on SMUEs and the computational node's power-level decision variables. When the transmission power $p_{iw}^T(t) = 0$, where $w \in \mathcal{W} = \{1, \ldots, W\}$ transmission power level, the SMUE decides to compute the input task locally, the energy consumption during local computing is calculated as $P_i^{exe}(t) = \kappa_i (f_i(t))^2, i \in \mathcal{I}, \kappa_i > 0$. Otherwise, the SMUE i decides to offload the task onto computational node $\hat{j}$. The transmission power $p_{iw}^T(t) = 1$ according to the policies, and the energy consumption is expressed as;

$$P_{i\hat{j}z}^{tr}(t) = \sum_{\hat{j}=1}^{\hat{J}} \alpha_{i\hat{j}z}(t) p_{i\hat{j}z}(t), i \in \mathcal{I}. \tag{23}$$

The power constraint of SMUE i is expressed as $\alpha_{i\hat{j}z}(t)(P_{i\hat{j}}^{exe}(t) + P_{i\hat{j}z}^{tr}(t)) \leq P_{i\hat{j}}^{max}, i \in \mathcal{I}$. Based on the above-mentioned cases, we can evaluate the energy consumption of transmission and execution expenses from SMUE i to the computational node. Assume that the SMUE is selected or associated with the UCH u based on the current policy and other SMUE-related information. In this instance, the energy consumption is also estimated based on UCH u limitations at the time slot t, including flying, hovering, and execution energy consumption. The transmission energy consumption for computation task offloading on UCH u is calculated as follows: $P_{iu}^{tr}(t) = P_i^{tr}(t).T_i^{tr}(t)$. The UCH u energy consumption after receiving the offloaded task from SMUE i at time slot t is expressed as: $P_{iu}^{exe}(t) = \sum_{i\in\mathcal{I}} \alpha_{i\hat{j}z}(t) \kappa_u (f_u(t))^2$.

Therefore, the total energy consumption of UCH u in the time slot t is calculated as:

$$P_u^{tot}(t) = \rho_1 P_u^{fl}(t) + \rho_2 P_u^{ho}(t) + \rho_3 P_u^{exe}(t), \tag{24}$$

[2]Because BS energy is not as crucial as UAV energy, we focus more on UAV energy resources than BS energy.

where ρ_1, ρ_2, ρ_3 are the variable parameters for the above UCH u energy consumption, and P_u^{fl}, P_u^{hov} represent UCH flying and hovering energy consumption, respectively.

To handle UCH fairness coverage, we specify the level of fairness between UCHs that serve SMUEs and SMUEs. Let us define the fairness level $\Upsilon_u(t) \in [0, 1]$ among UCHs as;

$$\Upsilon_u(t) = \frac{(\sum_{u=1}^{U} \sum_{t=1}^{T} \beta^u(t))^2}{\sum_{u=1}^{U} (\sum_{t=1}^{T} \beta^u(t))^2}, \tag{25}$$

where $\beta^u(t) = \frac{\sum_{i=1}^{I} \alpha_{i\hat{j}z}(t)}{I}, \beta^u(t) \in [0, 1]$ indicates the relative number of SMUEs served by UCH u at time slot t. For various reasons, not all SMUEs may be accessible to UCHs within a given time frame. To manage this circumstance, we determine the level of fairness among SMUEs $\bar{\Upsilon}_i(t) \in [0, 1]$ as:

$$\bar{\Upsilon}_i(t) = \frac{(\sum_{i=1}^{I} \sum_{t=1}^{T} \alpha_{i\hat{j}z}(t))^2}{\sum_{i=1}^{I} (\sum_{t=1}^{T} \alpha_{i\hat{j}z}(t))^2}. \tag{26}$$

Lastly, we define computation cost in terms of energy consumption and time delay of SMUE i at time slot t as $\mathbb{C}_{i\hat{j}}(t) = \bar{\chi}_{i\hat{j}}(t) + \chi_{i\hat{j}}(t)$, where $\bar{\chi}_{i\hat{j}}(t) = T_{i\hat{j}}(t) + P_{i\hat{j}}(t)$ and $\chi_{i\hat{j}}(t) = \omega^t T_{i\hat{j}}^{tot}(t) + \omega^e P_{i\hat{j}}^{tot}(t)$ represents the computation cost of SMUE i for computing task Λ_i locally and/or at the MEC server at time slot t.

B. Blockchain-based Virtualized Resource Allocation

In this part, we discuss a blockchain-based resource allocation framework to define the interactions between InPs, VNOs, and SMUEs. Since the majority of SMUEs involved in resource allocation communicate wirelessly and are untrusted nodes, ensuring the security of resource allocation transactions will be challenging. To build trust between participating nodes and prevent malicious node actions, we used a consortium blockchain, in which blockchain system entities such as InPs, VNOs, SMUEs, and consensus nodes have a blockchain account that allows them to join the resource allocation system and perform various actions. The edge servers owned by VNO (UVNO or MVNO) and the SDN controller process the complete blockchain network. The VNOs are in charge of encoding the transaction and resource requests, signing transactions, and connecting to the blockchain process. The VNOs also get incentives depending on the number of resources provided to SMUE. The main entities that interact in blockchain-based resource allocation are presented as follows:

1) Certificate Authority (CA): It initializes the resource allocation system, generates public parameters, and manages the operating entities.

2) SMUEs: The SMUEs send requests to the blockchain-enabled resource allocation system for power, computation, and radio resources to offload tasks.

3) MVNO: lease the resources of BSs in different slices from the InPs and sublease them to SMUEs connected to terrestrial infrastructure.

4) UVNO: Leases resources of UCHs from InPs and provides different resources to SMUEs when the MVNOs are overloaded or malfunctioning. Each UVNO/MVNO leases resources from InPs at a minimum price and sells them to

SMUEs to maximize their utility. For the sake of simplicity, we define UVNOs and MVNOs as VNOs. Therefore, InPs and VNOs continually adjust their pricing strategies to maximize their utilities and rewards while meeting the QoS requirements of SMUEs. The SDN controller manages the available resources of VNOs and offloads demands from SMUEs. The SDN controller tracks all resource allocation and offloading transactions through smart contracts (SC) and a control strategy. The SC is stored on a blockchain-based platform and automatically executes all or parts of the agreement between the resource provider and the requester. The SC ensures that the agreements reached between the entities involved in the resource allocation system are followed. The transaction coins work as VNOs' and SMUEs' digital assets used to purchase resources from resource providers. Each operator has a virtual wallet to manage individual resources such as digital coins.

In our scenario, when a resource request is sent to the blockchain system, its validity is verified via a distributed consensus process. The transaction confirmation process in our proposed blockchain-enabled resource allocation framework involves three phases.

1) Miner selection: In the consortium blockchain, only selected nodes can verify and validate transactions. In the ATG system, not all edge nodes are trusted; some malicious edge nodes may discard transaction records during their mining process. We implement a hybrid consensus scheme combining the advantages of practical byzantine fault tolerance and proof of reputation [51]. It selects nodes with a high reputation as "active consensus nodes" to ensure a reliable consensus process. The node with the highest reputation value acts as the leader, collects transactions from the transaction pool, builds a new block, and broadcasts it to the validator to verify its validity.

2) Consensus process: All consensus nodes audit the block broadcasted by the leader and exchange their results for mutual supervision. Each consensus node compares its audit results to others, then signs and sends its audit result to the leader along with its decision.

3) Blockchain update: The leader collects the responses from consensus nodes and analyzes them. If the majority of the consensus nodes agree that the block is valid, the leader appends the block to the chain and sends updates to all nodes. Otherwise, the block will be discarded. After that, the resource provider allocates resources to the requester, and the requester transfers digital coins to the wallet of the resource provider. The operation of the consortium blockchain in our proposed resource allocation framework is presented in Algorithm 1.

IV. Utility function

In this paper, we aim to maintain an optimal resource allocation system where InPs and VNOs maximize their utility/revenue while the SMUEs ensure their QoS requirements with optimal cost. Our framework leverages DRL, blockchain, and game theory to maintain secure transactions between resource providers and requestors while allowing them to make intelligent decisions based on optimal strategies. In our scenario, VNOs purchase resources (radio, computation, and transmission power) from InPs and allocate them to SMUEs to maximize their utilities. We define $\Phi_{jn} = \{\bar{\psi}, \bar{\phi}, \bar{\eta}\}$ to denote the total resources price, where $\bar{\psi}, \bar{\phi}$, and $\bar{\eta}$ are a radio, computation, and transmission power leasing price set by InPs to VNOs, respectively. We calculate the utility function of InPs $U_n(t)$, which is given as:

$$U_n(t) = \sum_{j=1}^{J} \left(\bar{\psi} S_{jn} B + \bar{\phi} F_{jn} + \bar{\eta} P_{jn}\right) - \Psi_n, \tag{27}$$

where Ψ_n denotes the power consumption, B denotes system bandwidth and related costs of InP n while leasing resources to VNO j at time slot t. And S_{jn}, F_{jn}, and P_{jn} present the radio, computation, and transmission power resources that are allocated/sold by InP n to VNO j, respectively. The VNO j determines its resource demands $\mathbb{R}_{jn} = \{S_{jn}, F_{jn}, P_{jn}\}$ based on the price set by InP n and the requested resources are allocated to the VNO j once both parties agree. Then, we calculate the utility of VNO j, the total of the utilities derived from the radio, computation, and transmission power resources. Let $\Phi_{ij} = \{\psi, \phi, \eta\}$ denote the prices that SMUEs charged by VNOs j, where ψ, ϕ, and η are the prices for each unit of the radio to transmit the input of an SMUE i to a computational node $\hat{j}$, computation resource, and transmission power, respectively. Let $\hat{\mathbb{R}}_{ij} = \{s_{ij}, \varrho_{ij}, e_{ij}\}$ denote the SMUE i resource fraction demands, where s_{ij}, ϱ_{ij}, and e_{ij} are the radio, computation, and transmission power resource fractions, respectively. The total utility of VNO j that allocates the radio resource to SMUE i at time slot t is expressed as:

$$\mu_{ij}(t) = \xi_{i\hat{j}}(t) s_{ij}(t) \psi B \Upsilon_{ij}(t) - S_{jn}(t) \bar{\psi} B, \tag{28}$$

where B is system bandwidth.

Next, we define a fraction/percentage of computation resources of VNO j allocated to SMUE i in slice z as $\varrho_{ij} \in [0, 1]$, hence $\sum_{z\in\mathcal{Z}} \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{I}_{jz}} \varrho_{ij} \leq 1$. The total utility of VNO j that allocates computation resource to SMUE i at time slot t is expressed as:

$$\vartheta_{ij}(t) = \phi\left(\varrho_{ij} \frac{F_j^{max}}{D_i} - \frac{f_i^{loc}}{D_i}\right) - \bar{\phi} F_{jn}. \tag{29}$$

In Equ. (29), the reciprocal function of $\frac{F_j^{max}}{D_i}$ is execution time consumption of offloaded intensive computation task and reciprocal of $\frac{F_i^{loc}}{D_i}$ is local task execution time.

Lastly, we calculate the utilities of VNO j for allocating transmission power to SMUE i. Let $e_{ij} \in [0, 1]$ define a fraction/percentage of transmission power allocation to SMUE i. The total utility of VNO for allocating transmission power is expressed as:

$$\theta_{ij} = \eta(e_{ij} P_{ij} - P_i^{loc}) - \bar{\eta} P_{jn}. \tag{30}$$

Therefore, based on the above three equations, we calculate the overall utility of VNO j at time slot t as:

$$U_j(t) = \sum_{z\in\mathcal{Z}} \sum_{i\in\mathcal{I}_{jz}} \alpha_{i\hat{j}z} \Big(\mu_{ij}(t) + \vartheta_{ij}(t) + \theta_{ij}(t)\Big) - \bar{\Psi}_j, \tag{31}$$

where $\bar{\Psi}_j$ denotes the power consumption and related costs of VNO j while leasing resources to SMUE i at time slot t. Furthermore, we calculate the utility of SMUE i as:

$$U_i(t) = \varpi_{ijz}(t) - \big(\mu_{ij}(t) + \vartheta_{ij}(t) + \theta_{ij}(t) + \mathbb{C}_{i\hat{j}}(t)\big), \quad (32)$$

where $\varpi_{ijz}(t) \in [0,1]$ is QoS satisfaction level of SMUEs in the network slice z. The overall system utility is expressed as:

$$U(t) = \sum_{n=1}^{N} U_n(t) + \sum_{j=1}^{J} U_j(t) + \sum_{i=1}^{I} U_i(t) \quad (33)$$

V. PROBLEM FORMULATION

A. Optimization Problem

In our scenario, we defined multiple variables for different scenarios. For the sake of simplicity we redefined decision variables as $\Pi(t) = \{\alpha_{ijz}, \alpha_{i\hat{j}z}^{i\to k}, \alpha_{i\hat{j}z}^{k\to u}, \alpha_{i\hat{j}z}^{i\to u}, \alpha_{i\hat{j}z}^{u\to SDN}\}$ and $\Pi \in \{0,1\}$. Based on the overall system utility function in eqn. (33), we formulate optimization problems as follows.

$$\boldsymbol{P_1} \quad \max_{\mathbf{\Pi},\mathbb{R},\mathbf{\Phi}} U(t) \quad (34a)$$

$$\boldsymbol{s.t} \quad C_1 : \Pi(t), \xi_{i\hat{j}z}(t), \beta_{iz}(t) \in \{0,1\} \quad (34b)$$

$$C_2 : (22) \quad (34c)$$

$$C_3 : \sum_{i\in\mathcal{I}_{jz}} \Pi(t)\xi_{i\hat{j}z}(t)s_{ij}(t) \le 1, \forall j \quad (34d)$$

$$C_4 : \sum_{i\in\mathcal{I}_{jz}} \Pi(t)\varrho_{ij} \le 1, \forall j \quad (34e)$$

$$C_5 : \sum_{i\in\mathcal{I}_{j}} \Pi(t)e_{ij} \le 1, \forall j \quad (34f)$$

$$C_6 : \Upsilon_u(t) \in [0,1], u \in \mathcal{U} \quad (34g)$$

$$C_7 : \varphi_{i\hat{j}z} \ge \varphi_{thr}, \forall i\hat{j} \in \mathcal{I}\hat{j}, \quad (34h)$$

where $\mathbf{\Pi}$, $\mathbb{R} \in \{\mathbb{R}_{jn}, \hat{\mathbb{R}}_{ij}\}$ and $\mathbf{\Phi} \in \{\Phi_{ij}.\Phi_{jn}\}$ denote the set of computation offloading decision variables in different scenarios, set of resources demands in different layers (i.e. resource demands of VNOs and SMUEs), and set of unit prices of different allocated resources (charged by InP n and VNO j), respectively. Constraint C_1 represents offloading decision variable, connection status variable between SMUE i and VNO j, SMUE i resource indicators at time slot t, respectively. Constraint C_2 indicates that the task of SMUE i can only be computed at one computational node at time slot t. The constraints C_3 indicate that the total allocated radio cannot exceed the available radio. C_4 denotes that the computation resources allocated to SMUE cannot exceed the maximum computing capacity of VNO, and C_5 represents the transmission power assigned to SMUE cannot exceed VNO's maximum power level.Constraint C_6 represents the fairness coverage status of the UCH u. Constraint C_7 specifies that the SINR between the SMUE i and the adjacent computational node $\hat{j}$ must be greater than or equal to the minimum threshold. As we know, the SMUEs/IoT devices are increasing exponentially in the 5G/B5G networks with dynamic network topology. The complexity of resource allocation, computation offloading, and association optimization problems, i.e., $\boldsymbol{P_1}$, is high. This problem is a mixed-integer and non-convex optimization problem due to binary decision vectors ($\mathbf{\Pi}$), a continuous indicator of resources with price ($\mathbf{\Phi}, \mathbb{R}$), and the objective function is non-convex. Then it is NP-hard and difficult to obtain an optimal solution. Some existing works try to handle such problems by decomposing the optimization problem into two sub-problems [52], even though solving optimization problems in ATG networks similar to [52]–[54] results in a curse of dimensionality and increased security issues. Additionally, utilizing various resources in virtualized hierarchical ATG network is limited in recent works.

B. Hierarchical Stackelberg Game Formulation

In various optimization problems, game theory has been examined to be an essential method for studying decentralized decision-making among strategic players [55]. In a Stackelberg game model, players are in the form of leaders and followers. In a Stackelberg game, the leaders can set resource pricing and continuously modify them based on their strategies. On the other hand, it enables the followers to choose their resource demands based on the prices set by the leaders and adjust their demands based on their strategies. In this way, both the leaders and followers achieve the optimal point that benefits them the most. Therefore, in our scenario, we model the interactions between the InPs, VNOs, and SMUEs as a three-stage MLMF Stackelberg game-based model.

In the upper layer of the proposed hierarchical game, the InPs interact with VNOs to lease their resources to maximize their utility. The InPs first determine the price per unit of resources based on the demand of the VNOs, and the VNOs act as followers and move based on the price set by the InPs. In the lower layer of the game, the VNOs interact with SMUEs to sublease the resources they owned to the SMUEs. The VNOs aim to maximize their utilities while the SMUEs purchase resources to compute tasks and ensure their QoS requirements with optimal cost. At this stage, the VNOs act as a leader and determine resource prices first, and then the SMUEs decide their demand according to the price that the VNOs set. To maintain the security of resource allocation among untrusted resource providers and requestors, the transactions relating to resource allocation and pricing are recorded in a blockchain digital ledger. When the SMUEs perform different operations such as computing, and communication in diverse network slices, they interact with VNOs to request resources. The communication process between InPs, VNOs, and SMUEs is shown in Fig. 2, and discusses their interaction in the blockchain-based 5G-RAN slicing.

Stage-I: Resource allocation and pricing model of InPs

At stage I, the InPs play the role of a leader, offering resources such as computational, transmission power, and radio resources to VNOs in various network slices. In addition, the InPs can set the unit price of each resource, and the VNOs are charged according to their own demands. According to eqn. (27), we formulate the sub-game optimization problem at this stage, which maximizes the revenue of InPs and explores the

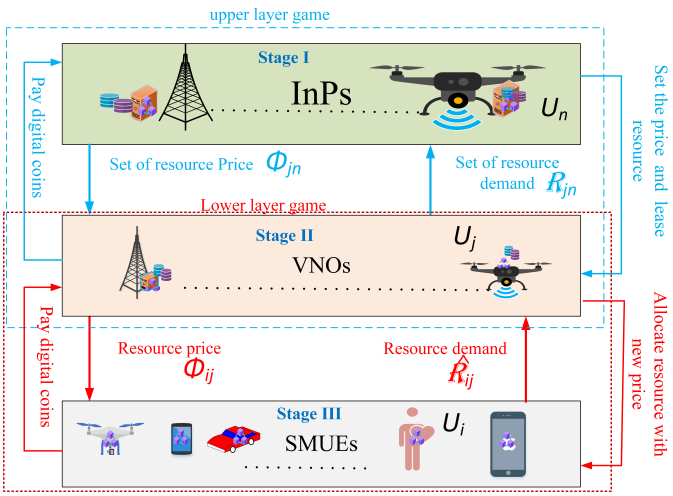


Fig. 2: Three stages Stackelberg game.

optimal price per unit of resources.

$$\boldsymbol{P_2} \quad \max_{\mathbb{R}_{jn}>0,\Phi_{jn}>0} U_n(t), \quad s.t: \sum_{j=1}^{J} \mathbb{R}_{jn} \leq \mathbb{R}_n, \tag{35}$$

where $\mathbb{R}_{jn}$ denotes set of resources at InP n leased to VNO j, including computation, transmission power, and bandwidth; Φ_{jn} denotes unit price of resources set by InP n and $\mathbb{R}_n$ denote the available resources of InP n.

Stage-II: VNOs Resource allocation model to SMUEs

At this stage, the VNOs are followers, can lease the resources from InPs, and sublease them to SMUEs in different network slices. We assume the VNOs of both slices jointly receive signals from SMUEs in a cooperative manner. The SMUEs offload tasks and, are handled by VNOs, allocate resources based on their demands. Therefore, the main objective of stage II is to maximize the utility of VNOs while satisfying the requirements of the SMUEs. Thus, we formulate the sub-game optimization problem based on eqn.(31) as follows.

$$\boldsymbol{P_3} \quad \max_{\hat{\mathbb{R}}_{ij}>0,\Phi_{ij}>0} U_j(t) \quad s.t: \sum_{i=1}^{I} \hat{\mathbb{R}}_{ij} \leq \mathbb{R}_{jn}. \tag{36}$$

Stage-III: SMUEs communication model, computation mode

In this stage, the SMUEs are followers and the VNOs are leaders. The SMUEs can determine the resources demands to offload tasks to VNOs through allocated access bandwidth. In stage III, the SMUEs can select the appropriate resource providers in a given network slice based on the optimal price, the available resources, and the QoS levels. In our scenario, SMUEs aim to get the optimal price for required resources, optimal offloading strategy, minimize costs, and ensure QoS. The objective of stage III is to improve the utility of SMUEs.

According to (32), the sub-game optimization problem of stage-III is expressed as;

$$\boldsymbol{P_4} = \max_{\Pi_{ijz},\xi_{ijz},\beta_{iz}} U_i(t) \quad s.t: \varpi_{ijz}(t) > 0 \tag{37}$$

Algorithm 1 Blockchain-empowered resource allocation

Initialize: $\mathcal{I},\mathcal{J},\mathcal{N},\mathcal{U},\mathcal{K}$ and NSB
Registration and certification of $\mathcal{I},\mathcal{J},\mathcal{N},\mathcal{U},\mathcal{K}$ and NSB
for time T **do**
  for all $\mathcal{I},\mathcal{J},\mathcal{N},\mathcal{U},\mathcal{K}$ and NSB **do**
    Verify the certificates
    if the certificates of $\mathcal{I},\mathcal{J},\mathcal{N},\mathcal{U},\mathcal{K}$ and NSB are verified
      Set-up three-stage Stackelberg game as in eqn. (38), (40) and (42)
      Run SC for resource allocation to create a block
    else
      Terminate SC
    end if
  end for
  for all validator nodes **do**
    The leader broadcasts block blk
    Each node receive blk and check its validity
    if Block blk is verified and legal **then**
      Set the block blk = 1
    else
    Set the block blk =0
    end if
    Send the audit results to the leader for analysis
    The leader accepts the audit results and adds blk to the chain if the majority of the nodes confirmed that it is valid; otherwise, discard blk
  end for
end for

C. Stackelberg Game Analysis

In the formulated hierarchical game, the strategy of each stage affects the strategies in the other stages. Thus, we use the backward induction method [56], [57] to analyze the proposed game model $\boldsymbol{P_2}, \boldsymbol{P_3}, \boldsymbol{P_4}$ or obtain Stackelberg equilibrium (SE) of the game. The Backward induction method can capture the sequential dependence of the decisions in the stages of the game. First, solve the problem at stage-III ($\boldsymbol{P_4}$), then stage-II ($\boldsymbol{P_3}$), and finally, at stage-I ($\boldsymbol{P_2}$).

With a given leader's strategy, it is possible to guarantee the existence of SE in the formulated hierarchical game, if the follower's best-response strategy set is a singleton.

Let's define game players in our scenario:

Definition 1: *The formulated InP-VNO-SMUE hierarchical game consists: Player set:* $\{N\} \cup \{J\} \cup \{I\}$, *which includes the InPs, VNOs, and SMUEs;*

Strategy set: $\{\boldsymbol{\Phi_{jn}}, \boldsymbol{\mathbb{R}_{jn}}, \boldsymbol{\Phi_{ij}}, \boldsymbol{\hat{\mathbb{R}}_{ij}}, \boldsymbol{\Pi}\}$, *where* Φ_{jn} *is the unit price of resources set on InPs,* Φ_{ij} *is charged unit prices of a set of resources from SMUEs to VNOs and* $\mathbb{R}_{jn}$ *denotes the*

leased resources on VNOs, depending on resource demands of SMUEs $\hat{\mathbb{R}}_{ij}$. Then, the strategy of InP n for unit price of resources is set $\{\boldsymbol{\Phi_{jn}} \in [\Phi_{jn}^{min}, \Phi_{jn}^{max}]\}$, the VNO j's strategies of resources and subleasing price are sets $\{\boldsymbol{\mathbb{R}_{jn}} \in [\mathbb{R}_{jn}^{min}, \mathbb{R}_{jn}^{max}]\}$ and $\{\boldsymbol{\Phi_{ij}} \in [\Phi_{ij}^{min}, \Phi_{ij}^{max}]\}$, the strategy of SMUE resource demands is set $\{\hat{\boldsymbol{\mathbb{R}}}_{\boldsymbol{ij}} \in [\hat{\mathbb{R}}_{ij}^{min}, \hat{\mathbb{R}}_{ij}^{max}]\}$ and the strategy of decision is set $\{\boldsymbol{\Pi} \in \{0,1\}\}$. Therefore, based on the utility function sets: $\{U_n, U_j, U_i\}$, we define Stackelberg equilibrium (SE) [58] as follows:

Definition 2: *To maximize the utility of players, in stage III, the players/SMUEs should make a best-decision (optimal decision), i.e., mainly the computation offloading while considering connection status (channel/links) and its resource constraints, and adjust their individual decision strategy and demands depending on the stage II decision while considering the computation costs.*

Within this game, given the optimal strategies of SMUEs i.e ($\hat{\mathbb{R}}_{ij}^*$ the set of optimal resource demands, Π^* set of optimal decisions). The SE exists at the lower layer of the game if the following conditions are met.

$$U_i\big(\Pi^*, \Phi_{ij}^*, \hat{\mathbb{R}}_{ij}^*\big) \geq U_i\big(\Pi, \Phi_{ij}^*, \hat{\mathbb{R}}_{ij}^*\big), \tag{38}$$

where Φ_{ij}^* is the optimal prices of resources set by VNO j.

Theorem 1: *Given the strategy of SMUEs to maximize their utility while minimizing computation costs, depending on the optimal decision (Π_{ijz}^*) and optimal resource demands ($\hat{\mathbb{R}}_{ij}^*$). We provide the second-order derivative of (32) to demonstrate the existence of SE and the uniqueness of the optimal solution of (P4).*

Proof 1:

$$\frac{\partial^2 U_i(t)}{\partial(\Pi, \hat{\mathbb{R}}_{ij})^2} \leq 0$$

$$\frac{\partial^2 U_i(t)}{\partial(\Pi)^2} = \frac{\partial^2 U_i(t)}{\partial(\Pi)^2\big(\Pi\varpi_{ijz} - \Pi\Phi_{ij}\hat{\mathbb{R}}_{ij} + \Pi\mathbb{C}i\hat{j}(t)\big)} \leq 0$$

$$= -\frac{2U_i(t)\big(\varpi_{ijz} + \Phi_{ij}\hat{\mathbb{R}}_{ij} + \mathbb{C}i\hat{j}(t)\big)^2}{\big(\Pi\varpi_{ijz} + \Pi\Phi_{ij}\hat{\mathbb{R}}_{ij} + \Pi\mathbb{C}i\hat{j}(t)\big)^3} \leq 0 \quad \text{and}$$

$$\frac{\partial^2 U_i(t)}{\partial(\hat{\mathbb{R}}_{ij})^2} = \frac{\partial^2 U_i(t)}{\partial(\Pi)^2\big(\Pi\varpi_{ijz} - \Pi\Phi_{ij}\hat{\mathbb{R}}_{ij} + \Pi\mathbb{C}i\hat{j}(t)\big)} \leq 0$$

$$= -\frac{U_i(t)(\Pi)^2(\Phi_{ij})^2}{\big(\Pi\varpi_{ijz} + \Pi\Phi_{ij}\hat{\mathbb{R}}_{ij} + \Pi\mathbb{C}i\hat{j}(t)\big)^3} \leq 0. \tag{39}$$

Therefore, U_i is strictly concave, and SE exists in this sub-game.

Definition 3: *At stage II, the VNOs can set the price of leased resources from InPs and choose optimal resource demands, but to set the price of resource fraction, it should not exceed the maximum price. The price was also set according to the SMUEs' requests and set the price of InPs.*

Within this game, given the strategies of VNOs i.e., the resource demands $\mathbb{R}_{jn}^*$ and the optimal set of price Φ_{ij}^*. The SE exits in the sub-game if the following conditions are met.

$$U_j\big(\Phi_{ij}^*, \mathbb{R}_{jn}^*, \hat{\mathbb{R}_{ij}^*}, \Phi_{jn}^*, \Pi^*\big) \geq U_j\big(\Phi_{ij}^*, \mathbb{R}_{jn}, \hat{\mathbb{R}_{ij}^*}, \Phi_{jn}, \Pi^*\big),$$
$$U_j\big(\Phi_{ij}^*, \mathbb{R}_{jn}^*, \hat{\mathbb{R}_{ij}^*}, \Phi_{jn}^*, \Pi^*\big) \geq U_j\big(\Phi_{ij}, \mathbb{R}_{jn}^*, \hat{\mathbb{R}_{ij}^*}, \Phi_{jn}^*, \Pi^*\big). \tag{40}$$

Theorem 2: *There exists an optimal resource demand selection $\mathbb{R}_{jn}^*$ of VNOs, an optimal resource price Φ_{jn}^* set by the InP n and an optimal resource price Φ_{ij}^* set by the VNO j to sublease resources to SMUE i, we prove the existence of SE in this sub-game using the second-order derivative of the utility function of $U_j(t)$ of eqn. (31) with respect to $\mathbb{R}_{jn}$ and Φ_{ij}, which is expressed as:*

Proof 2:

$$\frac{\partial^2 U_j(t)}{\partial(\mathbb{R}_{jn}, \Phi_{ij})^2} \leq 0$$

$$\frac{\partial^2 U_j(t)}{\partial(\Phi_{ij})^2} = \frac{\partial^2 U_j(t)}{\partial(\Phi_{ij})^2\big(\Pi\Phi_{ij}\hat{\mathbb{R}} - \Phi_{jn}\mathbb{R}_{jn} - \Psi_j\big)} \leq 0$$

$$= -\frac{2U_j(t)(\Pi)^2(\mathbb{R}_{jn})^2}{\big(\Pi\Phi_{ij}\hat{\mathbb{R}} + \Phi_{jn}\mathbb{R}_{jn} + \Psi_j\big)^3} \leq 0 \quad \text{and}$$

$$\frac{\partial^2 U_j(t)}{\partial(\mathbb{R}_{jn})^2} = \frac{\partial^2 U_j(t)}{\partial(\mathbb{R}_{jn})^2\big(\Pi\Phi_{ij}\hat{\mathbb{R}} - \Phi_{jn}\mathbb{R}_{jn} - \Psi_j\big)} \leq 0$$

$$= -\frac{2U_j(t)(\Phi_{jn}^2}{\big(\Pi\Phi_{ij}\hat{\mathbb{R}} + \Phi_{jn}\mathbb{R}_{jn} + \Psi_j\big)^3} \leq 0. \tag{41}$$

Therefore, U_j is strictly concave, and SE exists in this Stackelberg game.

Definition 4: *To maximize the utility of InPs, the leaders at stage-I make the best pricing decisions based on an optimal strategy and adjust their pricing strategies per unit of resources.*

Within this game, $\mathbb{R}_{jn}^*$ the optimal resource demand of VNOs and Φ_{jn}^* the optimal price set by the InP n, the point $(\mathbb{R}_{jn}^*, \Phi_{jn}^*)$ is the SE of the sub-game if it fulfilled the following expression;

$$U_n\big(\mathbb{R}_{jn}^*, \Phi_{jn}^*, \Phi_{ij}^*\big) \geq U_n\big(\mathbb{R}_{jn}^*, \Phi_{jn}, \Phi_{ij}^*\big). \tag{42}$$

Theorem 3: *There exists an optimal resource demand strategy R_{jn}^* of VNO j and optimal resource price Φ_{jn}^* set by InP n to lease resources to VNO j, and it controls the leasing price set by VNO Φ_{ij}^*; thus, we prove the existence of SE in this sub-game through the second-order derivative of eqn. (27) with respect to Φ_{jn}, which is expressed as:*

Proof 3:

$$\frac{\partial^2 U_n(t)}{\partial(\Phi_{jn})^2} \leq 0$$

$$= \frac{\partial^2 U_n(t)}{\partial(\Phi_{jn})^2(\Phi_{jn}\mathbb{R}_{jn} - \Psi_n)} \leq 0$$

$$= -\frac{2U_n(t)(\mathbb{R}_{jn})^2}{(\Phi_{jn}\mathbb{R}_{jn} + \Psi_n)^3} \leq 0. \tag{43}$$

Therefore, U_n is strictly concave, and SE exists in the upper layer of the game.

VI. MADRL-BASED UTILITY MAXIMIZATION AND PRICE OPTIMIZATION

The optimization problems formulated earlier are complex and large in state-action space (dimensionality curse problem) challenging to solve directly using a Stackelberg game. Therefore, we transform the problems into a stochastic

game (MDP game) and adopt a MADRL algorithm where a decision is made locally by each VNO and SMUE using a set of local resource demands. The MDP model is a potential game model for the RL technique and can handle sequential decision problems. In this game, players, such as InPs, VNOs, and SMUEs, act as agents that decide their actions based on observation and other agents' experience information. To address the dynamic ATG network conditions, the agents will make decisions at each set of time intervals. Agents obtain local observations based on their interactions with an unknown environment and execute actions in a distributed manner to enhance their utility.

The proposed scheme's main objective is to maximize resource providers' utility and satisfy resource requesters' requirements by allowing them to devise optimal strategies and policies while making decisions. We discretize the resource allocation problem into a series of time steps. At each time slot t, the SDN controls the ATG network, resource allocation, and price policy to make a decision and consider the state of the environment to maximize the utility function shown in (34).

In each stage of the stochastic game, each player (i, j, n) generates an observation $O_l(t)$, which is part of the state S_t at each time slot t, and gives its action $a_l(t)$ to the environment, then obtains a reward $r_l(t)$ from the environment. After executing the actions, the old state s_t changes into a new state s_{t+1}. Therefore, obtaining an optimal policy that maximizes the long-term reward depends on a set of state space and possible actions. State space, action space, and reward in our scheme are defined as follows.

State space: The state space is a description of the environment, denoted as $S = \{S_1, S_2, \ldots, S_l\}$, where S_l denotes the local state of players $L \in \{i, j, n\}$. The local states are defined as:

$$s(t) = \{\mathbb{R}_n, \mathbb{R}_j, \mathbb{R}_{jn}, \hat{\mathbb{R}}_{ij}, \Phi_{jn}, \Phi_{ij}, \xi_{i\hat{j}}, \varpi_{ijz}(t), \vartheta_{ijz}\}, \quad (44)$$

where $\mathbb{R}_n, \mathbb{R}_j, \mathbb{R}_{jn}, \hat{\mathbb{R}}_{ij}, \phi_{jn}, \xi_{ijz}, \varpi_{ijz}(t)$, and ϑ_{ijz} denote the available set of resources at InP n, available resources of VNO j, resource demands of VNOs, SMUEs' resource demands, the unit price per resources set by InP n and VNO j, connection status between a computational node with VNOs and SMUEs, QoS satisfaction of SMUE at time slot t, and trust value of nodes, respectively. Note: The InPs observe $\mathbb{R}_{jn}, \Phi_{jn}$, and $\mathbb{R}_{ij}$ from the previous state $s(t-1)$. The VNOs observe $\mathbb{R}_n, \Phi_{jn}$, and $\hat{\mathbb{R}}_{ij}$ from current state $s(t)$ and $\mathbb{R}_{jn}$ and Φ_{ij} from previous state $s(t-1)$. The SMUEs observe $\Phi_{ij}, \xi_{i\hat{j}}, \varpi_{ijz}(t), \vartheta_{ijz}$ from current state $s(t)$ to decide resource demands, association, and resource selections.

Action Space: At each state s_t, the learning agents select appropriate action a_t from the set of possible actions A, which is defined as $A = \{a_1, a_2, \ldots, a_l\}$, where l represents players/agents. The set of actions is defined as:

$$a(t) = \{\mathbf{\Pi}, \mathbb{R}, \mathbf{\Phi}\}, \quad (45)$$

where $\Pi \in \{\Pi_{1\hat{j}z}, \Pi_{2\hat{j}z}, \ldots, \Pi_{I\hat{j}z}\}$ indicates a set of offloading/association indicators, $\mathbb{R} \in \{\mathbb{R}_{1n}, \mathbb{R}_{2n}, \ldots, \mathbb{R}_{Jn}, \hat{\mathbb{R}}_{1j}, \hat{\mathbb{R}}_{2j}, \ldots, \hat{\mathbb{R}}_{Ij}\}$ indicates a set of different resources selecting by VNO j and SMUE i at time slot t, $\Phi \in \{\Phi_{i1}, \Phi_{i2}, \ldots, \Phi_{iJ}, \Phi_{j1}, \Phi_{j2}, \ldots, \Phi_{jN}\}$ indicates an optimal price of resources adjusted by VNO j and InP n, respectively.

Reward function: An agent can make its own decision in a decentralized execution manner to maximize the reward in a dynamic environment. In the proposed ATG network, we design a reward function to maximize the utility functions of InPs and VNOs while meeting the QoS requirements of SMUEs. At time slot t, the rewards of all agents in different stages are presented as follows. The reward function of InPs at stage-I is expressed as:

$$r_n(t) = U_n(t). \quad (46)$$

The reward function of VNOs at stage-II is expressed as:

$$r_j(t) = U_j(t). \quad (47)$$

The reward function of SMUEs at stage-III is expressed as;

$$r_i(t) = U_i(t). \quad (48)$$

Therefore, the system reward is the sum of all rewards in different stages, and is expressed as;

$$r(t) = \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} \sum_{n\in\mathcal{N}} \big(r_n(t) + r_j(t) + r_i(t)\big). \quad (49)$$

A. MADDPG-based Solution

Since the ATG network environment has a large problem space, multi-objective problems, a multi-agent environment, and unpredictable dynamic changes in time slots t, traditional single-agent DRL algorithms cannot cope with the environment's dynamics and complexity. Therefore, the MADDPG-based approach is appropriate for complex and dynamic

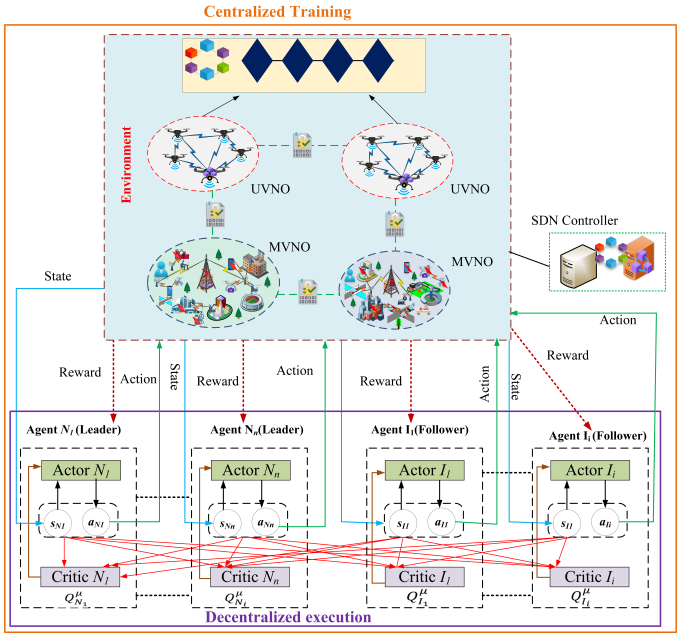


Fig. 3: MADRL framework for virtualized ATG network in 5G-RAN.

multi-agent environments because agents can learn collaboratively and improve network performance. One of the popular MADRL schemes is the MADDPG algorithm [59], which is a customized actor-critic network utilizing the DQN method that can manage a dynamic environment and is effective for cooperative policy learning of multi-agents with continuous action space. As shown in Fig. 3, our framework adopts a strategy based on centralized training and decentralized execution.

Actor network: The actor-network is a function that maps the resource allocation environment state s_t to an action a_t in order to find the best decision policies. In the actor-network, considering N leaders in each slice acting as agents, the neural network's policies are parameterized by $\theta^{\mu} = \{\theta_1^{\mu}, \theta_2^{\mu}, \ldots, \theta_N^{\mu}\}$. Each agent l takes continuous policies $\mu_l \sim \mu_{\theta_l^{\mu}}$ with regard to parameters θ_l^{μ}. For the deterministic policies, we can have a local action on each stage, which is expressed as $a_l = \mu_l(s_l|\theta_l^{\mu})$. To determine the utility function in each stage, we take local actions aggregated into a joint action L, which manages each player's role in each stage.

Critic network: The critic network is primarily used to evaluate the value of the agents' actions. It supplies information about the global state S and the policies of all agents. All states $(s_1, s_2, \ldots, s_L)$ and agents' actions $(a_1, a_2, \ldots, a_L)$ are inputs for the critic network. This means the critic $Q(S, A|\theta^Q)$ is a centralized action-value function with parameter θ^Q.

Target Network: To stabilize the training, DDPG employs an experience replay buffer in conjunction with a target network. The target policy of agent l is represented as $\mu_l^{'}$ with parameters $\theta_l^{\mu^{'}}$. The target of the critic is Q' with parameter $\theta_l^{Q\prime}$. Therefore, the parameters of actor and critic networks are periodically updated with the most recent θ_l^{μ} and θ^Q, which are expressed as:

$$\begin{aligned} \theta_l^{\mu^{'}} &\rightarrow \tau\theta_l^{\mu} + (1-\tau)\theta_l^{\mu^{'}} \\ \theta^{Q\prime} &\rightarrow \tau\theta^{Q} + (1-\tau)\theta^{Q^{'}}, \end{aligned} \tag{50}$$

where $\tau \in [0, 1]$ is soft update.

Experience replay buffer: Each agent has its own local experience replay buffer $(\mathcal{M}_l^{loc})$ storing tuples $\{s_l, a_l, r_l, s'\}$. Depending on the local observation state $s_l(t)$ from the environment, the agent l can take action $a_l(t)$ and return a reward $r_l(t)$. If the channel link is available and there is no network traffic on BS, agent l uploads the local information $s_l(t), a_l(t), r_l(t)$ to BS for successive training, else agent l uploads onto the nearest UCH u. Each agent should upload its experience replay buffer $\mathcal{M}_l^{loc}$ to the associated BS/UCH and download the global training weight of the actor-network $\mathcal{M}^u$ or $\mathcal{M}^k$ from the BS/UCH. The agent l then adjusts its policy and decides whether or not to allocate resources, offload the task through a subchannel, or compute locally. This relies on the current observation and policy's effect at the time frame t. At each t time slot, the global experience replay buffer stores a collection of tuples for all agents, including state, action, reward, and transition state denoted as $\mathcal{M} = \{e_1, e_2, \ldots, e_t\}$, where $e_t = \{s_t, a_t, r_t, s_{t+1}\}$, $\mathcal{M}^u, \mathcal{M}^k \in \mathcal{M}$. During centralized training, the critic network employs the actions and states of all agents to approximate the Q-value function of the current action of agent l. After each training episode, the actor and critic networks will be updated in the MADDPG algorithm. The gradient method is used to update the actor-network, and the update is calculated as follows:

$$\begin{aligned} \nabla_{\theta_l}(\theta_l) = \mathbb{E}_{S,a\sim\mathcal{M}}[\nabla_{a_l}\theta_l^{\mu}(S, a_1, \ldots, a_L) \\ \nabla_{\theta l}\mu\theta_l(s_l)|_{al} = \mu\theta_l(s_l)], \end{aligned} \tag{51}$$

where $\mathcal{M}$ is the experience replay buffer storing all agent experiences including $(S, S', a_1, \ldots, a_L, r_1, \ldots, r_L)$. The critic function θ_l^{μ} is updated as the following expression:

$$\mathcal{L}(\theta_l) = \mathbb{E}_{S,S',a_1,\ldots,a_L,r_1,\ldots,r_L}[(\theta_l^{\mu}(S, a_1, \ldots, a_L) - y)^2], \tag{52}$$

where $y = r_l + \gamma\theta_l^{\mu^{'}}(s', a_1', \ldots, a_L')|_{a_j'=\mu_j'(s_j)}$.

The loss function in (52) updates the critic network Q_l^{μ}, and the actor-network is updated by minimizing the policy

Algorithm 2 MADDPG algorithm for resource allocation

Initialize: For all agents l, $l \in \{1, 2, \ldots, L\}$, randomly initialize the critic network $Q(S, A|\theta^Q)$ and actor-network $\mu_l(s_l|\theta_l^{\mu})$ including parameter weights θ^Q and θ_l^{μ}
Initialize: Target networks Q_l' and μ_l' including weights $\theta^{Q'}$, $\theta_l^{\mu'}$
Initialize: the global replay buffer $\mathcal{M}$ at the SDN controller, and $\mathcal{M}^{loc}$ leaders, followers
Initialize: The soft update τ and discount factor γ
for episode = $1, 2, \ldots, K$ **do**
  Initialize state space $S = \{s_1, s_2, \ldots, s_L\}$
  Execute Algorithm 1
  for $t = 1, 2, \ldots, 200$ **do**
    Each agent select action with noise $\mathcal{N}$ $a_l = \mu_l(s_l|\theta_i^{\mu}) + \mathcal{N}(t)$
    All agent execute action $a(t) = \{a_1(t), a_2(t), \ldots, a_L(t)\}$
    Observe rewards $r(t) = \{r_1(t), r_2(t), \ldots, r_L(t)\}$
    Observe the new state $s_l(t+1)$;
    Save the tuples $\{s_l(t), a_l(t), r_l(t), s_l'(t+1)\}$ in $\mathcal{M}_l^{loc}$
    Agent l upload the tuple value from $\mathcal{M}_l^{loc}$ to controller $\mathcal{M}$
    Merge follower and leaders $\mathcal{M}_l^{loc}$ into $\mathcal{M}$ at SDN controller
    Download $\mathcal{M}$ from SDN controller
    $s_l \leftarrow s_l'$
    for agent $l = 1$ to L **do**
      Sample a random mini-batch of H samples tuples (s^j, a^j, r^j, s'^j) from $\mathcal{M}$;
      Set $y^j = r_l^j + \gamma Q_l^{\pi^{'}}\left(S'^j, a_1', \ldots, a_L'\right)|_{a_t'=\mu_s'(s_t^j)}$
      Update critic-network by minimizing loss as (52)
      Update actor using sample policy gradient as (51)
    end for
    Update the target network parameters for each agent l as (50)
  end for
end for

gradient of agent l, where H is a random mini-batch size of samples, and it is expressed as:

$$\nabla_{\theta_l} J \approx \frac{1}{H} \sum_j \nabla_{\theta_l} \mu_l(o_l^j) \nabla_{a_l} Q_l^{\mu}\left(K^j, a_1^j, \ldots, a_l^j\right)|_{a_l=\mu_l(o_l^j)}, \tag{53}$$

where j is the index of samples. The MADDPG algorithm, as presented in **Algorithm 2** has two procedures: collecting observed data and training procedures. We first initialize the replay buffer, actor-network, and critic network parameters with weight (lines 1-3), and define the number of episodes and training time steps. Then, the agents collect observed data (lines 3-14). Agent at stage-I determines the price of resources by observing state s_{t-1}. Agent of stage II estimates the resource demands based on the price set at stage I, both s_t and s_{t-1}. Agent of stage II also determines the price of resources by observing s_{t-1}. Agent of stage III decides on association and resource demand based on the price set at stage II, both s_t and s_{t-1}. Estimate the available resources at each stage in different slices (line 7). Each agent executes the action, obtains the reward, and generates a new state (lines 9-14). The experience is stored in the experience replay buffer. In the training step (lines 14-21), we use a policy training that uses batch sampling from the replay buffer. Then, the actor-network and the critic-network are updated based on a randomly selected sample.

B. Complexity Analysis

In the proposed algorithm, the actor-network and the critic network are fully connected. Let Z_n^{ac} be the number of neurons in the n-th layer of the actor-network, for n-th layer of actor-network computational complexity is $\mathcal{O}\left(Z_{n-1}^{ac} Z_n^{ac} + Z_n^{ac} Z_{n+1}^{ac}\right)$ and for N-th layer of actor-network computational complexity is calculated as $\mathcal{O}\sum_{n=2}^{N-1}\left(Z_{n-1}^{ac} Z_n^{ac} + Z_n^{ac} Z_{n+1}^{ac}\right)$. We define Z_m^{cr} as the number of neurons of m-th layer of critic network, then the computational complexity of m-th layer is expressed as $\mathcal{O}\left(Z_{m-1}^{cr} Z_m^{cr} + Z_m^{cr} Z_{m+1}^{ac}\right)$ and for M-th layer of critic network computational complexity is $\mathcal{O}\sum_{m=2}^{M-1}\left(Z_{m-1}^{cr} Z_m^{cr} + Z_m^{cr} Z_{m+1}^{cr}\right)$. The proposed algorithm computational complexity for training is calculated as $\mathcal{O}\Big(3 * H * T * L * K\big(\sum_{n=2}^{N-1}\left(Z_{n-1}^{ac} Z_n^{ac} + Z_n^{ac} Z_{n+1}^{ac}\right) + \sum_{m=2}^{M-1}\left(Z_{m-1}^{cr} Z_m^{cr} + Z_m^{cr} Z_{m+1}^{cr}\right)\big)\Big)$, where H, L, T, and K denote the mini-batch size, number of agents, training steps, and maximum episode, respectively. The execution complexity of each agent is calculated as $\mathcal{O}\Big(T * K\big(\sum_{n=2}^{N-1}\left(Z_{n-1}^{ac} Z_n^{ac} + Z_n^{ac} Z_{n+1}^{ac}\right)\big)\Big)$.

VII. Performance Evaluation

A. Scenario Configuration

In our simulations, we used a core i7 server with a 2.4GHz Intel Xeon CPU and 32GB RAM to evaluate the performance of our proposed MADDPG algorithm in a blockchain-based multi-UAV-enabled 5G-RAN. We used TensorFlow 2.00 with Python 3.7 on Windows 10 for our experiments.

The ATG network system has ground, and aerial infrastructures owned by InPs, which are virtualized into three slices managed by different MVNOs and UVNOs (1 BS and 1 UCH in each slice). Each slice has an average number of 60 SMUE, which are distributed uniformly. The BS coverage radius is 500m, and the UAV network (UCH) coverage is r_u=800m with altitude H_u=100m. For the probabilistic model, $\iota = 9.61, \varrho = 0.16$, carrier frequency is f = 2GHz, path-loss $\eta^{LoS} = 1$, and $\eta^{NLoS} = 20$. The system bandwidth is 20 MHz, divided into 100 RBs for each BS and UCH, and each RB occupies 180 kHz of bandwidth. The overall transmission power is 125 dBm; each BS and UCH have 30 dBm and 25 dBm transmission power, respectively. The computation capacity is 150 GHz/s, divided into 100 each for BS and UCH, which have 20 GHz/s and 15GHz/s computation capacities, respectively. The sizes of computing tasks of SMUEs are distributed in [100-15000] KBps in each slice. The required CPU cycle for computing tasks is distributed in the range [0.5-1.5] Gcycle. The prices of computing, radio, and transmission power resources are distributed in the ranges of [1-10] units/Mbps, [0.5-2.5] units/MHz, and [1-5] units/J, respectively. The transaction block size ranges from [50-500] KB, with a 0.42-second block propagation latency. Every transaction in the block is between 0 and 0.573 KB in size.

TABLE II: Simulation Parameters.

Parameters	Values
Number of MVNO, BS	3
Number of UVNO, UCH	3
Number of SMUEs	[20-60]
BS coverage radius	500m
UCH coverage radius	800m
Distance between BS	100m
Distance between UCH	120m
UCH altitude	100m
System bandwidth	20MHz
Computation capacity of BS and UCH	20GHz/s, 15GHz/s
Size of computation task	[100-15000]KBps
Required CPU cycle for task	[0.5-1.5]Gcycle
Number of episodes	2500
Discount factor	0.9
Mini batch size	128
learning rate	0.01
Soft update parameter	0.001

We deploy a fully connected neural network (NN) with a critic-network and an actor-network in this proposed multi-agent approach. For each agent, we deploy four hidden layers in both the actor and the critic NNs set as 64 in the first layer and 32 in the second layer. We set the mini-batch size as 256 and the replay buffer size as 10^7. We set agent action selection probability $\epsilon \in [0, 1]$ and utilize ReLU as the activation function for the hidden layers, and the sigmoid function is employed at the output layer. To optimize the loss function, we use the Adam Optimizer.

We evaluate the performance of our proposed MADDPG algorithm by comparing it with baseline algorithms (MADQN [60], DDPG [52], A3C [61], Greedy [52]) in slice scenarios. Other simulation parameters are summarized in Table II.

B. Total Utility Analysis

From Fig. 4, we can observe that the total system utility stabilizes as the number of episodes increases. All four al-

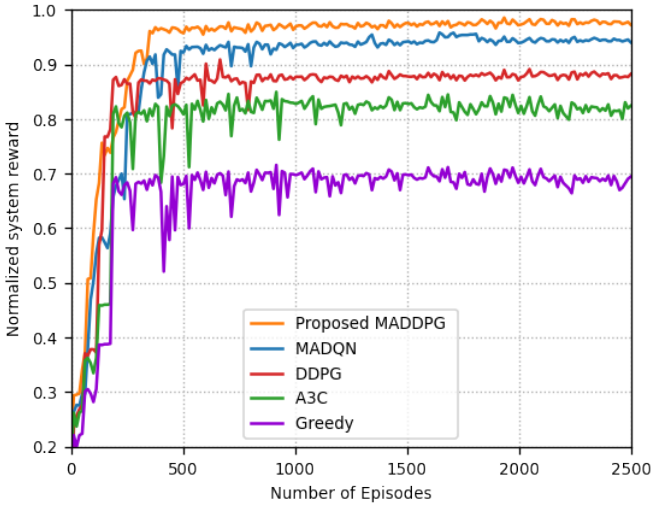
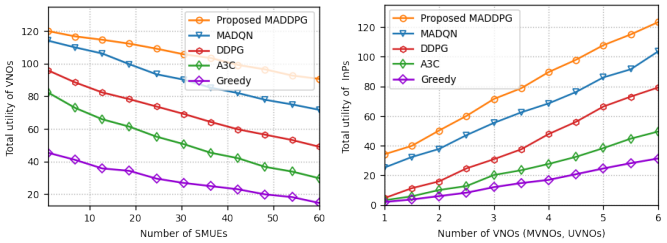


Fig. 4: System reward with episodes.

gorithms achieve convergence, implying that the agents can obtain optimal policies in different episodes of our proposed framework. The proposed MADDPG algorithm achieves better convergence with approximately 450 episodes, and has better system utility, with a system utility of around 125. Each agent observes other agents' actions and makes decisions independently to maximize utility, depending on other agents' actions and current states. Compared with MADDPG, MADQN has lower performance but is still better than the other two algorithms. The three algorithms converge slower than the MADDPG after 500 to 900 episodes, with a system utility of 118 to 105. The greedy algorithm's performance and reward value are worse than other algorithms because it does not consider future findings. This implies that baseline algorithms have lower decision policies than proposed ones. We can conclude that the proposed MADDPG algorithm in virtualized multi-UAV enabled 5G-RAN with blockchain technology can obtain the best optimal policies to maximize the total system utility and satisfy the QoS of the SMUEs, compared with other baseline algorithms.



(a) VNOs' utility. (b) InPs' utility.

Fig. 5: Comparison of utility of VNOs and InPs with varying numbers of SMUEs and VNOs, respectively.

C. Impacts of SMUEs and VNOs in the System Utility

In this simulation, we evaluate the effects of increasing SMUEs and VNOs in the utilities of both VNOs and InPs, which indirectly impacts the system utility. In our scenario, we consider two folds of the resource allocation process: InPs and VNOs in Stackelberg game stages I and II, and VNOs and SMUEs in Stackelberg game stages II and III. Then, we consider 2 InPs as resource providers at stage I, and 6 VNOs (MVNOs and UVNOs) buy resources from InPs at stage II. The VNOs sell/provide resources to 60 SMUEs in the virtualized network. Based on this, Fig. 5a and 5b show the impacts of SMUEs and VNOs on the utilities of VNOs and InPs, respectively.

Fig. 5a illustrates the total utilities of different algorithms concerning the number of SMUEs. As the number of SMUEs increases, the utilities of all algorithms continue to decrease. However, the utilities of DDPG and A3C algorithms are decreasing more rapidly than the proposed MADDPG and MADQN schemes; the main reason for this performance is that the agents can not learn from other agents' actions and take quick action when the BS is under failure or overloaded situation and SMUEs requests a lot of resources. In addition, the greedy algorithm had the lowest utility value of the four algorithms due to a lack of learning. Furthermore, the utilities of the proposed algorithm are higher and decrease more slowly than other baseline algorithms.

Fig. 5b shows that as the number of VNOs increases, the InPs utility increases in all algorithms. This is due to the fact that raising the VNOs enables more resource requesters in stage III to obtain more system resources. However, the A3C and DDPG algorithms have lower InPs utilities and better than greedy, which implies lower system utilities than the MADDPG and MADQN schemes. Then, VNOs can allocate many resources to SMUEs, while collecting digital coins increases the utility. Therefore, we conclude that the proposed MADDPG algorithm obtained higher InPs utility than baseline algorithms in a virtualized UAVs-enabled 5G-RAN.

D. Effects of Resource Price on VNOs Utilities

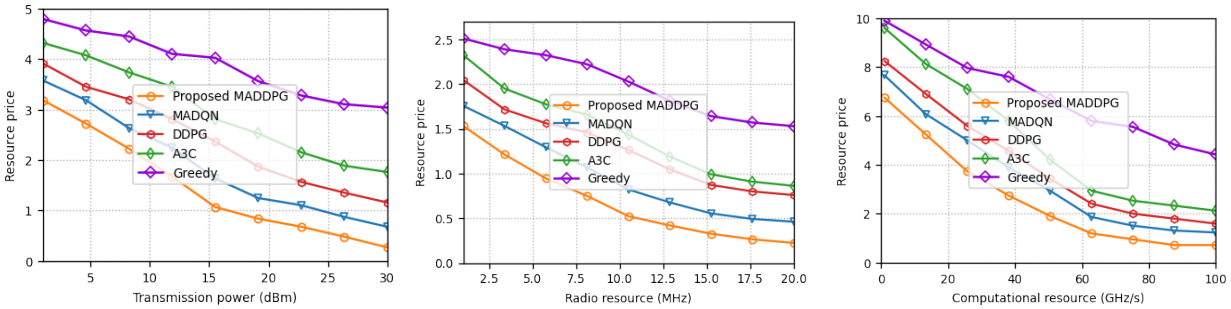
In this simulation, we evaluate the effects of resource price increases for different resource types. From Fig. 6, one can see that an increase in resource prices can increase the VNOs' utility as well as the system's utility. However, our proposed MADDPG algorithm achieves better utility compared to the baseline algorithms. For simplicity in the illustration, we consider only VNOs as resource providers to SMUEs, making the analysis suitable for stage II.

Fig. 6a shows the effects of ϕ, which are unit computing resource prices for executing the offloading tasks in a different virtualized network. We can observe that the proposed scheme allows VNOs to obtain high utilities when resource prices for computing offloading tasks increase. When SMUEs acquire more resources from VNOs, energy consumption and latency can be minimized, along with the associated expenses. The task offloading increases even if the resource price per resource increases and SMUEs prefer to offload computing-intensive tasks. Therefore, the proposed scheme ensures SMUEs satisfaction and maximizes VNOs. utilities more than baseline schemes.

Fig. 6b and 6c show the effects of ψ, η, which are unit prices for accessing transmission power and radio in a different

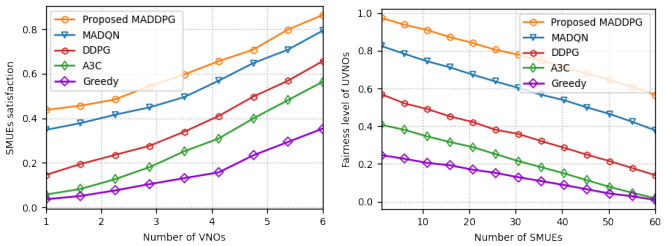
(a) (b) (c)

Fig. 6: Effects of resources price on system utility.



(a) (b) (c)

Fig. 7: Comparison of different resources price based on the amount of resource.



(a) SMUEs' satisfaction. (b) Fairness level.

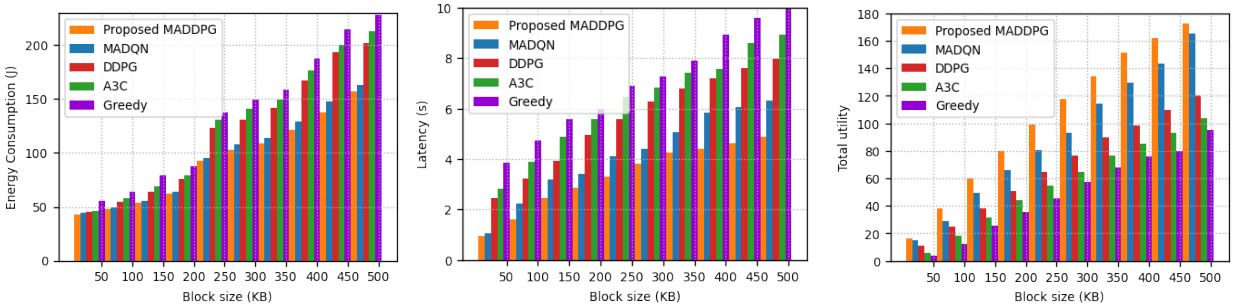
Fig. 8: Comparison of SMUEs' satisfaction under different VNO numbers and the UCH fairness level under different SMUE numbers.

virtualized network. From Fig. 6b, we observe that as the unit price for transmission power increases, the utilities of VNOs increase. We can observe that the proposed scheme can obtain the highest utilities with increased prices for transmission power. When the SMUEs are charged more to get transmission power, it can resolve the power issues. The energy constraint can be minimized by offloading more computing-intensive tasks, even if the price per resource increases. In this way, the SMUE can reduce its computation costs, energy consumption, and latency, resulting in better QoS satisfaction. In this regard, the proposed scheme ensures SMUEs satisfaction and maximizes VNOs' utilities more than baseline schemes.

From Fig. 6c, we observe that the VNOs' utilities increase with increasing prices of bandwidth resources in all schemes. It indicates that the bandwidth demands of SMUE increase to access/lease and connect with networks in a short period of time. The MADQN, DDPG, and A3C schemes' utilities increase slowly due to agents' struggles to connect with the nearest node, and the resources are insufficient for all SMUEs. The greedy algorithm utility increases most slowly, and the maximum utility is approximately 28.65; the SMUEs pay a higher price and incur more costs. However, with the proposed scheme, system utility rapidly increases after bandwidth prices exceed 1.25 units/MHz. It implies the agents prefer to access bandwidth resources to maintain the connection problem, ensure QoS, and minimize the computation costs of SMUEs.

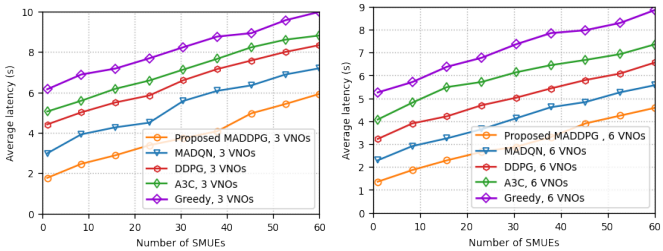
E. Analysis of Resource Allocation

In Fig. 7, we evaluate the proposed MADDPG algorithm's performance in pricing and allocating resources between the resource providers and resource requesters, comparing it with the baselines. From Fig. 7a, we observe that the amount of transmission power increases while its price decreases in all algorithms. When the transmission power is 10 dBm, the SMUEs will pay approximately 1.99, 2.45, 3.0, 3.65, and 4.65 through the proposed MADDPG, MADQN, DDPG, A3C, and greedy algorithms, respectively. After the transmission power reached 10 dBm, both algorithms quickly declined the price per transmission power resource. It implies that with limited resources, the resource provider sets maximum prices due to the limitation of resources and higher demands. The proposed MADDPG algorithm decreases the pricing by 26.62%, 49.16%, 69.84%, 85.53% MADQN, DDPG, A3C, and greedy, respectively.



(a) Block size vs energy consumption. (b) Block size vs latency. (c) Block size vs utility.

Fig. 9: Effects of block size.



(a) (b)

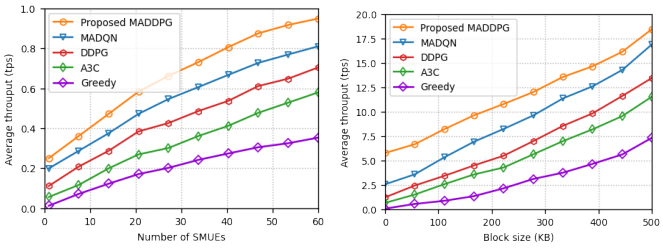
Fig. 10: Number of SMUEs with average latency.

From Fig. 7b, we notice that when the amount of bandwidth increases, the bandwidth price in all algorithms decreases. When the bandwidth is around 6.5 MHz, the resource price is approximately 0.947, 1.296, 1.564, 1.776, and 2.345 through the proposed MADDPG, MADQN, DDPG, A3C, and greedy algorithms, respectively. After the bandwidth reached 7 MHz, both algorithms quickly reduced the price per bandwidth. Due to limited radio resources and increasing demand, the resource provider can set the maximum price. The proposed MADDPG algorithm decreases the pricing by 33.03%, 60.00%, 71.34%, and 96.29% for MADQN, DDPG, A3C, and greedy, respectively.

From Fig. 7c, we notice that when the number of computational resources increases, the computational resource price in all algorithms decreases. When the computational resource is around 15 GHz/s, the resource price is approximately 5.238, 6.071, 6.898, and 8.122 through the proposed MADDPG, MADQN, DDPG, and A3C algorithms, respectively. After the computational resources reached 15 GHz/s, both algorithms quickly reduced the price per computational resource. Due to limited computational capacity and increasing offloading demand, the resource provider can set the maximum price. The proposed MADDPG algorithm decreases the pricing by 27.23%, 41.69%, 61.02% and 86.91% for MADQN, DDPG, A3C, and greedy, respectively.

F. Analysis of SMUEs' Satisfaction and Fairness level

Fig. 8a shows the impact of the number of VNOs on SMUEs satisfaction rate. We observed that the SMUE satisfaction rate increases as the number of VNO increases. This is because more resources are available at optimal prices, which motivates SMUEs to access different resources. The satisfaction rate of SMUEs increases rapidly, particularly after the number of VNOs exceeds 3 in all schemes. Even though MADQN yields better SMUEs' QoS satisfaction than DDPG and A3C, the proposed scheme achieved superior performance than all baseline schemes. The MADDPG scheme increases the SMUEs satisfaction rate by 31.69%, 51.065%, 59.23%, and 72.19%, MADQN, DDPG, A3C and greedy, respectively. Fig. 8b depicts the effect of the number of SMUE on the UCHs coverage level (UCH fairness level). We observe that the fairness level of UCH decreases rapidly in the four baseline schemes. The increasing number of SMUEs would complicate ATG communication services by requesting more resources and consuming more UCH resources. The fairness level is reduced by MADQN, DDPG, A3C and greedy, 30.869%, 49.58%, 65.773%, and 81.61%, respectively, compared with the MADDPG scheme. However, the MADDPG scheme would work with the agents to reduce the UCH burden and improve communication services. As a result, as the number of SMUEs increases, the fairness level of UCH gradually decreases.



(a) (b)

Number of SMUEs and trans.block with throughput.

G. Blockchain Performance Analysis

This part analyzes the impact of block size on energy consumption, latency, and utility. From Fig. 9a and 9b , which shows the number of block sizes increasing from 50 KB to 400 KB, the energy consumption and latency of block processing are increasing simultaneously. As shown in Fig. 9a, we observe that the energy consumption of all algorithms is lower than

80J between 50 KB and 200KB block size. In particular, after 250KB, the energy consumption increases quickly in the DDPG, A3C, and greedy algorithms. This implies that the network is congested with non-cooperative resource allocation and blockchain operation processes, and each agent strives to achieve its optimal policy. The greedy algorithm's energy consumption is worse than others due to its limited consensus experience, decision-making capabilities, and ability to obtain optimal association. However, in MADDPG and MADQN, the agents cooperatively learn optimal policies and share experiences in a distributed fashion; due to this, the network operator does not consume more resources. Thus, with these two algorithms, the energy consumption increases more slowly than with the DDPG, A3C, and greedy algorithms.

Fig. 9b compares different algorithms in terms of communication latency versus block size. As we observe from this figure, the communication latency increases with increasing block size in all algorithms. However, with the proposed MADDPG and MADQN algorithms, the communication latency rises more slowly than with other algorithms. Moreover, the communication latency of the proposed algorithm is still lower than that of the MADQN, DDPG, A3C, and greedy algorithms. This implies that the baseline algorithms face both computing data and a lack of efficient resource allocation.

In Fig. 9c, we compare the total utilities with different block sizes. This figure shows that the proposed scenario can allow resource providers to achieve more utilities with increasing block size in all algorithms. Nevertheless, the total utilities are not increasing rapidly due to different constraints, such as energy consumption and latency of the blockchain system, which increase as block size increases. Time-sensitive data can get more priority and ensure security in the MADDPG and MADQN algorithms. The total utility increased in the A3C and DDPG algorithms, but the MADDPG and MADQN algorithms performed better. The overall utility of the greedy algorithm is lower than that of another algorithm and increases slowly. However, the proposed MADDPG algorithm outperforms the baseline algorithms and increases the total utilities by 11.63%, 35.51%, 50.12%, and 78.34% compared with the MADQN, DDPG, A3C, and greedy algorithms, respectively.

Moreover, we evaluated the performance of the blockchain with increasing resource providers and requesters. Fig. 10 shows that as the number of SMUE in the system increases, so does the latency of the blockchain in all compared schemes. We evaluated the blockchain latency with 3 VNOs and the number of SMUEs ranging from 5-60, as shown in 10a. As a result, while the latency of all compared algorithms increases, the MADDPG algorithm achieves the lowest latency. Similarly, we evaluated the latency achievements of the compared schemes with 6 VNOs and a range of 5-60 SMUEs, as shown in Fig. 10b. Based on this figure, one can conclude that the increases in VNOs and SMUEs can achieve better blockchain latency. Furthermore, the proposed MADDPG has lower latency than the other baseline schemes. This is because increasing both VNOs and SMUEs can reduce system resource scarcity and SMUE costs. The MADDPG algorithm handles the complexity raised by multiple VNOs and SMUEs. In addition, we compare the average throughput of the proposed scheme and the baseline schemes as the number of SMUEs and block size increase. Fig. 11a shows the average throughput as the number of SMUEs increases. The figure shows that the throughput of all compared algorithms increases as the number of SMUEs increases. It is because transaction volume increases when the number of SMUEs grows. However, the proposed MADDPG algorithm has higher throughput than the other schemes. The reason for this achievement is that MADDPG can handle the complexity of resource allocation. On the other hand, Fig. 11b shows the simulation results that show how block size affects the average throughput of the compared algorithms. The figure shows that all algorithms have higher throughput as block size increases. However, the proposed MADDPG algorithm outperforms other schemes.

VIII. Conclusion

In this paper, we presented a new framework for resource allocation in multi-UAV-enabled 5G-RAN that combine MADRL schemes, virtualization, and blockchain technology. To maximize the utilities, we considered a joint optimization of optimal resource allocation and optimal pricing per resource. The VNOs lease virtual resources from the InPs and sublease resources to the SMUEs alleviating the resource scarcity of SMUEs. We formulated the optimization problems of InPs, VNOs, and SMUEs as a three-stage Stackelberg game model. Since the defined optimization problems are complex and continuous action space problems, we transformed the optimization problems into MDP games to handle the complexity and dynamics of the system. We then applied a MADRL algorithm called the MADDPG to solve the problems that adopted centralized training with a decentralized execution scheme. Each InP, VNO, and SMUE is an agent to obtain an optimal offloading and resource allocation policy to maximize utilities and ensure QoS. To achieve the objective, the agent cooperates and shares information while making an independent decision. The MADDPG scheme ensures better convergence than the MADQN, DDPG, A3C, and greedy schemes. The simulation results showed that the proposed MADDPG scheme for blockchain-based computation offloading and resource allocation maximizes utilities and ensures QoS. Therefore, the proposed MADRL framework with blockchain technology plays a vital role in multi-UAV-enabled 5G-RAN to ensure security and maximize utility while simultaneously improving the network's performance. In future work, we will explore the distributed machine-learning-based virtualized space-air-ground integration networks (SAGINs) in digital twin-enabled 6G to allocate resources to resolve diversified SMUE demands.

References

[1] S. Henry, A. Alsohaily, and E. S. Sousa, "5g is real: Evaluating the compliance of the 3gpp 5g new radio system with the itu imt-2020 requirements," *IEEE Access*, vol. 8, pp. 42 828–42 840, 2020.

[2] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5g wireless network slicing for embb, urllc, and mmtc: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55 765–55 779, 2018.

[3] X. Wang, P. Krishnamurthy, and D. Tipper, "Wireless network virtualization," in *2013 International Conference on Computing, Networking and Communications (ICNC)*, 2013, pp. 818–822.

[4] C. Liang and F. R. Yu, "Wireless virtualization for next generation mobile cellular networks," *IEEE Wireless Communications*, vol. 22, no. 1, pp. 61–69, 2015.

[5] F. Song, J. Li, C. Ma, Y. Zhang, L. Shi, and D. N. K. Jayakody, "Dynamic virtual resource allocation for 5g and beyond network slicing," *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 215–226, 2020.

[6] P. L. Vo, M. N. H. Nguyen, T. A. Le, and N. H. Tran, "Slicing the edge: Resource allocation for ran network slicing," *IEEE Wireless Communications Letters*, vol. 7, no. 6, pp. 970–973, 2018.

[7] Y. Gao, J. Cao, P. Wang, J. Yin, M. He, M. Zhao, M. Peng, S. Hu, Y. Sun, J. Wang, S. Cheng, Y. Guo, Y. Du, Y. Cai, J. Huang, and K. Qiu, "Intelligent uav based flexible 5g emergency networks: Field trial and system level results," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020, pp. 138–143.

[8] Z. Xiong, Y. Zhang, W. Y. B. Lim, J. Kang, D. Niyato, C. Leung, and C. Miao, "Uav-assisted wireless energy and data transfer with deep reinforcement learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 85–99, 2021.

[9] Y. Gao, J. Cao, P. Wang, J. Wang, M. Zhao, S. Cheng, S. Hu, and W. Lu, "Uav based 5g wireless networks: A practical solution for emergency communications," in *2020 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science*, 2020, pp. 1–4.

[10] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and L. Hanzo, "Multi-agent deep reinforcement learning-based trajectory planning for multi-uav assisted mobile edge computing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 73–84, 2021.

[11] P. Wei, K. Guo, Y. Li, J. Wang, W. Feng, S. Jin, N. Ge, and Y.-C. Liang, "Reinforcement learning-empowered mobile edge computing for 6g edge intelligence," *IEEE Access*, vol. 10, pp. 65 156–65 192, 2022.

[12] Z. Xiong, Y. Zhang, D. Niyato, R. Deng, P. Wang, and L. Wang, "Deep reinforcement learning for mobile 5g and beyond: Fundamentals, applications, and challenges," *IEEE Vehicular Technology Magazine*, vol. 14, no. 2, pp. 44–52, 2019.

[13] Y. Zhou, F. Tang, Y. Kawamoto, and N. Kato, "Reinforcement learning-based radio resource control in 5g vehicular network," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 611–614, 2020.

[14] F. Tang, Y. Zhou, and N. Kato, "Deep reinforcement learning for dynamic uplink/downlink resource allocation in high mobility 5g hetnet," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 12, pp. 2773–2782, 2020.

[15] M. Elsayed and M. Erol-Kantarci, "Reinforcement learning-based joint power and resource allocation for urllc in 5g," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[16] X. Wang and T. Zhang, "Reinforcement learning based resource allocation for network slicing in 5g c-ran," in *2019 Computing, Communications and IoT Applications (ComComAp)*, 2019, pp. 106–111.

[17] M. Alsenwi, N. H. Tran, M. Bennis, S. R. Pandey, A. K. Bairagi, and C. S. Hong, "Intelligent resource slicing for embb and urllc coexistence in 5g and beyond: A deep reinforcement learning based approach," *arXiv preprint arXiv:2003.07651*, 2020.

[18] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*, 2017, pp. 557–564.

[19] A. Chaer, K. Salah, C. Lima, P. P. Ray, and T. Sheltami, "Blockchain for 5g: Opportunities and challenges," in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.

[20] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[21] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2019.

[22] W. Chen, Z. Xu, S. Shi, Y. Zhao, and J. Zhao, "A survey of blockchain applications in different domains," ser. ICBTA 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 17–21. [Online]. Available: https://doi.org/10.1145/3301403.3301407

[23] A. Chaer, K. Salah, C. Lima, P. P. Ray, and T. Sheltami, "Blockchain for 5g: opportunities and challenges," in *2019 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2019, pp. 1–6.

[24] L. Xue, W. Yang, W. Chen, and L. Huang, "Stbc: A novel blockchain-based spectrum trading solution," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 1, pp. 13–30, 2022.

[25] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, L. Lyu, and Y. Liu, "Privacy-preserving blockchain-based federated learning for iot devices," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1817–1829, 2021.

[26] S. Zhou, H. Huang, W. Chen, P. Zhou, Z. Zheng, and S. Guo, "Pirate: A blockchain-based secure framework of distributed machine learning in 5g networks," *IEEE Network*, vol. 34, no. 6, pp. 84–91, 2020.

[27] Q. Hu, W. Wang, X. Bai, S. Jin, and T. Jiang, "Blockchain enabled federated slicing for 5g networks with ai accelerated optimization," *IEEE Network*, vol. 34, no. 6, pp. 46–52, 2020.

[28] A. M. Seid, J. Lu, H. N. Abishu, and T. A. Ayall, "Blockchain-enabled task offloading with energy harvesting in multi-uav-assisted iot networks: A multi-agent drl approach," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3517–3532, 2022.

[29] A. E. Azzaoui, S. K. Singh, Y. Pan, and J. H. Park, "Block5gintell: Blockchain for ai-enabled 5g networks," *IEEE Access*, vol. 8, pp. 145 918–145 935, 2020.

[30] C. Qiu, H. Yao, X. Wang, N. Zhang, F. R. Yu, and D. Niyato, "Ai-chain: Blockchain energized edge intelligence for beyond 5g networks," *IEEE Network*, vol. 34, no. 6, pp. 62–69, 2020.

[31] J. Gil Herrera and J. F. Botero, "Resource allocation in nfv: A comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, 2016.

[32] A. Belbekkouche, M. M. Hasan, and A. Karmouch, "Resource discovery and allocation in network virtualization," *IEEE Communications Surveys Tutorials*, vol. 14, no. 4, pp. 1114–1128, 2012.

[33] Z. Yuan and G.-M. Muntean, "Airslice: A network slicing framework for uav communications," *IEEE Communications Magazine*, vol. 58, no. 11, pp. 62–68, 2020.

[34] Y.-H. Xu, J.-H. Li, W. Zhou, and C. Chen, "Learning-empowered resource allocation for air slicing in uav-assisted cellular v2x communications," *IEEE Systems Journal*, pp. 1–4, 2022.

[35] S. Song, C. Lee, H. Cho, G. Lim, and J. Chung, "Clustered virtualized network functions resource allocation based on context-aware grouping in 5g edge networks," *IEEE Transactions on Mobile Computing*, vol. 19, no. 5, pp. 1072–1083, 2020.

[36] H. Cao, Y. Hu, and L. Yang, "Towards intelligent virtual resource allocation in uavs-assisted 5g networks," *Computer Networks*, vol. 185, p. 107660, 2021.

[37] C. Zhang, M. Dong, and K. Ota, "Fine-grained management in 5g: Dql based intelligent resource allocation for network function virtualization in c-ran," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 428–435, 2020.

[38] C. Qiu, H. Yao, F. R. Yu, F. Xu, and C. Zhao, "Deep q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5871–5883, 2019.

[39] G. O. Boateng, G. Sun, D. A. Mensah, D. M. Doe, R. Ou, and G. Liu, "Consortium blockchain-based spectrum trading for network slicing in 5g ran: A multi-agent deep reinforcement learning approach," *IEEE Transactions on Mobile Computing*, pp. 1–15, 2022.

[40] N. Pathak, A. Mukherjee, and S. Misra, "Aerialblocks: Blockchain-enabled uav virtualization for industrial iot," *IEEE Internet of Things Magazine*, vol. 4, no. 1, pp. 72–77, 2021.

[41] N. Hu, Z. Tian, Y. Sun, L. Yin, B. Zhao, X. Du, and N. Guizani, "Building agile and resilient uav networks based on sdn and blockchain," *IEEE Network*, vol. 35, no. 1, pp. 57–63, 2021.

[42] H. Xu, W. Huang, Y. Zhou, D. Yang, M. Li, and Z. Han, "Edge computing resource allocation for unmanned aerial vehicle assisted mobile network with blockchain applications," *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, pp. 3107–3121, 2021.

[43] C. Yao, C. Jiang, Z. Liu, J. Chen, and J. Li, "Optimal capacity allocation and caching strategy for multi-uav collaborative edge caching," in *2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2021, pp. 905–910.

[44] T. Wang, Y. Lu, J. Wang, H.-N. Dai, X. Zheng, and W. Jia, "Eihdp: Edge-intelligent hierarchical dynamic pricing based on cloud-edge-client collaboration for iot systems," *IEEE Transactions on Computers*, vol. 70, no. 8, pp. 1285–1298, 2021.

[45] A. Asheralieva and D. Niyato, "Distributed dynamic resource management and pricing in the iot systems with blockchain-as-a-service and uav-enabled mobile edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1974–1993, 2020.

[46] X. Lin, J. Wu, A. K. Bashir, J. Li, W. Yang, and J. Piran, "Blockchain-based incentive energy-knowledge trading in iot: Joint power transfer and ai design," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[47] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource trading in blockchain-based industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3602–3609, 2019.

[48] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5g network slice broker," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 32–39, 2016.

[49] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal lap altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[50] A. M. Seid, G. O. Boateng, B. Mareri, G. Sun, and W. Jiang, "Multi-agent drl for task offloading and resource allocation in multi-uav enabled iot edge network," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4531–4547, 2021.

[51] H. N. Abishu, A. M. Seid, Y. H. Yacob, T. Ayall, G. Sun, and G. Liu, "Consensus mechanism for blockchain-enabled vehicle-to-vehicle energy trading in the internet of electric vehicles," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 946–960, 2022.

[52] A. M. Seid, G. O. Boateng, S. Anokye, T. Kwantwi, G. Sun, and G. Liu, "Collaborative computation offloading and resource allocation in multi-uav-assisted iot networks: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12 203–12 218, 2021.

[53] S. Luo, H. Li, Z. Wen, B. Qian, G. Morgan, A. Longo, O. Rana, and R. Ranjan, "Blockchain-based task offloading in drone-aided mobile edge computing," *IEEE Network*, vol. 35, no. 1, pp. 124–129, 2021.

[54] Z. Guan, H. Lyu, D. Li, Y. Hei, and T. Wang, "Blockchain: A distributed solution to uav-enabled mobile edge computing," *IET Communications*, vol. 14, no. 15, pp. 2420–2426, 2020.

[55] Y. Zhan, C. H. Liu, Y. Zhao, J. Zhang, and J. Tang, "Free market of multi-leader multi-follower mobile crowdsensing: An incentive mechanism design by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 19, no. 10, pp. 2316–2329, 2020.

[56] T. D. Tran and L. B. Le, "Stackelberg game approach for wireless virtualization design in wireless networks," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[57] Z. Li, F. Jia, A. Mate, S. Jabbari, M. Chakraborty, M. Tambe, and Y. Vorobeychik, "Solving structured hierarchical games using differential backward induction," *arXiv preprint arXiv:2106.04663*, 2021.

[58] J. Cruz Jr, "Survey of nash and stackelberg equilibrim strategies in dynamic games," in *Annals of Economic and Social Measurement, Volume 4, number 2*. NBER, 1975, pp. 339–344.

[59] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6382–6393.

[60] Y. Wang, H. Liu, W. Zheng, Y. Xia, Y. Li, P. Chen, K. Guo, and H. Xie, "Multi-objective workflow scheduling with deep-q-network-based multi-agent reinforcement learning," *IEEE Access*, vol. 7, pp. 39 974–39 982, 2019.

[61] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.

Abegaz Mohammed Seid (Member, IEEE) received his B.Sc. and M.Sc. degrees in Computer Science from Ambo University and Addis Ababa University, Ethiopia, in 2010 and 2015, respectively. He received a Ph.D. degree in Computer Science and Technology from the University of Electronic Science and Technology of China (UESTC) in 2021. He is currently a post-doctoral fellow with the College of Science and Engineering at Hamad Bin Khalifa University, Doha, Qatar. He served as a graduate assistant and lecturer, as well as a member of the academic committee and an associate registrar at Dilla University, Ethiopia, from 2010 to 2016. Dr. Abegaz has published more than twenty one scientific conferences and journal papers. His research interests include a wireless network, mobile edge computing, blockchain, machine learning, Vehicular network, IoT, machine learning, UAV Network, IoT, and 5G/6G wireless network.

Aiman Erbad (aerbad@ieee.org) is an Associate Professor and ICT Division Head in the College of Science and Engineering, Hamad Bin Khalifa University (HBKU). Prior to this, he was an Associate Professor at the Computer Science and Engineering (CSE) Department and the Director of Research Planning and Development at Qatar University until May 2020. He also served as the Director of Research Support responsible for all grants and contracts (2016–2018) and as the Computer Engineering Program Coordinator (2014–2016). Dr. Erbad obtained a Ph.D. in Computer Science from the University of British Columbia (Canada) in 2012, a Master of Computer Science in embedded systems and robotics from the University of Essex (UK) in 2005, and a B.Sc. in Computer Engineering from the University of Washington, Seattle in 2004. He received the Platinum award from H.H. The Emir Sheikh Tamim bin Hamad Al Thani at the Education Excellence Day 2013 (Ph.D. category). He also received the 2020 Best Research Paper Award from Computer Communications, the IWCMC 2019 Best Paper Award, and the IEEE CCWC 2017 Best Paper Award. His research received funding from the Qatar National Research Fund, and his research outcomes were published in respected international conferences and journals. He is an editor for KSII Transactions on Internet and Information Systems, an editor for the International Journal of Sensor Networks (IJSNet), and a guest editor for IEEE Network. He also served as a Program Chair of the International Wireless Communications Mobile Computing Conference (IWCMC 2019), as a Publicity chair of the ACM MoVid Workshop 2015, as a Local Arrangement Chair of NOSSDAV 2011, and as a Technical Program Committee (TPC) member in various IEEE and ACM international conferences (Globecom, NOSSDAV, MMSys, ACMMM, IC2E, and ICNC). His research interests span cloud computing, edge intelligence, Internet of Things (IoT), private and secure networks, and multimedia systems. He is a senior member of IEEE and ACM.

Hayla Nahom Abishu received his B.Sc. degree in Computer Science and Information Technology from Haramaya University in 2007 and M.Sc. degree in Computer Science and Networking from Dilla University in 2017, Ethiopia. He is currently studying PhD degree in Computer Science and Technology in University of Electronic Science and Technology of China (UESTC). He is also a member of the Mobile Cloud-Network Research Team, UESTC. His research interests include Mobile Computing, wireless network, Blockchain, UAV Network, IoT, Network security and Machine Learning.

Abdullatif Albaseer (Member, IEEE) is a Post-doctoral Research Fellow at the Smart Cities and IoT Lab, Hamad Bin Khalifa University, Doha, Qatar. He received an M.Sc. in Computer Networks from King Fahd University of Petroleum & Minerals (KFUPM), Dhahran, Saudi Arabia, in 2017, and a Ph.D. in computer science and engineering from Hamad Bin Khalifa University (HBKU), Doha, Qatar, in 2022. His current research interests include Federated Learning over Wireless Network Edge, IoT, Smart cities, and cybersecurity in smart grid. Dr. Albaseer published more than fifteen conference and journal papers in ICC, Globecom, and IEEE transactions and invented and invented five patents in the area of federated learning and wireless network edge.

Mohamed Abdallah received his B.Sc. degree from Cairo University, Giza, Egypt, in 1996, and his M.Sc. and Ph.D. degrees from the University of Maryland at College Park, College Park, MD, USA, in 2001 and 2006, respectively.,From 2006 to 2016, he held academic and research positions with Cairo University and Texas A & M University in Qatar, Doha, Qatar. He is currently a Founding Faculty Member with the rank of Associate Professor with the College of Science and Engineering, Hamad Bin Khalifa University, Doha. He has published more than 150 journals and conferences and four book chapters and co-invented four patents. His current research interests include wireless networks, wireless security, smart grids, optical wireless communication, and blockchain applications for emerging networks. He is a recipient of the Research Fellow Excellence Award at Texas A& M University in Qatar in 2016, the Best Paper Award in multiple IEEE conferences, including IEEE BlackSeaCom 2019 and the IEEE First Workshop on Smart Grid and Renewable Energy in 2015, and the Nortel Networks Industrial Fellowship for five consecutive years, 1999–2003. His professional activities include an Associate Editor of the IEEE Transactions on Communications and the IEEE Open Access Journal of Communications, the Track Co-Chair of the IEEE VTC Fall 2019 Conference, the Technical Program Chair of the 10th International Conference on Cognitive Radio-Oriented Wireless Networks, and a technical program committee member of several major IEEE conferences.

Mohsen Guizani (S'85–M'89–SM'99–F'09) received the B.S. (with distinction) and M.S. degrees in electrical engineering, the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a professor and as appointed associate provost for faculty affairs and institutional advancement at Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), United Arab Emirates (UAE). Previously, he served in different academic and administrative positions at the University of Idaho, Western Michigan University, University of West Florida, University of Missouri-Kansas City, University of Colorado-Boulder, and Syracuse University. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He is currently the Editor in-Chief of the IEEE Network Magazine, serves on the editorial boards of several international technical journals and the Founder and Editor-in-Chief of Wireless Communications and Mobile Computing journal (Wiley). He is the author of nine books and more than 600 publications in refereed journals and conferences. He guest edited a number of special issues in IEEE journals and magazines. He also served as a member, Chair, and General Chair of a number of international conferences. Throughout his career, he received three teaching awards and four research awards. He is the recipient of the 2017 IEEE Communications Society Wireless Technical Committee (WTC) Recognition Award, the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks and the 2019 IEEE Communications and Information Security Technical Recognition (CISTC) Award for outstanding contributions to the technological advancement of security. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer. He is a Fellow of IEEE and a Senior Member of ACM.