Received 19 January 2024; revised 27 February 2024; accepted 6 March 2024. Date of publication 12 March 2024; date of current version 2 April 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3376335

Resilient Disaster Relief in Industrial IoT: UAV Trajectory Design and Resource Allocation in 6G Non-Terrestrial Networks

AMIR MOHAMMADISARAB[1], ALI NOURUZI[1], ATA KHALILI[2] (Member, IEEE), NADER MOKARI[1] (Senior Member, IEEE), BIJAN ABBASI ARAND[1] (Senior Member, IEEE), AND EDUARD A. JORSWIECK[3] (Fellow, IEEE)

[1]Department of Electrical and Computer Engineering, Tarbiat Modares University, Tehran 14115-111, Iran

[2]Institute for Digital Communications, Friedrich-Alexander-University Erlangen–Nurnberg, 91054 Erlangen, Germany

[3]Institute of Communications Technology, TU Braunschweig, 38106 Braunschweig, Germany

CORRESPONDING AUTHOR: N. MOKARI (e-mail: nader.mokari@modares.ac.ir)

The work of Eduard A. Jorswieck was supported in part by the Federal Ministry of Education and Research Germany (BMBF) as part of the 6G Research and Innovation Cluster (6G-RIC) under Grant 16KISK031.

It is noteworthy that part of this work was published in the 2022 IEEE Conference on Standards for Communications and Networking (CSCN) [1] [DOI: 10.1109/CSCN57023.2022.10050976].

ABSTRACT This research explores strategies to augment the connectivity among users within Hierarchical Non-Terrestrial Networks (HNTNs) dedicated to Disaster Relief Services (DRS). The primary goal is to optimize radio resources, computing capacities, and the trajectories of Unmanned Aerial Vehicles (UAVs) at each time slot, aiming to maximize the number of satisfactory connections (NSCs). UAVs function as aerial base stations (ABSs), establishing links for reduced capability (RedCap) user equipment (UE) using power domain non-orthogonal multiple access (PD-NOMA). Given the potential inoperability of terrestrial networks during disasters, swift data transmission is critical for mission-critical (MC) UEs. Therefore, end-to-end (E2E) delay is a crucial quality of service (QoS) constraint. The proposed problem is solved using a multi-agent recurrent deterministic policy gradient (MARDPG) algorithm, where the ABSs collaborate to maximize the NSCs and determine their optimal policy by interacting with the environment. Additionally, a sharing experience module (SEM) is incorporated, enabling agents to encode actions and observations using long short-term memory (LSTM), allowing each agent to utilize the historical actions and observations of other agents. To demonstrate the superiority of MARDPG, three algorithmic benchmarks and four different system models are implemented. The numerical results demonstrate the impact of various parameters, such as the number of subcarriers, users, and the maximum tolerable E2E delay on the NSCs. Furthermore, different scenarios indicate that MARDPG outperforms the benchmarks, achieving approximately a 6 percent optimality gap and a 91 percent fairness for achievable rate among users.

INDEX TERMS HNTNs, DRS, UAVs, PD-NOMA, MARDPG.

I. INTRODUCTION

A. MOTIVATIONS AND STATE-OF-THE-ART

Wireless communication technologies have undergone significant advancements in recent years, emerging as a critical facilitator for upcoming consumer applications and services [2], [3]. The transition from the fourth-generation (4G) to the fifth generation leads to a categorizing of quality of services (QoSs), which are named ultra-reliable low-latency communications (URLLC), massive machine-type communication (mMTC), and enhanced mobile broadband (eMBB) [4], [5]. Accordingly, one of the mMTC use cases is the Internet of Things (IoT), which requires massive connections [6], [7]. Nonetheless, more different and precise QoSs are involved in the six-generation (6G) [8]. Indeed, new services are created by emerging applications. For example, it can be a mixture of conventional

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

QoSs, which is introduced in 3rd Generation Partnership Project (3GPP) Release 17 (Rel-17) [9], named reduced-capability (RedCap) user equipment (UE). A QoS category is needed for industrial applications under the RedCap use case. This category should take into account factors such as reliability, latency, massive connectivity, and guaranteed bit rate. Therefore, industrial IoT (IIoT) can be considered as one of the use cases of RedCap UEs [10]. Accordingly, IIoT devices monitoring industrial infrastructures, especially critical ones, require a delay and connection-aware service [11]. This means that radio and computational resources are required to gurantee the associated services. Furthermore, preparing coverage and local task processing could be challenging and interesting due to some unique characteristics of required IIoT services, such as being time-sensitive and requiring massive connection [12].

Airborne communications will constitute a fundamental element of the 6G architecture, playing a pivotal role in the evolution of wireless communications. Their significance lies in the distinctive attributes they possess, particularly the capacity to extend coverage and reestablish connectivity in Disaster Relief (DR) scenarios [13]. Specifically, DR is a significant situation that requires innovative solutions [14]. Traditional terrestrial networks may not be functional during catastrophic events like earthquakes, floods, and other non-natural causes of disasters. As a result, DR paradigms are being developed that do not rely on these networks [14]. Therefore, non-terrestrial networks (NTNs) can serve as a viable option to provide coverage in disaster-stricken areas. The 3rd Generation Partnership Project (3GPP) Release 17 introduces new network structures for NTN, including high-altitude platforms (HAPS) and low earth orbit (LEO) satellites [10]. The elevated altitude of these systems results in considerable path loss and extended round-trip time (RTT) [10]. Thus, a hierarchical non-terrestrial network (HNTN) that operates from lower to higher altitudes emerges as a promising structure to address these challenges effectively. Unmanned aerial vehicles (UAVs), commonly known as drones, offer several critical potential wireless communication applications with intrinsic characteristics such as mobility and adaptive altitude [15]. In wireless networks, UAVs can be used as aerial base stations (ABSs) to improve reliability, capacity, coverage, and energy efficiency [16], [17]. UAVs are used as enablers for various applications consisting of military, surveillance, rescue, telecommunications and medical equipment delivery [18], [19], [20], [21]. Various HNTN structures can be exploited according to the characteristics of the corresponding environment and use case. Emerging technologies like Mobile Edge Computing (MEC) [22], [23], [24], [25] enable partial or complete processing of user data. Furthermore, the integration of collaborative data computing (CDC), combining MEC [26], fog [27], and cloud computing in diverse configurations, presents a promising approach to address prevalent data processing challenges in wireless communication.

The mismatch between mathematical tractability and the exponential complexity of wireless networks makes traditional convex optimization approaches inefficient and incapable of meeting the precise QoS requirements of emerging applications [28]. To address this issue, machine learning (ML) has emerged as a key enabler to manage high complexity for real-time implementation. In this context, deep Reinforcement Learning (DRL) has been investigated for comprehensive inputs as well as more accurate results. The use of deep reinforcement learning (DRL) algorithms to solve optimization problems is increasingly controversial [29]. Some claimed that these solutions could not meet standards such as solid mathematical proofs of convergence and suboptimal results compared to exact methods. Others, however, argued that despite these weaknesses, they still have some advantages [30]. In particular, they can be used to solve large, NP-hard, online problems that are intractable using standard algorithms [31]. Nevertheless, they should be personalized by taking into account the characteristics of each environment, the corresponding domain knowledge, and problem instead of implementing the typical algorithm [32].

B. RELATED WORKS

The discussion of existing work related to our work is included in this section. It is noticeable that related work can be divided into three parts in terms of the areas involved.

1) NTN

Minimizing the age of information (AoI) in two subsequent stages for an integrated terrestrial network (TN) and LEO is investigated in [33]. In addition, non-orthogonal multiple access (NOMA) is utilized in TN, and multiple satellites supply orthogonal access for other users for status updates. In [34], a solution is proposed to address the challenges of using traditional terrestrial cellular technologies for time-sensitive Internet of Things (IoT) applications. The solution involves a combination of mobile edge computing (MEC) and non-terrestrial networks (NTN), specifically unmanned aerial vehicles (UAVs) and satellites. The aim is to minimize latency in the complex propagation scheme. Similarly, in [35], the authors suggest using UAV-based cloud services for IoT nodes with a partially offloading scenario. They focus on optimizing energy efficiency by reducing the number of drones and minimizing costs for ground nodes while maintaining quality of service (QoS) requirements. Reference [36] analyzes data quantities and packet loss rates by applying a Markov chain to illustrate the data collection's reliability and using AoI to represent the data freshness.

2) IIOT

When a catastrophe takes place, it can severely disrupt critical infrastructures like communication networks, leading to significant problems and inconveniences. In order to address these challenges, a study conducted by researchers in [37] investigates the utilization of UAV-based telecommunications infrastructure to fulfill the communication requirements of

IoT nodes during natural emergencies like earthquakes. The study aims to optimize various aspects such as UAV trajectory, mission completion time, and energy consumption simultaneously. Moreover, in rural areas and distinct environments, mobile base stations are utilized as relays and data collectors to accommodate the growing number of IoT devices. Achieving lower delays between components in IIoT is one of the important requirements. The authors of [38], take into account the effect of constantly changing environmental factors over time on data flow scheduling for large-scale IIoT networks. They propose distributed optimal transport (OT) algorithm to optimize the scheduling problem. To enhance the safety and accuracy of IIoT networks, controlling delay should be considered. A uRLLC-based service with joint communication and delay control is investigated in [39], in which the optimal block length of codewords is selected for controlling IIoT devices' delay. In [40], device-to-device (D2D) link scheduling is studied in UAV-aided IIoT networks, where robustness and low complexity are considered. First, the geographical map of transmission links is used as input to a sparse convolutional neural network (SCNN). Second, in order to maximize the achievable rate and optimize the D2D scheduling, the output feature map from the SCNN is processed by a deep deterministic policy gradient-based reinforcement learning model. Joint trajectory design and data offloading in UAV-assisted IIoT networks are studied in [41]. The authors use a Bernstein-type inequality to reformulate the constraints in the energy minimization problem and decompose two different subproblems.

3) ARTIFICIAL INTELLIGENCE ACROSS WIRELESS COMMUNICATION

In [42], multi-agent deep reinforcement learning (MADRL) is utilized to tackle the difficulty of integrated UAV and LEO, such as the trajectory of UAVs and LEO real-time orbiting via the NTN. The main problem considered is how to forward data between two distant ground terminals through SAT and UAV relays to enhance efficiency. The uplink delay of mission-critical users concerning keeping fairness among them is studied in [43]. The author mixed long-short term memory (LSTM) with Q-learning to solve the proposed problem, which achieved higher performance than the tabular Q-learning approach and Round Robin algorithms. In [44], the authors investigate the maximization of the uplink sum rate for NOMA-assisted IoT networks by adopting a non-static method for clustering users. Furthermore, they proposed a solution that combines the DRL algorithm and SARSA, demonstrating better performance and lower complexity compared to conventional DRL approaches.

C. NOVELTY AND CONTRIBUTIONS

The previous subsection explains the existing closest work. However, there is a need for improvement in resource allocation techniques, specifically for RedCap UEs, which are the IIoT users discussed in this paper. These techniques should incorporate a new QoS category that takes into account reliability, latency, massive connectivity, and guaranteed bit rate based on the specific requirements of each use case. For instance, in disaster-stricken areas, IIoT nodes require DR services that offer both massive connectivity and end-to-end latency assurance. To the best of the authors' knowledge, the literature has not yet addressed the problem of maximizing the number of satisfactory connections (NSCs) in terms of end-to-end latency. The conventional works are based on two common models. The first is total or average delay minimization, which was studied in [43], [45]. Secondly, minimizing the maximum delay is considered in [23], [24], [25]. However, they did not perform well because it is still possible for some users to have an E2E delay greater than the threshold. This can exacerbate the catastrophic situation. Apart from that, there is still an open question as to which type of computing method is better, solid MEC or collaborative computing (COC). This paper bridges the existing gaps by focusing on MEC-assisted joint resource allocation, task processing, and trajectory design. We consider the constraints of end-to-end delay and computational resources to maximize the number of satisfactory connections (NSCs). The main contributions of this paper are summarized as follows:

- To ensure satisfactory quality of service (QoS) for all users, it is important to guarantee end-to-end (E2E) delay for all covered users. Simply minimizing the average or maximum delay of users does not guarantee that the QoS will be satisfactory for all users. There is still a possibility that the latency experienced by some users may be unacceptable and exceed the maximum tolerable delay. To address this issue, we propose maximizing the number of satisfactory connections (NSCs) in terms of E2E delay, ensuring that all established connections meet the desired QoS.
- To cover the disaster area, we used a temporary solution by utilizing a hierarchical network of UAVs and HAPS as an ABS and relay, respectively. As far as we know, this approach of using such a structure to provide coverage for the RedCap UEs in the disaster area is both relevant and innovative.
- Instead of using algorithms that are centralized and rely on a single agent, which can lead to increased signaling overhead, we employ a decentralized approach using a multi-agent recurrent deterministic policy gradient (MARDPG) technique. As far as we know, this is the first time MARDPG has been applied in this particular area. Additionally, to improve collaboration among agents, we utilize a sharing experience module (SEU) that encodes each agent's observations and actions and facilitates their exchange using long-short term memory (LSTM). The optimality gap, which measures the difference between the best-known solution and the best bound, is a key criterion for evaluating algorithm performance. To determine the optimality gap

of MARDPG, we employ the Exhaustive Search (ES) algorithm. Thanks to its outstanding performance and lower complexity, MARDPG achieves an optimality gap of 6.67%, which can be justified regarding its performance.

- The simulation outcomes indicate that MARDPG outperforms other algorithms such as multi-agent deep deterministic policy gradient (MADDPG), distributed soft actor-critic (DSAC), and Greedy. Additionally, the results suggest that maximizing NSCs is a more effective approach than minimizing average delay or minimizing maximum delay. We also examined the impact of increasing the number of subcarriers, users, and maximum tolerable E2E delay, and found that our proposed method obtained more satisfying connections compared to minimizing average delay and maximum delay.
- The simulation results also show that MEC performs better than COC (MEC and Fog) when most of the data is processed in MEC. As the proportion of data offloaded to fog increases, the performance of COC increases dramatically due to the higher computational resources available to fog.

It is noteworthy that a portion of this work was previously published in the 2022 IEEE Conference on Standards for Communications and Networking (CSCN) [1]. The differences between the conference version and this iteration are outlined below. Firstly, in the system model, we incorporated a precise energy model for UAV movement and imposed constraints on UAV velocity and energy consumption. Additionally, we introduced UAV velocity optimization, treating velocity as an optimization variable. Furthermore, we included the computational resources allocated in the prior time slot as a state, extending beyond the Context Information Service (CIS). The analysis of signaling overhead and computational complexity of the proposed algorithm is of high importance, and we have included it in this version. In the simulation, we introduced DSAC and Greedy as algorithmic benchmarks, alongside MADDPG. Subsequently, we added a comparison between Mobile Edge Computing (MEC) and Collaborative Computing (COC). The examination of the optimality gap of the proposed algorithm is crucial, and we addressed it in this version by implementing the exhaustive search algorithm. Lastly, we computed the fairness score among users in terms of transmission rate in this version.

D. PAPER ORGANIZATION

The rest of this paper is organized into five parts as follows: In Section II, we explain the system model. Next, the proposed solution is explained in Section III, and the convergence is analyzed in Section III-A of this Section. After that, the computational complexity and signaling overhead are studied in Section IV. In addition, simulation results are illustrated in Section V, which consists of seven scenarios. Indeed, the simulation environment, comparison between different DRL algorithms, comparison between different objective functions, the performance of COC, trajectory design, optimality gap and exhaustive search, and fairness analysis are explained in Sections V-A–V-G, respectively. Eventually, the conclusion and future works are expressed in Section VI.

Symbol Notations: To denote the vector A and the i-th element of this vector, we use the bold upper symbol as $\mathbf{A}$ and a_i, respectively. Likewise, $a_{i,j}$ define the i,j element of matrix $\mathbf{A}$. In addition, $\mathcal{B}$ and b_j denote the set B and the j-th element of it, respectively. We use $|c|$ to define the absolute value of c. To define the expectation of d, we use $\mathbb{E}[d]$. Also, we define the norm of vector X, as $\|X\| = \sqrt{\sum_{i=1}^{n} |x_i|^2}$.

II. SYSTEM MODEL

In this paper we consider the HNTN that supports IIoT nodes spread out through the disaster area in the uplink scenario, consisting of UAVs and monitoring IIoT nodes as users as shown in Fig. 1. We have two types of NTN components: the first is considered as an aerial base station (ABS) to provide links and MEC for nodes. Secondly, a HAPS is a communication component that relays the messages between ABSs in the disaster area and outside. Let $\mathcal{U} = \{1, \ldots, u, \ldots, U\}$ denote the set of IIoT nodes where U shows the total number of nodes. Furthermore, $\mathcal{B} = \{1, \ldots, b, \ldots, B\}$ is considered as the set of ABSs where B represents the total number of ABSs and a denotes HAPS. In particular, the total operating time is 100 s, which is divided into 2000 slots. The time slot is represented by t, and its length is 50 ms, denoted by Δ. Moreover, the structure of each timeslot is depicted in Fig. 2. It is noteworthy that Δ is the length of the time slot, and δ is the length of the traveling (trajectory) part. First of all, the CSIs are gathered. Then, the problem is solved. Eventually, UAVs start to move to the next location. It is noticeable that during CSI gathering and problem-solving, UAVs hover. For CSI gathering, we assumed that IIoT nodes send known pilot signals to UAVs, and UAVs use these known signals to estimate the channel response and calculate the CSI. The resource allocation and the role of each component are provided as follows [46],

- IIoT nodes: They gathered the required information from the environment. They request to connect to ABS and send their data.
- UAVs: They have the aerial base station roles of providing communication resources (power and subcarrier) and computational resources (CPU cycle) for IIoT nodes. They also move to new locations in each time slot to find the best location (trajectory design) [47].
- HAPS: In the main scenario of the paper in **Section II**, it only has the role of the relay to receive controlling data from the UAV and send it to the outside. We did not consider any resources for it in this **Section** [47]. However, the reason for considering HAPS in our paper is to investigate and compare two different computing models, MEC and collaborative computing (MEC and

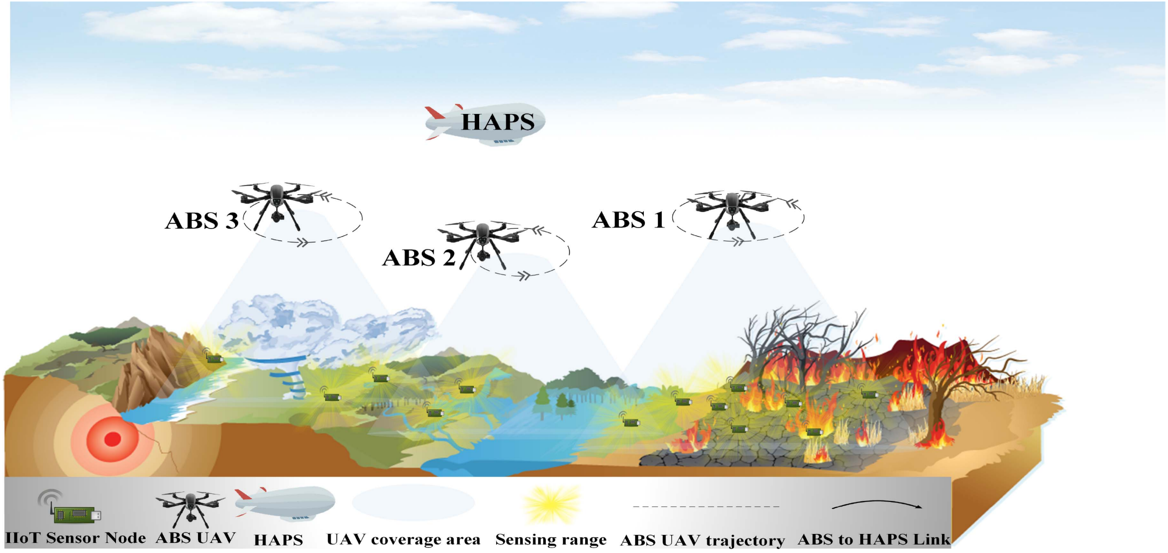


FIGURE 1. Proposed System Model with IIoT Nodes Throughout Disaster Area, UAVs Layer as MECs and ABSs, and HAPS Layer as Fog. Disasters can be earthquake, tornado, flooding, and fire, respectively.

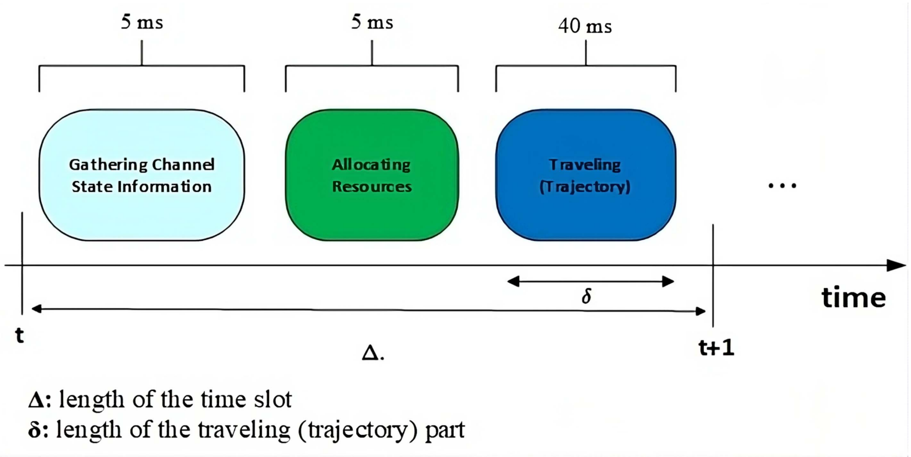

5 ms
5 ms
40 ms
Gathering Channel State Information
Allocating Resources
Traveling (Trajectory)
...
δ
time
t
Δ.
t+1
Δ: length of the time slot
δ: length of the traveling (trajectory) part

FIGURE 2. Frame structure with three phases: gathering channel state information, allocating resources, and trajectory travel.

Fog). To this end, we considered computational computing for HAPS in **Section V-D**. Indeed, HAPS has computational resources, and a portion of IIoT nodes' data are offloaded to it for processing [47].

Let us consider the locations of IIoT nodes, ABSs, and HAPS as $\boldsymbol{l}_u = [x_u, y_u, z_u]$, $\boldsymbol{l}_b^t = [x_b^t, y_b^t, z_b^t]$, and $\boldsymbol{l}_a^t = [x_a^t, y_a^t, z_a^t]$, respectively. Therefore, the distance between node u and ABS b and ABS b and a by using a 3D Cartesian coordinate system in unit of $[m]$ can be written as:

$$
\begin{aligned}
d_{b,u}^t &= \|\boldsymbol{l}_b^t - \boldsymbol{l}_u\| = \sqrt{\left(x_b^t - x_u\right)^2 + \left(y_b^t - y_u\right)^2 + \left(z_b^t - z_u\right)^2}, \\
&\quad \forall u \in \mathcal{U}, \forall b \in \mathcal{B}, \forall t, \\
d_{b,a}^t &= \|\boldsymbol{l}_b^t - \boldsymbol{l}_a^t\| = \sqrt{\left(x_b^t - x_a^t\right)^2 + \left(y_b^t - y_a^t\right)^2 + \left(z_b^t - z_a^t\right)^2}, \\
&\quad \forall b \in \mathcal{B}, \forall t.
\end{aligned} \tag{1}
$$

The computation task of node u is represented by $\digamma_u$, which consists of DL_u bits. For processing each bit of this task, Ω_u CPU cycles are needed [1], [48]. PD-NOMA allows more users to be served with linked resources compared to other multiple access (MA) techniques. Accordingly, bandwidth W is divided into several orthogonal subcarriers that can be represented by $\mathcal{K} = \{1, \ldots, k, \ldots, K\}$ where K is the total number of subcarriers and the bandwidth of each one is $\hat{W} = W/K$. We define the binary variable $\psi_{u,b,k}^t \in \{0, 1\}$ to indicate that subcarrier k is allocated to node u in time slot t or not. $g_{u,b,k}^t$ and $p_{u,b,k}^t$ denote the Rayleigh fading channel gain with the reference-distance unit power gain 1.4×10^{-4} and transmission power on the $k-$th subcarrier from node u to ABS b at time slot t, respectively. Consequently, the average signal-to-interference-plus-noise ratio (SINR) for node u and ABS b on the k-th subcarrier is expressed as [1], [49],

$$\gamma^t_{u,b,k} = \frac{p^t_{u,b,k}\left(d^t_{b,u}\right)^{-\zeta} g^t_{u,b,k}}{I^{\text{intra},t}_{u,b,k}+I^{\text{inter},t}_{u,b,k}+\sigma^2}\left(\frac{\Pr^{\text{LoS}}_{u,b,k}}{\eta^{\text{LoS}}}+\frac{1-\Pr^{\text{LoS}}_{u,b,k}}{\eta^{\text{NLoS}}}\right),$$
$$\forall u\in\mathcal{U},\forall b\in\mathcal{B},\forall k\in\mathcal{K},\forall t, \qquad (2)$$

where ζ is the path loss exponent coefficient, σ^2 denotes the receiver noise variance at the ABS, and $\eta^{\text{LoS}}(\eta^{\text{NLoS}})$ is the (non-) line of sight excessive path loss value. $\Pr^{\text{NLoS}}_{u,b,k}$ and $\Pr^{\text{LoS}}_{u,b,k}$ are the NLoS and LoS probabilities according to

$$\Pr^{\text{LoS, t}}_{u,b,k} = \frac{1}{1+\alpha\exp\left(-\beta\left[\arcsin\left(z^t_b/d^t_{b,u}\right)-\alpha\right]\right)}, \qquad (3)$$
$$\forall u\in\mathcal{U},\forall b\in\mathcal{B},\forall k\in\mathcal{K},\forall t,$$

$$\Pr^{\text{NLoS, t}}_{u,b,k} = 1-\Pr^{\text{LoS}}_{u,b,k}, \forall u\in\mathcal{U},\forall b\in\mathcal{B},\forall k\in\mathcal{K},\forall t, \quad (4)$$

where factors α and β are determined based on the conditions of the environment as an example, $\alpha = 4.88$ and $\beta = 0.43$ for the suburban ecosystem [49]. Also, the intra-cell interference can be calculated by,

$$I^{\text{intra},t}_{u,b,k} = \sum_{j\in\mathcal{J}_{b,c}} \psi^t_{j,b,k}p^t_{j,b,k}g^t_{j,b,k}, \forall u\in\mathcal{U},\forall b\in\mathcal{B},\forall k\in\mathcal{K},\forall t, \ (5)$$

where $\mathcal{J}_{b,c} = \{j \mid j\in\mathcal{J}_{b,c}, \gamma_{u,b,k} > \gamma_{j,b,k}\}$ represents the set of users that are served by ABS b and contains the users within $\mathcal{J}_{b,c}$ with worse channels. The ABS decodes user messages using successive interference cancellations (SIC) [1], [50]. At the ABS, decoding is always conducted from the node with a better channel quality and SINR to the node with a worse channel quality and SINR, or else a significant amount of power will be required by the node with a worse channel quality and SINR to offset the path loss [1], [51]. In addition, inter-group interference can be characterized by

$$I^{\text{inter},t}_{u,b,k} = \sum_{\substack{j=1,\\ j\neq b}}^{B}\sum_{o=1}^{U} \psi^t_{o,j,k}p^t_{o,j,k}g^t_{o,j,k}\forall u\in\mathcal{U},\forall b\in\mathcal{B},\forall k\in\mathcal{K},\forall t.$$
$$(6)$$

As a result, the transmission rate between node u and ABS b is computed by

$$R^t_{u,b,k} = \hat{W}\psi^t_{u,b,k}\log_2\left(1+\gamma^t_{u,b,k}\right), \forall u\in\mathcal{U},\forall b\in\mathcal{B},$$
$$\forall k\in\mathcal{K},\forall t. \qquad (7)$$

The ABS moves to the new locations with constant velocity v_b by $\boldsymbol{l}^{t+1}_b = \boldsymbol{l}^t_b + \tilde{l}^t_b, \tilde{l}^t_b \in \mathcal{MA}$, where $\tilde{l}^t_b = v^t_b\delta$ in $[m]$ and δ in $[s]$ denote the UAV's distance and travel time within a given time slot t, respectively. Then, $\mathcal{MA}$ represents the set of all possible drone movements that consists of all hovering and all other movements in the direction of x, y, z that can be expressed as follows:

$$\mathcal{MA} = \left\{\underbrace{\begin{bmatrix}0\\0\\0\end{bmatrix}^T}_{\text{hover}}, \underbrace{\begin{bmatrix}\tilde{l}^t_b\\0\\0\end{bmatrix}^T}_{\text{+x-axis}}, \underbrace{\begin{bmatrix}-\tilde{l}^t_b\\0\\0\end{bmatrix}^T}_{\text{-x-axis}}, \underbrace{\begin{bmatrix}0\\\tilde{l}^t_b\\0\end{bmatrix}^T}_{\text{+y-axis}}, \underbrace{\begin{bmatrix}0\\-\tilde{l}^t_b\\0\end{bmatrix}^T}_{\text{-y-axis}}, \underbrace{\begin{bmatrix}0\\0\\\tilde{l}^t_b\end{bmatrix}^T}_{\text{+z-axis}}, \underbrace{\begin{bmatrix}0\\0\\-\tilde{l}^t_b\end{bmatrix}^T}_{\text{-z-axis}}\right\}. \qquad (8)$$

where $\tilde{l}^t_b \in \mathcal{R}$. Power consumption by the UAVs is influenced by kinetic energy and telecommunication power [52]. While the power consumed by receiving and sending messages is negligible compared to the energy used for moving [52]. We consider the movement of ABS with constant velocity v_b for all actions. Based on the tasks that are performed, the ABS consumption model has two components: movement and processing as $E^{\text{mov},t}_b$and $E^{\text{proc},t}_b$, respectively. According to [53], at a speed of v^t_b, the power consumption of the ABS can be calculated as follows:

$$P^{\text{kinetic},t}_b\left(v^t_b\right) = P_0\left(1+\frac{3|v^t_b|^2}{M^2_{\text{tip}}}\right)+\frac{1}{2}d_0\varrho\vartheta A_{\text{rotor}}|v^t_b|^3$$
$$+P_i\sqrt{\sqrt{1+\frac{|v^t_b|^4}{4v^4_0}}-\frac{|v^t_b|^2}{2v^2_0}}, \qquad (9)$$

where P_0 and P_i represent the vane profile power and the hovering power, consecutively. d_0, ϱ, M_{tip}, v_0, ϑ, $|v^t_b|$, and A_{rotor} denote the fuselage drag ratio, the air density, the tip speed of the rotor vane, the mean rotor induced velocity in hovering status, rotor solidity, the speed vector size of ABS, and the rotor disc area, respectively [53]. Next, movement requires the following amount of energy:

$$E^{\text{mov},t}_b = P^{\text{kinetic},t}_b\cdot\delta, \qquad (10)$$

where δ is the traveling time (a part of the time slot that the UAV moves from one point to another. Please look at Fig. 2) with the speed of v^t_b between two time slots. Additionally, the processing energy is

$$E^{\text{proc},t}_b = \kappa_b\cdot\left(f^t_b\right)^3, \qquad (11)$$

here κ_b is CPU switched capacitance of ABS b and f^t_b indicates ABS u frequency at time slot t. Therefore, we can define the total energy consumption of ABS as $E^{\text{Total},t}_b = E^{\text{mov},t}_b + E^{\text{proc},t}_b$. The nodes offload their data to ABSs for processing. Due to the significance of this information for proactively relieving the disaster, the end-to-end (E2E) delay imposed on each node's data should be taken into account. Therefore, the E2E delay consists of transmission, queueing, and processing delays in our system. It is worth noting that we do not consider propagation delay because of a negligible amount compared to other types. The transmission delay between node u and ABS b is given by,

$$D^{\text{tran},t}_{u,b} = \frac{DL_u}{\sum^K_{k=1}R^t_{u,b,k}}, \forall u\in\mathcal{U},\forall b\in\mathcal{B},\forall t. \qquad (12)$$

We consider a buffer with limited capacity for each ABS. An M/M/1 queuing system is a single-queue single-server queuing system. As each user can also link with one ABS (as a MEC server), M/M/1 model is appropriate to be used for the queueing delay [1], [54]. Also, the packet (task) arrival rate based on Poisson process at time slot t for ABS b is DL^t_b in packet per second ([pps]). Also, μ_b is the service (computational) rate of ABS b in [pps]. Thus, the average queuing delay in [s] (as we consider each user has one packet, the unit of it is second) for node u's data is given by,

$$D_{u,b}^{\text{queu},t} = \frac{1}{\mu_b - DL_b^t}, \forall u \in \mathcal{U}, \forall b \in \mathcal{B}, \forall t. \tag{13}$$

In ABS b, the processing delay that is associated with computational tasks on data node u can be calculated by

$$D_{F_u,b}^{\text{proc},t} = \frac{(1-\varepsilon)DL_u\Omega_u}{f_{F_u,b}^t}, \forall u \in \mathcal{U}, \forall b \in \mathcal{B}, \forall t, \tag{14}$$

here $f_{F_u,b}^t$ indicates the computational resources that are allocated to task F_u that is pertained to node u data which is provided by ABS b in (CPU cycle/s). Additionally, ε represents the amount of data offloaded to the next computing layer, such as fog, whereas in this paper, we only consider the MEC layer. Therefore, $\varepsilon = 0$ means that all data are computed in MEC (as we mentioned, COC (MEC and fog) is implemented as one of our benchmarks, and ε is defined for simplicity in V-D). Now, the E2E delay can be calculated by,

$$D_u^{\text{E2E},t} = D_{u,b}^{\text{tran},t} + D_{u,b}^{\text{queu},t} + D_{F_u,b}^{\text{proc},t}, \forall u \in \mathcal{U}, \forall b \in \mathcal{B}, \forall t. \tag{15}$$

Under critical circumstances, particularly when infrastructures are monitored by IIoT nodes, fast detection of the system's fault is essential for proactively preventing or relieving the disaster, which motivates us to consider maximizing the NSCs. To this end we introduce $\chi_{u,b}^t$ as an indicator of NSC which shows that the node u under coverage of ABS b receives at least one subcarrier ($\psi_{u,b,k}^t = 1, \forall k \in \mathcal{K}$), and the E2E delay of this node is less than the threshold. The NSCs maximization problem can be expressed as follows for all t

$$\mathcal{OP}: \max_{\mathbf{P},\mathbf{\Psi},\mathbf{L},\mathbf{F},\mathbf{V}} \Lambda = \sum_{b=1}^{B}\sum_{u=1}^{U} \chi_{u,b}^t \tag{16a}$$

$$\text{s.t.} \quad \text{C1}: \|\boldsymbol{l}_b^{t+1} - \boldsymbol{l}_b^t\| < v_b^t\delta^t, \forall b \in \mathcal{B}, \tag{16b}$$

$$\text{C2}: d^{\min} < d_{b,\hat{b}}^t, \forall b, \hat{b} \in \mathcal{B}, \tag{16c}$$

$$\text{C3}: z^{\min} < z_b^t < z^{\max}, \forall b \in \mathcal{B}, \tag{16d}$$

$$\text{C4}: \sum_{u=1}^{U} \psi_{u,b,k}^t \leq C^{\text{Th}}, \forall b \in \mathcal{B}, \forall k \in \mathcal{K}, \tag{16e}$$

$$\text{C5}: p^{\min} \leq \sum_{b=1}^{B}\sum_{k=1}^{K} \psi_{u,b,k}^t p_{u,b,k}^t \leq p^{\max}, \forall u \in \mathcal{U}, \tag{16f}$$

$$\text{C6}: \sum_{j=1}^{J} f_{F_j,b}^t \leq f_b^{\max}, \forall b \in \mathcal{B} \tag{16g}$$

$$\text{C7}: D_u^{\text{E2E},t} \leq D^{\text{Th}}, \tag{16h}$$

$$\text{C8}: \psi_{u,b,k}^t \in \{0,1\}, \forall u \in \mathcal{U}, \forall b \in \mathcal{B}, \forall k \in \mathcal{K}, \tag{16i}$$

$$\text{C9}: \chi_{u,b}^t \in \{0,1\}, \forall u \in \mathcal{U}, \forall b \in \mathcal{B}, \tag{16j}$$

$$\text{C10}: \sum_{b \in \mathcal{B}} \psi_{u,b,k}^t \leq 1, \tag{16k}$$

$$\text{C11}: \chi_{u,b}^t \leq \psi_{u,b,k}^t, \tag{16l}$$

$$\text{C12}: E_b^{\text{Total},t} < E^{\max}, \forall b \in \mathcal{B}, \forall t, \tag{16m}$$

$$\text{C13}: v_b^t < v^{max}, \forall b \in \mathcal{B}, \forall t, \tag{16n}$$

where $\mathbf{P}$ indicates the uplink power allocation and $[\mathbf{P}]_{u,b,k} = p_{u,b,k}^t$. Similarly, $[\mathbf{\Psi}]_{u,b,k} = \psi_{u,b,k}^t$ represents the uplink subcarrier allocation matrix. $[\mathbf{F}]_{u,b} = f_{F_u,b}^t$ is CPU cycle allocation matrix, and $[\mathbf{L}]_u = \mathbf{l}_b^t$ are the locations matrix of ABSs. In addition, the $\mathbf{V}$ indicates the matrix of all ABSs' velocity. Accordingly, (16b) ensures that the velocity of ABS b does not exceed the limits of UAVs in time slot t. To avoid the collision between the ABSs, the distances between them should be greater than the limit in (16c). Each type of UAV has a height limitation in (16d). (16e) indicates the subcarrier allocation for the PD-NOMA scheme which is restricted to C^{Th}. The practical limitation of IIoT node's battery size leads to limit the transmission power of each node between minimum $p_{\min}$ and maximum $p_{\min}$ that (16f) assures it. (16g) shows the computational resource of ABSs are limited, and the total CPU cycles that are allocated to the nodes' processing task should be less than the ABS maximum resource, i.e., $f_b^{\max}$. Also, (16h) ensures that the E2E delay of node u is less than the threshold (D^{Th}) to count it as a satisfying connection. Indeed, we ensure E2E delay instead of restricting each of the involved delays. Our intention is that if we put constraints on each delay component, ABS does not have enough flexibility to solve the problem. (16i) and (16j) indicate the subcarrier assignment and number of satisfactory connections indicators are binary variables. (16k) represents that each user is assigned only to one ABS. Finally, (16l) indicates that maximum value of $\chi_{u,b}^t$ is limited to $\psi_{u,b,k}^t$. Moreover, (16m) ensures that the energy consumption of ABS is always bounded and less than the maximum. Apart from that, the velocity of a UAV can not be any amount, and it should be limited due to the inherent restriction of flying. Thus, (16n) ensures that the velocity of the UAV at each time slot is less than the maximum.

III. PROPOSED ALGORITHM

Channel estimation is performed to obtain channel gains. However, it is impractical to move each UAV to all possible locations to acquire perfect CSI since such a sweeping search will consume significant power and time [1], [55]. Due to the necessity of an online algorithm for solving our formulated complex optimization problem in (16a), MARDPG is proposed to tackle this issue. As a modification of the Deterministic Policy Gradient (DPG) method, the MARDPG is intended to manage situations involving numerous interacting agents in a sequential decision-making setting [56]. Agents in multi-agent systems frequently have to cooperate to accomplish a shared or personal objective, and MARDPG tackles the difficulties these situations present. The main features of MARDPG are as follows

- *Recurrent Neural Networks (RNNs):* RNNs are frequently used in MARDPG's architecture. RNNs are employed to capture the temporal dependencies in each agent's observations and actions since they are well-suited for processing sequential data. This is especially crucial in situations where it is not possible to fully observe the state of the environment at any given time.
- *Deterministic Policy Gradient (DPG):* The DPG framework, a model-free, off-policy actor-critic method for continuous action spaces, provides the foundation for MARDPG. In contrast to stochastic policies, which are more difficult to optimise, deterministic policies, which map states to certain actions, are the main emphasis of DPG.
- *Multi-Agent Method:* MARDPG takes agent interactions into account in a multi-agent setting. Based on its own observations, actions, and maybe the observations and acts of other agents, each agent develops its own policy. The algorithm considers the influence of an action by one agent on the state observed by other agents as well as the joint action space.
- *Centralized Training, Decentralized Execution (CTDE):* MARDPG frequently adheres to the CTDE model. The algorithm may use centralised training during training, in which the learning process is enhanced by taking into account data from all agents. However, the necessity for communication during execution is minimised since, during execution, each agent takes decisions in a decentralised manner based on its local observations [57].
- *Collaboration:* We developed a sharing experience module (SEM), which enables agents to encode actions and observations using long short-term memory (LSTM), allowing each agent to utilise the historical actions and observations of other agents.

In this study, the ABSs are agents, and each takes action based on its policy. Agent b at time slot t observes state s^t, and takes the action a^t, correspondingly. The environment then shifts into the next state s^{t+1}, and the agent b receives a reward for choosing that action. Regarding our system model, the state space, action space, and reward function represented by $\mathcal{A}$, $\mathcal{S}$, and r^t, respectively, and are characterized as follows:

- **State:** The observed state by agent b (ABS b) at time slot t consists of the channel state information (CSI) between node u and ABS b, $g^t_{u,b,k} \forall k \in \mathcal{K}$ and the preliminary information about computational resources that are allocated to task $\mathcal{F}_u$ that is pertained to node u data, which is provided by ABS b, $f^{t-1}_{\mathcal{F}_u,b}$. Thus, the state can be characterized as follows
$$s^t_b = \left[g^t_{u,b,k}, f^{t-1}_{\mathcal{F}_u,b}\right], b \in \mathcal{B}, u \in \mathcal{U}, \forall t. \qquad (17)$$
For sake of simplicity, we denote the whole state matrix at time slot t with $\boldsymbol{S}^t$, which contains the states of all agents.
- **Action:** The uplink power allocation matrix, $\mathbf{P}^t_b$, uplink subcarrier allocation matrix, $\boldsymbol{\Psi}^t_b$, computational resource allocation matrix, $\mathbf{F}^t_b$, and location selection of ABS, $\mathbf{l}^t_b$ are the agent's b actions at time slot t. Hence, the action space of agent b at time slot t is $a^t_b = \{\mathbf{P}^t_b, \boldsymbol{\Psi}^t_b, \mathbf{F}^t_b, \mathbf{l}^t_b\}$. Moreover, we denote the whole action matrix with $\boldsymbol{A}^t$, which contains the actions of all agents.
- **Reward:** Regarding the multi-agent algorithm DRL, each agent (ABS) receives its rewards, r^t_b based on its action a^t_b, and the total reward is the sum of all agents' rewards. Specifically, if the agent does an appropriate action that satisfies constraints (16b), (16c), (16d), (16e), (16f), (16g), (16h), (16i), (16j), (16k), (16l), (16m) and (16n), it gets a positive reward, $r^t_b = \Lambda^t_b$. Otherwise, it receives a negative reward, $r^t_b = -10$, as a punishment to let it know that it made a mistake in choosing an action. Recall that the goal of the optimization problem is to maximize NSCs, Λ. Therefore, the total reward function is $r^t = \Lambda$. Accordingly, agents cooperatively try to maximize Λ. Indeed each agent maximizes the objective function via its covered nodes, $r^t_b = \Lambda^t_b$, and the total reward can be expressed by $r^t = \sum_{b=1}^{B} r^t_b$.

Regarding the deep deterministic policy gradient (DDPG) model [58], [59], the actor-critic model is used to develop our approach. Our model considers three components: a critic, actors, and experience sharing. Typical actor-networks model each agent using deterministic policy to map states to corresponding actions. In addition, the critic seeks to reveal the expected future rewards actions by assessing the action-value function. Note that, instead of each agent utilizing a history of its observations and actions, it can use others' observations and actions. For this purpose, the SEU is used to get all agents' observations and actions and encode them by exploiting long-short term memory (LSTM) [60]. Notably, in this multi-agent DRL, the state of the environment, s^t is shared between all agents. In contrast, the reward, $r^t = r(s^t, a^t_b)$, actions a^t_b, and observations, o^t_b, are all locally chosen by actors (agents), $\forall b \in \mathcal{B}$. Particularly, agent b takes action a^t_b due to its policy, $\varphi^t_b(s^t)$, and receives a reward $r^t_b = r(s^t, a^t_b)$. The action-value function at the critic is $Q^{\pi}(s^t, a^t_1, \ldots, a^t_B)$ that is utilized for calculating the total future reward and return. It is based on all agents' actions. Moreover, agent b observes the environment at time slot t, o^t_b. The state of the environment is an experience of observations and all agents' actions, $s^t = f(o^1, a^1, \ldots, a^{t-1}, o^t)$ based on partially observed Markov decision process (POMDP). As we mentioned, we consider the LSTM-based experience sharing encodes all previous observations and actions from the agents' whole system transactions history. Denote $\mathcal{H}^{t-1}$ as a system history vector that can be expressed by $\mathcal{H}^{t-1} = f_{\text{LSTM}}(\mathcal{H}^{t-2}, [o^{t-1}_b, a^{t-1}_b]; \phi)$ for all agents $b \in \mathcal{B}$. By using $\mathcal{H}^{t-1}$, the environment state can be rewritten $s_t \simeq \{\mathcal{H}^{t-1}, o^t_b\}$, $\forall b \in \mathcal{B}$. Recall that we have the continuous actions, agent's action can be defined as a vector of real values (we use semi-equal instead of equal because observations are not exact).

Since deterministic policy is used instead of stochastic policy, the actor's policy can be parameterized by ω_b, $\varphi_b^t(s^t; \omega_b)$, and each agent can exploit other ones actions and observations, $\mathcal{H}^{t-1}$. Considering these factors, the agent's action is, $a_b^t = \varphi_b^t(s^t) \simeq \varphi_b^t(\mathcal{H}^{t-1}, o_b^t; \omega_b)$. In the rest of paper, we express $o^t = \{o_1^t \dots o_B^t\}$ and $a^t = \{a_1^t \dots a_B^t\}$. Thus, the action-value and policy function can be rewritten as $Q^\pi(\mathcal{H}^{t-1}, a^t, o^t; \theta)$ and $\varphi_b^t(\mathcal{H}^{t-1}, o^t; \omega_b)$, respectively. According to the Bellman equation in Q-learning [61], the critic can be learned by minimizing the following loss,

$$\Delta(\theta) = \mathbb{E}_{o^t, \mathcal{H}^{t-1}}\left[\left(Q^\pi\left(\mathcal{H}^{t-1}, a^t, o^t; \theta\right) - \Upsilon^t\right)^2\right], \quad (18)$$

where $\Upsilon^t = r^t + \kappa Q^\pi(\mathcal{H}^t, o^{t+1}, \varphi_b^{t+1}(\mathcal{H}^{t-1}, o^{t+1}); \theta)$. The parameters of each actor are taken into account when maximizing expected total rewards. The target function at time slot t can be expressed by $\Gamma(\omega_b) = \mathbb{E}_{o^t, \mathcal{H}^{t-1}}[Q^\pi(\mathcal{H}^{t-1}, o^t, \varphi_b^t(\mathcal{H}^{t-1}, o_b^t; \omega_b); \theta)]$. The gradient with respect of parameters are calculated by utilizing chain rule as

$$\nabla_{\omega_b}\Gamma(\omega_b) \simeq \mathbb{E}_{o^t, \mathcal{H}^{t-1}}\left[\nabla_{\omega_b} Q^\pi\left(\mathcal{H}^{t-1}, o^t, \varphi_b^t\left(\mathcal{H}^{t-1}, o_b^t; \omega_b\right); \theta\right)\right], \quad (19)$$

that can be rewritten as,

$$\begin{aligned}\nabla_{\omega_b}\Gamma(\omega_b) &= \mathbb{E}_{o^t, \mathcal{H}^{t-1}}\Big[\nabla_{\varphi_t^b(\mathcal{H}^{t-1}, o_b^t; \omega_b)} Q^\pi\left(\mathcal{H}^{t-1}, o^t, \varphi_b^t\left(\mathcal{H}^{t-1}, o_b^t; \omega_b\right); \theta\right) \\ &\qquad \nabla_{\omega_b}\varphi_b^t\left(\mathcal{H}^{t-1}, o_b^t; \omega_b\right)\Big]. \quad (20)\end{aligned}$$

Furthermore, the SEU based on LSTM can be trained with respect to the following loss function minimization as

$$\begin{aligned}\Delta(\varpi) &= \mathbb{E}_{o^t, \mathcal{H}^{t-1}}\left[\left(Q^\pi\left(f_{\text{LSTM}}\left(\mathcal{H}^{t-2}, \left[o_b^{t-1}, a_b^{t-1}\right]; \phi\right), a^t, o^t; \theta\right) - \Upsilon^t\right)^2\right] \\ &\quad - \mathbb{E}_{o^t, \mathcal{H}^{t-1}}\left[Q^\pi\left(f_{\text{LSTM}}\left(\mathcal{H}^{t-2}, \left[o_b^{t-1}, a_b^{t-1}\right]; \phi\right), a^t, o^t; \theta\right)\right]. \quad (21)\end{aligned}$$

The proposed solution is summarized in Algorithm 1. Instead of updating online, all trajectories are saved in replay buffer $\mathcal{D}$ in order to learn with the minibatch updating [62]. Further, a minibatch of episodes is randomly selected and simultaneously evaluated at each training step to update the parameters of the actors and critic networks. Apart from MARDPG, we implemented some other DRL algorithms as a benchmark. First, the MADDPG is implemented [63]. The only difference between MADDPG and our proposed solution is that MADDPG utilizes a fully connected deep networks for optimizing actor-critic parameters. In contrast, this aim uses a recurrent neural network (RNN) in MARDPG. SAC is also a DRL algorithm for continuous action space maximum entropy RL based on off-policy actor-critic parameters [64]. Therefore, the optimal policy focuses on maximizing the entropy-regularized reward instead of maximizing the discounted cumulative reward. On the other hand, the actor aims to maximize the expected reward and entropy [64]. In this work, we utilize the distributed SAC, the multi-agent version of which the actors do not share the policy. Moreover, a greedy algorithm is implemented as another benchmark along with MADDPG and DSAC. According to this algorithm, each action is chosen based on a greedy policy [13], [65].

Algorithm 1: Multi-Agent RDPG Algorithm

```
Run the environment simulator and generate IIoT nodes' locations and UAVs' initial positions
Initialize the actors' parameters $\omega_1, \dots, \omega_B$ for $B$ actor networks and critic parameters $\theta$
Initialize the replay buffer $\mathcal{D}$
for episodes $e = 1$ to $E$ do
    for each timestep $t$ and $o^t \neq$ terminal do
        Initialize the SEU $\mathcal{H}^0$, $t = 1$
        for agent $b = 1$ to $B$ do
            The agent chooses an action based on its policy: $a_b^t = \varphi_b^t\left(\mathcal{H}^{t-1}, o_b^t\right)$
        The agent receives its reward $r_t^b$, and the global reward is calculated as $r^t$
        The agent derives a new observation $o_b^{t+1}$, $t = t + 1$
        Save the episode $\left\{\mathcal{H}^0, o^1, a^1, \dots\right\}$ in $\mathcal{D}$
    Sample a random minibatch of episodes $G$ from $\mathcal{D}$
    for each episode $e$ in $G$ do
        for $t = T$ downto 1 do
            Update the critic with respect to loss function minimization:
            $\Delta(\theta) = \left(Q^\pi\left(\mathcal{H}^{t-1}, a^t, o^t; \theta\right) - \Upsilon^t\right)^2$, where
            $\Upsilon^t = r^t + \kappa Q^\pi\left(\mathcal{H}^t, o^{t+1}, \varphi_b^{t+1}(\mathcal{H}^{t-1}, o^{t+1}); \theta\right)$
            Update target parameter: $\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$
            Update actor $b$ by maximizing the following target function:
            $\Gamma(\omega_b) = \left(Q^\pi\left(\mathcal{H}^{t-1}, o^t, \varphi_b^t\left(\mathcal{H}^{t-1}, o_b^t; \omega_b\right); \theta\right)\right)$
            Update the SEU with respect to the following loss minimization:
            $\Delta(\varpi) =$
            $\left(Q^\pi\left(f_{\text{LSTM}}\left(\mathcal{H}^{t-2}, [o_b^{t-1}, a_b^{t-1}]; \phi\right), a^t, o^t; \theta\right) - \Upsilon^t\right)^2$
            $- Q^\pi\left(f_{\text{LSTM}}\left(\mathcal{H}^{t-2}, [o_b^{t-1}, a_b^{t-1}]; \phi\right), a^t, o^t; \theta\right)$
            Update target parameters:
            $\varpi' \leftarrow \tau'\varpi + (1 - \tau')\varpi'$
            $\omega_b' \leftarrow \tau'\omega_b + (1 - \tau')\omega_b'$
```

A. CONVERGENCE ANALYSIS

In this subsection, the convergence of MARDPG and other DRL-algorithmic benchmarks are analyzed. Considering the principles of the Q-learning algorithm [66], as $t \to \infty$, the Q-function converges to the optimal value if the following constraints,

$$\sum_{t=0}^{\infty}\epsilon = \infty, \quad \sum_{t=0}^{\infty}\epsilon^2 < \infty, \quad \sum_{t=0}^{\infty}\hat{\epsilon} = \infty, \quad \sum_{t=0}^{\infty}\hat{\epsilon}^2 < \infty, \quad (22)$$

$$\lim_{t \to 0}\frac{\epsilon}{\hat{\epsilon}} = 0, \quad (23)$$

TABLE 1. The computational complexity and signaling overhead of proposed MARDPG and algorithmic benchmarks.

Algorithm	Computational Complexity	Signaling Overhead
MARDPG	$\mathcal{O}\left((E \times H_{\text{batch}} \times H_{\text{LSTM}}) \times \text{CC}_{\text{neural}} \times B\right)$	$16 \times \left(\underbrace{\lvert \boldsymbol{S}^t \rvert}_{\text{Information of state}} + \underbrace{\lvert \boldsymbol{A}^t \rvert}_{\text{Information of action}} + \underbrace{B}_{\text{Information of reward (number of agents)}}\right)$
MADDPG	$\mathcal{O}\left((E \times H_{\text{batch}}) \times \text{CC}_{\text{neural}} \times B\right)$	$16 \times \left(\underbrace{\lvert \boldsymbol{S}^t \rvert}_{\text{Information of state}} + \underbrace{\lvert \boldsymbol{A}^t \rvert}_{\text{Information of action}} + \underbrace{B}_{\text{Information of reward (number of agents)}}\right)$
DSAC	$\mathcal{O}\left((E \times H^b_{\text{batch}}) \times \text{CC}^b_{\text{neural}}\right)$	$16 \times \left(\underbrace{\lvert \boldsymbol{S}^t \rvert}_{\text{Information of state}} + \underbrace{\lvert \boldsymbol{A}^t \rvert}_{\text{Information of action}} + \underbrace{B}_{\text{Information of reward (number of agents)}}\right)$
Greedy	$\mathcal{O}\left((E \times H_{\text{batch}}) \times \text{CC}_{\text{neural}}\right)$	$16 \times \left(\underbrace{\lvert \boldsymbol{S}^t \rvert}_{\text{Information of state}} + \underbrace{\lvert \boldsymbol{A}^t \rvert}_{\text{Information of action}}\right)$

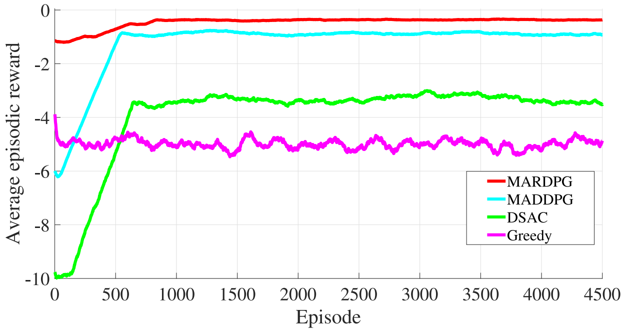


FIGURE 3. Reward function convergence of MARDPG, MADDPG, DSAC, and Greedy.

are satisfied and the learning rate of actor (ϵ) and critic ($\hat{\epsilon}$) networks are deterministically increased. Moreover, the results and reward function can meet the convergence requirements because the reward function, $|r(s^t, a^t_b)|$, is bounded. The convergence analysis of MARDPG, MADDPG, DSAC, and greedy are illustrated in Fig. 3. Our proposed solution achieves higher performance over episodes than other benchmarks.

IV. COMPUTATIONAL COMPLEXITY AND SIGNALING OVERHEAD ANALYSIS

Computational complexity (CC) is an important criterion to indicate how our approach is computed and how many resources are required. Indeed, CC indicates how many resources are required to implement this algorithm [67]. A higher CC means that this algorithm is more expensive and complicated to implement. The CC of neural networks with N layers and $\tilde{k}_n$ neurons can be calculated by $\text{CC}_{\text{neural}} = \sum_{n=1}^{N-1} \tilde{k}_n \tilde{k}_{n+1}$ [68]. As a result of utilizing the neural network, the number of input layers equals the size of the state, $|\boldsymbol{S}^t|$, while the number of neurons in the output layer equals the number of actions, $|\boldsymbol{A}^t|$. Therefore, the CC of MARDPG is $\text{CC}_{\text{MARDPG}} = \mathcal{O}((E \times H_{\text{batch}} \times H_{\text{LSTM}}) \times \text{CC}_{\text{neural}} \times B)$, where H_{batch}, H_{LSTM}, E, and B represent the size of the batch memory, the size of LSTM memory, number of episodes, and number of agents, respectively. The CC of the other algorithmic benchmarks can be found in Table 1. It is noticeable that the CC of DSAC needs some explanations. Regarding those agents in the distributed DRL algorithm trying to solve the problem parallelly, the CC, in this case, is equal to the maximum CC of agents as follows:

$$\mathcal{O}\Big(\max\Big(\Big(E \times H^1_{\text{batch}}\Big) \times \text{CC}^1_{\text{neural}}\Big) \times \cdots \times \Big(\Big(E \times H^B_{\text{batch}}\Big) \times \text{CC}^B_{\text{neural}}\Big)\Big) \tag{24}$$

However, when the agents are the same and their actions are equal, the maximum CC is equal to the CC of each one. Therefore, we can simplify the CC by following:

$$\Big(E \times H^1_{\text{batch}}\Big) \times \text{CC}^1_{\text{neural}} = \cdots = \Big(E \times H^B_{\text{batch}}\Big) \times \text{CC}^B_{\text{neural}}, \tag{25}$$

$$\begin{aligned} &\mathcal{O}\Big(\max\Big(\Big(\Big(E \times H^1_{\text{batch}}\Big) \times \text{CC}^1_{\text{neural}}\Big) \times \ldots \\ &\times \Big(\Big(E \times H^B_{\text{batch}}\Big) \times \text{CC}^B_{\text{neural}}\Big)\Big)\Big) \\ &= \mathcal{O}\Big(\Big(E \times H^b_{\text{batch}}\Big) \times \text{CC}^b_{\text{neural}}\Big), \forall b \in \mathcal{B}. \end{aligned} \tag{26}$$

Furthermore, each algorithm requires Information to solve the problem, known as signaling overhead (SO). In fact, SO helps us measure how much data is needed to solve the specific problem with a given algorithm [67]. Furthermore, a higher signaling overhead implies the necessity to coordinate, manage, or control a greater amount of information to address the problem. In the case of reinforcement learning, this Information is divided into three categories. We need three groups of Information to solve the problem. Firstly, information about the state that is represented by $|\boldsymbol{S}^t| = |\boldsymbol{g}^t|$. Secondly, the required data for actions, which is indicated by $|\boldsymbol{A}^t| = |\mathbf{P}^t| + |\boldsymbol{\Psi}^t| + |\mathbf{F}^t| + |\mathbf{L}|$ (it consists of four parts: the uplink power allocation matrix, uplink subcarrier allocation matrix, computational resource allocation matrix, and location selection of ABS, respectively). Thirdly, the number of agents, B, presents the Information needed for reward. Last but not least, we presume that each resource matrix element is modeled by "Float 16" bits. Hence, the SO for multi-agent algorithms, MARDPG and MADDPG, is $16 \times (|\boldsymbol{S}^t| + |\boldsymbol{A}^t| + B)$. However, for the greedy algorithm that is not multi-agent, SO is $16 \times (|\boldsymbol{S}^t| + |\boldsymbol{A}^t|)$. Unless the CC for the DSAC, SO is just like other multi-agent algorithms because DSAC is a multi-agent approach, where the only difference is that the agents do not share the policy. Accordingly, the data for solving the problem is similar

TABLE 2. Learning and environmental parameters.

Parameters	Description	Value
-	Size of area	$2000 \times 2000 \times 200$ m
B	The number of ABS	3
U	The number of IIoT nodes	30
$K, \hat{W}$	The number of subcarriers, the bandwidth of each subcarrier	8 and 120 kHz
σ^2	AWGN variance	-90 dB
$\mu_b = f_b^{\max}$	Computational resource of each MEC (Service rate)	4×10^9 CPU cycle/second
$\mu_b = f_b^{\max}$	Computational resource of each Fog (Service rate)	4×10^{10} CPU cycle/second
DL_b^t	Packet Arrival Rate for MEC	30 pps
DL_a^t	Packet Arrival Rate for Fog	10 pps
DL_u	Packet's Length	100 Kbits
Ω_u	Needed CPU Cycle for Processing one bit of the user's paket (task)	2 CPU Cycle
p^{min}, p^{max}	Minimum and maximum uplink data transmission powers	100 mW and 800 mW
d^{min}	Minimum distance between two ABS to prevent a collision	2 m
$\eta_{\mathrm{LoS}}, \eta_{\mathrm{NLoS}}$	The LoS and the NLoS mean excessive path loss	0.1 dB and 21 dB
D^{Th}	The maximum tolerable E2E delay	100 ms
$z^{\min}, z^{\max}$	The minimum and maximum flying altitude	20 m and 100 m
C^{Th}	The maximum number of nodes that can operate in one subcarrier	2
$v_b^{\max}$	The maximum velocity of each UAV	4 m/s
ε	The portion of data that is offloaded to fog	0, 0.2, 0.5, and 0.7
$\mathcal{B}$	Batch size	64
E	Number of episodes	4500
$\mathcal{D}$	SEU storage size	800000
κ	Discount factor	0.87
ϵ	Learning rate of the actor networks	0.00001
$\hat{\epsilon}$	Learning rate of the critic network	0.00005
-	Hidden layers activation function	ReLU
-	Output layer activation function	tanh
-	Number of hidden layers	2
-	Numbers of neurons located in each hidden layer	512
τ	Period for updating the network	0.001

to MADDPG and MARDPG. The SO of these methods is summarized in Table 1.

V. SIMULATION

In this section, we present numerical results to evaluate the performance of the multi-agent DRL algorithm in joint resource allocation and trajectory design in a non-terrestrial network-enabled IIoT nodes for disaster relief. We consider seven scenarios. First, the considered environment is described. Second, a comparison is made between MARDPG and other algorithmic benchmarks. In fact, we want to find out which algorithm performs better in our case [69]. Next, two conventional objective functions are implemented with the same algorithm (MARPDG) in order to prove the superiority of the proposed objective function. This scenario aims to understand the performance of objective functions, not algorithms. Next, the performance of MEC is compared to COC (MEC and Fog). In fact, it helps to determine which computational structure is better for our case [70], [71]. After that, the trajectory design of the UAVs is provided to realize that the UAVs operate correctly without collision. Then, solving optimization problems with non-exact methods is useful in many cases, as long as the optimality gap is determined and ensured to be acceptable. Therefore, we implemented the exhaustive search algorithm to determine the global optimum [72]. Then, we measured the MARDPG optimality gap using the result of the exhaustive search algorithm as a reference. Finally, in the last scenario, by calculating the fairness score, we can realize how much resources are fairly distributed among the users [73].

A. SIMULATION ENVIRONMENT

We consider that 30 ($U = 30$) IIoT nodes randomly spread out through the squared disaster zone of size 2000 m $\times$ 2000 m. Additionally, 3 UAVs ($B = 3$) can fly from the minimum height of $z^{\min} = 20$ m to the maximum height of $z^{\max} = 100$ m. In addition, the minimum distance between two ABS to prevent a collision is $d^{\min} = 2$ m [50]. Moreover, we consider $K = 8$ subcarriers for each ABS, and each subcarrier has bandwidth $\hat{W} = 120$ kHz bandwidth. Accordingly, each subcarrier can be assigned to a maximum of two nodes, $C^{\mathrm{Th}} = 2$. Furthermore, the minimum and maximum uplink data transmission powers are $p^{\min} = 100$ mW and $p^{\max} = 800$ mW, respectively. The additive white Gaussian noise (AWGN) variance is $\sigma^2 = -90$ dB [74]. Furthermore, the LoS and the NLoS mean excessive path loss values are $\eta_{\mathrm{LoS}} = 0.1$ dB and $\eta_{\mathrm{NLoS}} = 21$ dB. We consider each user's packet (task) to be 100 kbits. Apart from that, the computational resource of each MEC (ABS) is $f_b^{\max} = 4 \times 10^9$ CPU cycle/second (the MEC's service rate is equal to the computational resource of MEC). In addition, the computational resource of each Fog (HAPS) is $f_b^{\max} = 4 \times 10^{10}$ CPU cycle/second (the Fog'sservice rate is equal to the computational resource of Fog). Eventually, the maximum tolerable E2E delay for each node is $D^{\mathrm{Th}} = 100$ ms. In addition, all the learning and environmental parameters can be found in Table 2, [75]. Additionally, we consider 2000 timeslots, each of which has a length of 50 ms. Therefore, in each episode, the problem is run for 2000 timeslots. Hence, the average reward in last 100 timeslots is considered as the reward of one episode. It is noticeable that the config of the computer that was used

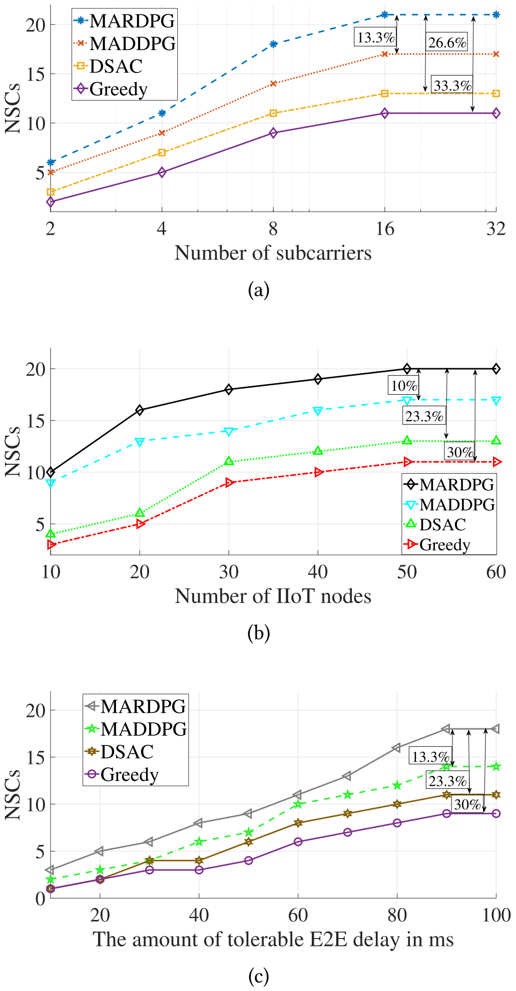


FIGURE 4. Performance analysis of MARDPG, MADDPG, DSAC, and Greedy.

for running the simulations is: It has 48 gigabytes (GB) of random-access memory (RAM), Intel Core i5–11400F up to 4.5 GHz, one terabyte (TB) of hard disk drive (HDD), two 500 GB solid state drives, and NVIDIA GeForce RTX 2080. Learning networks are also implemented in Python 3.7, Tensorflow 2.6 library, and Keras 1.7 library. Also, all the source codes related to this paper can be found in [76].

B. COMPARISON BETWEEN DIFFERENT DRL ALGORITHMS

In this Subsection, we aim to compare three algorithmic benchmarks whose environment and parameters are the same, which can be observed in Figure 4. Increasing the number of subcarriers grows the NSCs until it reaches $K = 16$, as can be observed in Fig. 4(a). After that, the total number of connections remained unchanged because of E2E delay not only depends on the radio resources (subcarriers), but also on the computational resources and buffer throughput. As a result, the NSCs would not increase even though more bandwidths are added to the system. The effect of raising the IIoT nodes is studied to provide more justification for assuring the efficiency of our architecture.

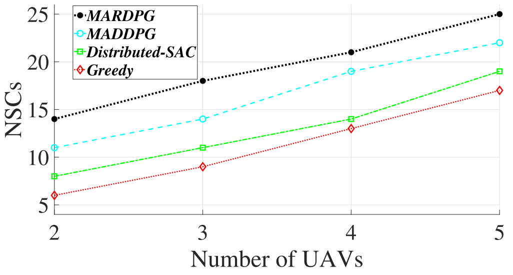


FIGURE 5. Performance analysis of MARDPG, MADDPG, DSAC, and Greedy by increasing the number of UAVs.

As observed from Fig. 4(b), the capacity of the system to admit more E2E delay-aware connections is limited, and it is significantly increased from $U = 10$ to $U = 30$. Nevertheless, NSC moderately increased and remained unchanged when the number of IIoT nodes expanded from 50 to 60. Moreover, NSC is substantially affected by the maximum tolerable E2E delay for each IIoT node, D^{Th}. Accordingly, Fig. 4(c) shows that with increasing D^{Th} from 10 ms to 100 ms, the NSCs also increases. The NSCs first raises until $D^{\mathrm{Th}} = 90$ ms and then saturates. Significantly, MARDPG achieves higher performance than MADDPG, DSAC, and Greedy, which can be a result of LSTM and SEU. For example, in Figure 4(a), MARDPG has 13.3%, 26.6%, and 33.3% better gain than MADDPG, DSAC, and Greedy, respectively. According to Section IV and Table 1, though MARDPG is more complex than others, it has a significant performance gap between other methods and MARDPG, whereby more complexity can be justified (SO of all multi-agent approaches are similar, and just Greedy is less than other algorithms).

In addition, to show the scalability of the proposed method, we examine the effect of increasing the number of UAVs. When the number of UAVs are increased, NSCs are increased. Indeed, by increasing the number of UAVs, we raise the amount of resources. As a result, NSCs are incremented, as seen in Fig. 5. Similarly, when the number of UAVs are increased from two to five (we did not consider one UAV because we want to be still multi-agent and also more than five UAVs cannot be operational for the dimension of our environment), MARDPG achieves better performance than MADDPG, DSAC, and Greedy.

C. COMPARISON BETWEEN DIFFERENT OBJECTIVE FUNCTIONS

The reasons why NSCM is more efficient than the two conventional methods are given in this subsection. The first approach minimizes the average delay that the model can be found [43], [45]. Secondly, minimizing the maximum delay that the model is considered in du2017computation, li2018min, li2022min. Indeed, the algorithm (MARDPG) and problem in (16a) (constraints and optimization variables) are similar, and just the objective is changed. As can be seen in Fig. 6(a), NSCs reached a maximum when K increased to 16, followed by saturation from $K = 16$ to

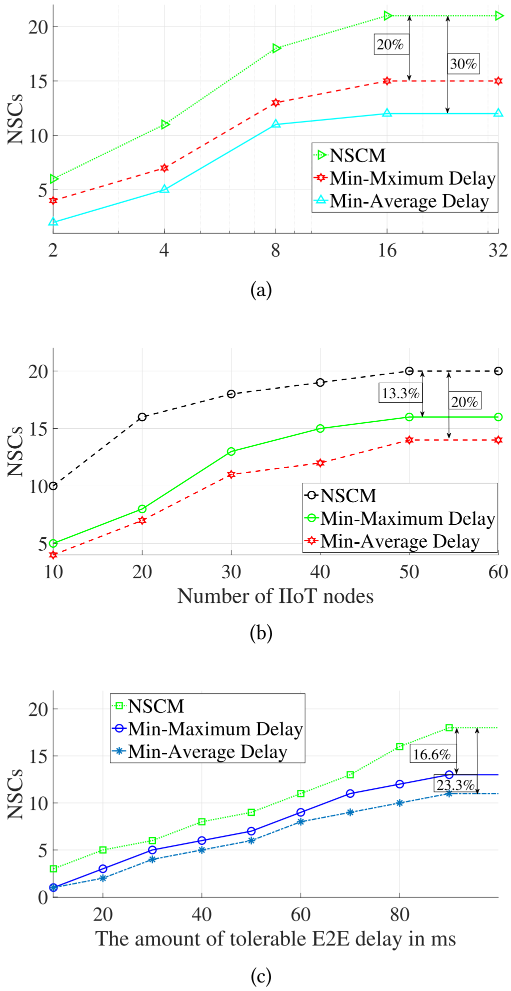


FIGURE 6. Performance analysis of NSCM, Min-Max delay, and Min-Average delay that are implemented with MARDPG.

$K = 32$. The reason for this is that other resources are limited, and NSCs cannot be constantly increased by the increase in radio resources alone. This pattern is also true for Figure 6(b) and 6(c). Obviously, the results indicate that NSCM is superior to the two other models. Although objective functions differ in these three dissimilar models, their CC and SE are equal to the MARDPG calculated in the first row of Table 1 because all algorithm specifications (actions, state, and reward) remain unchanged. As a result, it is notable that our proposed method (NSCM) with the same CC and SE showed a better performance. For instance, NSCM has roughly 13% better gain than minimizing maximum delay and 20% than minimizing average delay in Figure 6(b).

D. THE PERFORMANCE OF COLLABORATIVE COMPUTING

In this scenario, the main goal is to compare the performance of MEC and COC with three different ε, 0.2, 0.5, and 0.7. Again our algorithm remains unchanged, and only the computing model changes, which can be found in [25], [26]. When the computation model is changed, the structure of the problem in (16a), moderately differs. Assuming that transmission and propagation delays in offloading data from each UAV to HAPS are negligible, only the processing and queuing delays of HAPS to compute the portion of tasks are needed to be defined. According to the equations (14) and (13), the queuing ($D^{\text{queu},t}_{u,a}$) in [s] and processing ($D^{\text{proc},t}_{F_{u},a}$) delays of HAPS in [$s$], a are by follows

$$D^{\text{queu},t}_{u,a} = \frac{1}{\mu_a - DL^t_a}, D^{\text{proc},t}_{F_{u},a} = \frac{\varepsilon DL_u \Omega_u}{f^t_{F_{u},a}}, \tag{27}$$

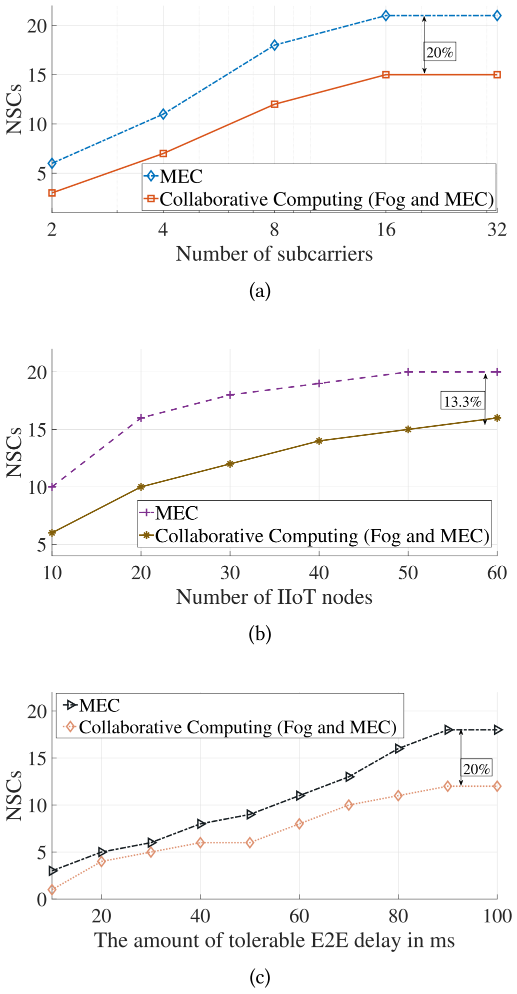
where $DL^{\max,t}_a$, μ_a, and $f^t_{F_{u},a}$ represent packet arrival rate at time slot t for HAPS a, service rate of HAPS a, $f^{\max}_a$ in CPU cycle per second, and computational resources that are allocated to task F_u that is pertained to node u data which are provided by HAPS a, respectively. As a result, the E2E delay for the COC scenario can be as follows,

$$D^{\text{E2E},t}_u = D^{\text{tran},t}_{u,b} + D^{\text{queu},t}_{u,b} + D^{\text{proc},t}_{F_{u},b} + D^{\text{queu},t}_{u,a} + D^{\text{proc},t}_{F_{u},a}, \quad \forall u \in \mathcal{U}, \forall t. \tag{28}$$

Next, three changes were made to modify the problem (16a) for implementing COC. Firstly, the optimization variable $\mathbf{F}$ changed to $[\mathbf{F}]_{u,b,a} = (f^t_{F_{u},b}, f^t_{F_{u},a})$, means that HAPS now is an agent in addition to three UAVs, and try to allocate CPU cycle ($f^t_{F_{u},a}$) to a portion of data that is offloaded to it. Secondly, one constraint should be added that is $\sum^J_{j=1} f^t_{F_{j},a} \leq f^{\max}_a$. Thirdly, the state in III, cannot only be channel information, and those about prior CPU cycle allocation are needed. Thus, the state should be altered to $s^t_{b,a} = [g^t_{u,b,k}, f^{t-1}_{F_{u},b}, f^{t-1}_{F_{u},a}], b \in \mathcal{B}$. The CC and SO of MEC and COC are in Table 3. It is obvious that COC is more complex than MEC because of the number of agents; therefore, the number of neurons is more in it. Furthermore, due to the size of the state, action space, and the number of agents being more than MEC, COC imposes higher signaling overhead. As evident from Figure 7(a), MEC showed a better performance than COC (20% better) when most data was still processed in MEC. As $\varepsilon = 0.5$, NSCs of COC started from 5 (1 less than the NSCs of MEC) and steadily soared until it intersected the MEC's graph in Figure 8(a). Finally, it finished with a 10% gain over MEC. When ε increased to 0.7, the NSCs reflected the same pattern, and COC's performance gap leveled to around 13% in Figure 9(a). Increasing ε from 0.2 to 0.7 leads to dramatic growth in the performance of COC. In Figure 7(b) MEC has roughly 13% better gain when $N = 60$. However, COC bridges this gap and achieved higher NSC, which is about 16% and 20% more than MEC in Figures 8(b) and 9(b). This trend is similar in Figures 7(c), 8(c), and 9(c). However, these achievements cost more CC and SO, as shown in Table 3. To sum up, in the case of mission-critical IIoT, lower CC and SO are more crucial, which can be attained with acceptable lower performance. Based on that, MEC is a better computing model.

TABLE 3. The computational complexity and signaling overhead of MEC-MARDPG and collaborative computing-MARDPG.

Model	Computational Complexity	Signaling Overhead
MEC-MARDPG	$\mathcal{O}\left((E \times H_{\text{batch}} \times H_{\text{LSTM}}) \times \text{CC}_{\text{neural}} \times B\right)$	$16 \times \left(\underbrace{\lvert \boldsymbol{S}^t \rvert}_{\text{Information of state}} + \underbrace{\lvert \boldsymbol{A}^t \rvert}_{\text{Information of action}} + \underbrace{B}_{\text{Information of reward (number of agents)}} \right)$
COC-MARDPG	$\mathcal{O}\left((E \times H_{\text{batch}} \times H_{\text{LSTM}}) \times \text{CC}_{\text{neural}} \times (B+1)\right)$	$16 \times \left(\underbrace{\lvert \boldsymbol{S}^t \rvert}_{\text{Information of state}} + \underbrace{\lvert \boldsymbol{A}^t \rvert}_{\text{Information of action}} + \underbrace{(B+1)}_{\text{Information of reward (number of agents)}} \right)$

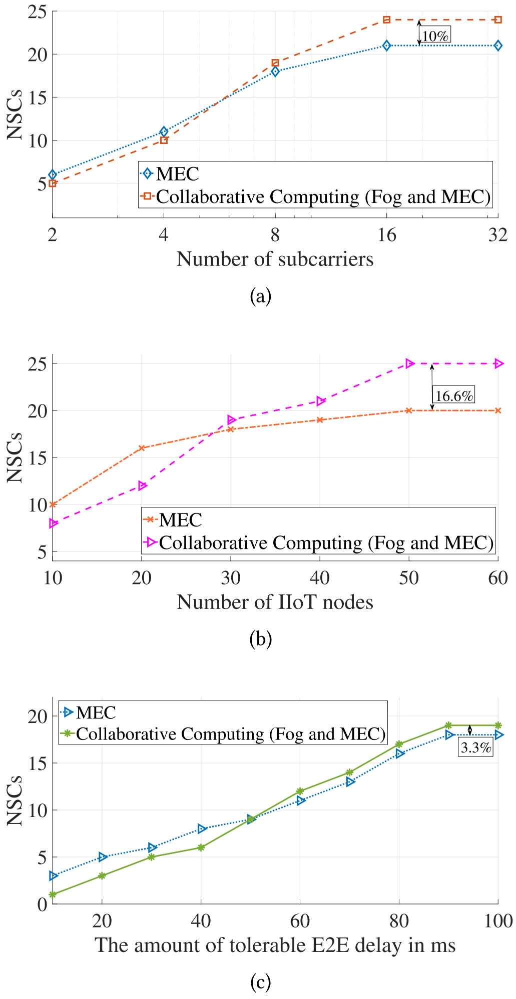


(a)

(b)

(c)

FIGURE 7. Performance analysis of MEC and COC with $\varepsilon = 0.2$. (a) The effect of increasing subcarriers on the NSCs with DTh = 100 ms, B = 3, and U = 30 (b) The effect of increasing IIoT nodes on the NSCs with DTh = 100 ms, B = 3, and K = 8 (c) The effect of increasing the maximum tolerable E2E delay on the NSCs with U = 30, B = 3, and K = 8.



(a)

(b)

(c)

FIGURE 8. Performance analysis of MEC and COC with $\varepsilon = 0.5$. (a) The effect of increasing subcarriers on the NSCs with DTh = 100 ms, B = 3, and U = 30 (b) The effect of increasing IIoT nodes on the NSCs with DTh = 100 ms, B = 3, and K = 8 (c) The effect of increasing the maximum tolerable E2E delay on the NSCs with U = 30, B = 3, and K = 8.

E. TRAJECTORY DESIGN

In this part, the result of the trajectory design of UAVs is provided. Figure 10 illustrates the trajectories of UAVs and the IIoT node locations in 3D. As can be seen, the UAVs try to find and move to the locations that lead to more NSCs. The shorter distance between the node and the ABS means that less transmission power is required to send data to the ABS. On the other hand, the optimal allocation of power is linked to the way in which UAVs move. As a result, the UAVs tried to keep these distances as short as possible. This tended to result in more UAV paths overlapping.

F. OPTIMALITY GAP AND COMPARISON TO EXHAUSTIVE SEARCH

It is generally accepted that determining how far the proposed algorithm is from the global optimal value can help to understand the optimality of the algorithm. Nonetheless, the actions in reinforcement learning are taken with noise, and exploration and exploitation are involved, so it seems impossible to reach a global optimum. In other words, achieving global value cannot be guaranteed for most DRL algorithms with common reinforcement learning structures.

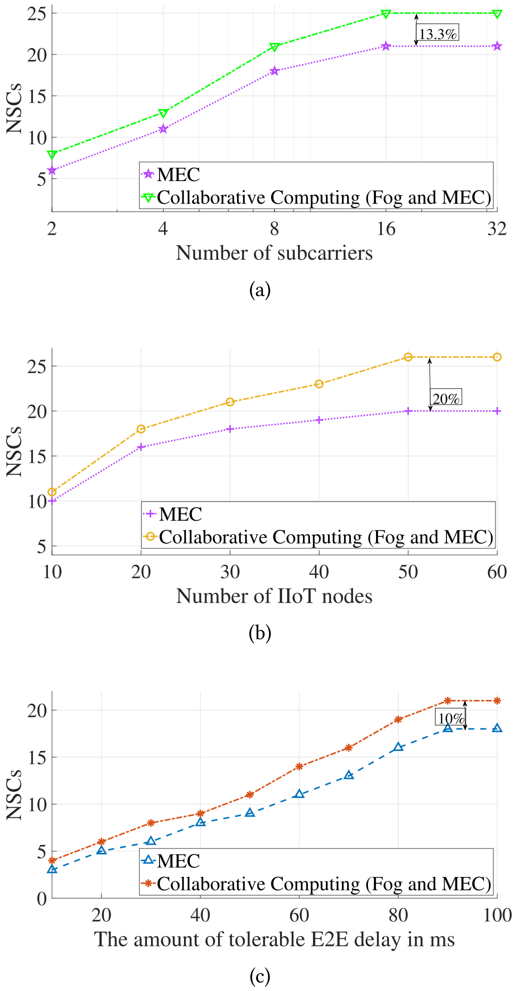


FIGURE 9. Performance analysis of MEC and COC with $\varepsilon = 0.7$. (a) The effect of increasing subcarriers on the NSCs with DTh = 100 ms, B = 3, and U = 30 (b) The effect of increasing IIoT nodes on the NSCs with DTh = 100 ms, B = 3, and K = 8 (c) The effect of increasing the maximum tolerable E2E delay on the NSCs with U = 30, B = 3, and K = 8.

TABLE 4. Determining the optimality gap of proposed solution by comparing to exhaustive search method.

Algorithm	NSCs	Computational Complexity	Run time	Optimality Gap
MARDPG	18	$\mathcal{O}\left(\left(E \times H_{\text{batch}} \times H_{\text{LSTM}}\right) \times \text{CC}_{\text{neural}} \times B\right)$	12 ms	6.67%
Exhaustive Search	20	$\mathcal{O}\left(E \times \lvert\mathcal{U}\rvert \times \lvert\mathcal{K}\rvert \times B!\right)$	960 ms	-

Furthermore, our proposed problem is NP-hard, mixed-integer, online, and large, which makes it difficult, if not sometimes intractable, for conventional solution approaches. However, DRL algorithms are still promising for solving such problems, despite their suboptimal value and performance. But, the optimality gap should be defined to determine its amount. To this end, we exploited an exhaustive search (Brute-force search) algorithm to determine this gap due to its capability to solve the online problem [77], [78], [79]. The simulation setup is just like the mentioned initialization in Section V-A ($K = 8, U = 30, B = 3, D^{\text{Th}} = 100$ ms). As can be observed in Table 4, MARDPG roughly has a 6% gap compared to the exhaustive search. Although the exhaustive search algorithm has a better performance than MARDPG, its running time is dramatically high (80 times higher than MARDPG), and it has a higher complexity in terms of CC. On the basis of this, we conclude that it is perfect for the solution of small scale problems, not large scale problems like the one we are proposing. As a result, MARDPG may be a better choice as the main solution. It is worth mentioning that as the optimality gap increases, the efficiency of the algorithm decreases. In terms of disaster relief, it will be a more promising system if your method can provide more NSCs. More NSCs mean more IIoT nodes can send their control information. It will lead to better disaster management, resource allocation, and the prevention of further infrastructure damage. Otherwise, a lack of information can cause the situation to get out of control. In a disaster relief scenario, we need to protect critical infrastructure that is monitored by IIoT nodes. To do this, we need to establish communication in disaster areas [80]. Considering the scalability and flexibility of non-terrestrial networks, they are suitable for this scenario [10]. In addition, it is better to process data locally than to use other options far from the disaster area due to the additional transmission delay. For this reason, we used non-terrestrial networks with hierarchical structures equipped with computing resources.

G. FAIRNESS ANALYSIS

Analyzing the system model in aspects of fairness in the transmission rate of users is within the scope of this Subsection. Until now, we considered various examinations to evaluate and compare our performance with other benchmarks. Although the user's transmission rate is involved in (7) and indirectly guaranteed in constraint (16h), it is crucial to determine how it is fairly allocated to users. In other words, we want to evaluate the power and subcarrier allocations in terms of fairness. As mentioned, the set of users is denoted by $\mathcal{U} = \{1, \dots, u, \dots, U\}$, where U is the total number of IIoT nodes. In addition, the maximum uplink achievable rate represented by $R^{t}_{u,b,k}$ in equation (7), in which the average rate of each user on all subcarriers during time slots can be presented by $\mathbb{E}[\sum_{k=1}^{K} R^{t}_{u,b,k}]$. Hence, according to [81], the fairness score (FS) can be formulated as follows,

$$\text{FS} = \frac{\left(\sum_{u\in\mathcal{U}} \mathbb{E}\left[\sum_{k=1}^{K} R^{t}_{u,b,k}\right]\right)^2}{|\mathcal{U}| \cdot \sum_{u\in\mathcal{U}} \mathbb{E}\left[\sum_{k=1}^{K} R^{t}_{u,b,k}\right]^2}. \tag{29}$$

Considering this formula, we analyze the FS in terms of rate with a different number of users from 10 to 60. As can be seen in Table 5, the FS score gradually decreased from around 98% to approximately 70% when $U = 60$. This decreasing trend is a result of resource limitations. Nevertheless, it is noticeable that for our simulation setup ($U = 30$), the acceptable FS is achieved (91%).

VI. CONCLUSION

In this paper, a MARDPG-based resource allocation and trajectory design in HNTN-enabled IIoT networks was

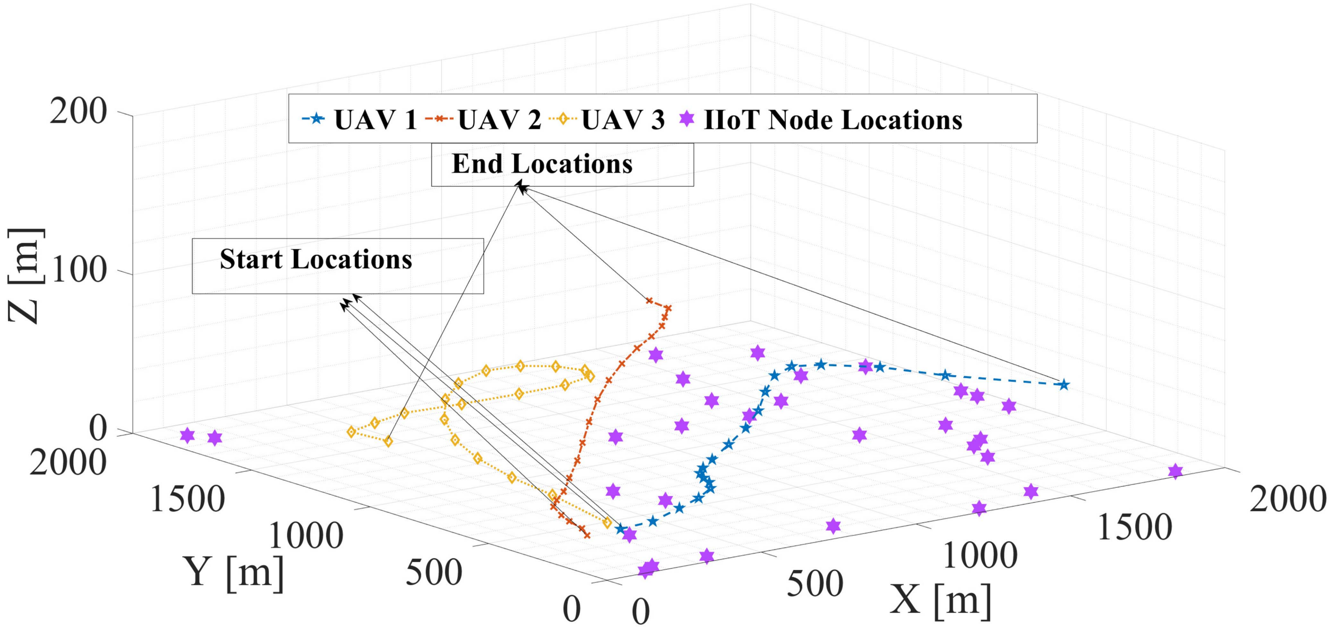


FIGURE 10. The trajectories of UAVs in the proposed scheme.

TABLE 5. The fairness score among IIoT nodes.

Number of IIoT nodes, U	10	20	30	40	50	60
FS	98.1%	95%	91%	86.6%	79.27%	70.44%

developed to maximize NSCs in a disaster area, taking into account computational resources and E2E delays. The proposed MARL algorithm utilizes an LSTM-based SEU to enhance the collaboration between agents where a group of UAVs (as agents) were continuously trying to learn total and individual rewards. In addition, the impact of increasing the number of subcarriers, IIoT nodes, and maximum tolerable E2E delay on NSCs were considered as three main criteria. Simulation results demonstrated that not only MARDPG had a better convergence as compared to MADDPG, DSAC, and Greedy, but also achieved a higher performance in NSC maximization. Furthermore, it is noteworthy that NSCM achieved a higher gain for the same CC and SO than two conventional objective functions, minimizing the maximum delay and the average delay minimization in terms of NSCs. Additionally, the results revealed that the MEC structure is superior to COC with lower CC and SO, provided that most data are processed in it. Besides that, exhaustive search algorithm was exploited as an indicator for the optimality gap, showing that MARDPG had a gap of approximately 6 %. Finally, 91% FS was achieved in terms of achieved uplink transmission rates.

REFERENCES

[1] A. Mohammadisarab, A. Khalili, A. Nouruzi, N. Mokari, B. A. Arand, and E. A. Jorswieck, "Joint resource allocation, task processing, and trajectory design for UAV-assisted Industrial IoT users in 6G networks," in *Proc. IEEE Conf. Stand. Commun. Netw. (CSCN)*, 2022, pp. 71–77.

[2] L. Chettri and R. Bera, "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, Jan. 2019.

[3] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road towards 6G: A comprehensive survey," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 334–366, 2021.

[4] A. Nasrallah et al., "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 88–145, 1st Quart., 2019.

[5] S. Parkvall, E. Dahlman, A. Furuskar, and M. Frenne, "NR: The new 5G radio access technology," *IEEE Commun. Stand. Mag.*, vol. 1, no. 4, pp. 24–30, Dec. 2017.

[6] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6G wireless communication systems: applications, requirements, technologies, challenges, and research directions," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 957–975, 2020.

[7] D. C. Nguyen et al., "6G Internet of Things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, Jan. 2022.

[8] C.-X. Wang et al., "On the road to 6G: Visions, requirements, key technologies, and testbeds," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 905–974, 2nd Quart., 2023.

[9] "Study on further NR RedCap UE complexity reduction, Version 0.0.1," 3GPP, Sophia Antipolis, France, Rep. TR 38.865, 2022. [Online]. Available: shorturl.at/nptz2

[10] "Study on narrow-B and Internet of Things (NB-IoT)/enhanced machine type communication (eMTC) support for non-terrestrial networks (NTN), Version 17.0.0," 3GPP, Sophia Antipolis, France, Rep. TR 36.763, 2021. [Online]. Available: shorturl.at/jqY58

[11] M. A. Ferrag, M. Debbah, and M. Al-Hawawreh, "Generative AI for cyber threat-hunting in 6G-enabled IoT networks," 2023, *arXiv:2303.11751*.

[12] T. K. Rodrigues and N. Kato, "Hybrid centralized and distributed learning for MEC-equipped satellite 6G networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1201–1211, Apr. 2023.

[13] T. Naous, M. Itani, M. Awad, and S. Sharafeddine, "Reinforcement learning in the sky: A survey on enabling intelligence in NTN-based communications," *IEEE Access*, vol. 11, pp. 19941–19968, 2023.

[14] G. Araniti, A. Iera, S. Pizzi, and F. Rinaldi, "Toward 6G non-terrestrial networks," *IEEE Netw.*, vol. 36, no. 1, pp. 113–120, Jan./Feb. 2021.

[15] W. Zhou et al., "Priority-aware resource scheduling for UAV-mounted mobile edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 7, pp. 9682–9687, Jul. 2023.

[16] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2334–2360, 3rd Quart., 2019.

[17] A. Khalili, A. Rezaei, D. Xu, and R. Schober, "Energy-aware resource allocation and trajectory design for UAV-enabled ISAC," 2023, *arXiv:2302.10124*.

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

[19] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, "Drone networks: Communications, coordination, and sensing," *Ad Hoc Netw.*, vol. 68, pp. 1–15, Jan. 2018.

[20] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2624–2661, 4th Quart., 2016.

[21] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging UAVs for disaster management," *IEEE Pervasive Comput.*, vol. 16, no. 1, pp. 24–32, Jan.–Mar. 2017.

[22] A. Ali, T. Mallick, S. Sakib, M. S. Hossain, and Y.-D. Lin, "Provisioning fog services to 3GPP subscribers: Authentication and application mobility," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 4926–4931.

[23] N. Li, W. Hao, F. Zhou, S. Yang, and N. Al-Dhahir, "Min-max latency optimization for IRS-aided cell-free mobile edge computing systems," 2022, *arXiv:2206.04205*.

[24] Q. Li, J. Lei, and J. Lin, "Min-max latency optimization for multiuser computation offloading in fog-radio access networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2018, pp. 3754–3758.

[25] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2018.

[26] Y. Li, B. Yang, H. Wu, Q. Han, C. Chen, and X. Guan, "Joint offloading decision and resource allocation for vehicular fog-edge computing networks: A contract-Stackelberg approach," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15969–15982, Sep. 2022.

[27] Z. Fei, Y. Wang, J. Zhao, X. Wang, and L. Jiao, "Joint computational and wireless resource allocation in multicell collaborative fog computing networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 11, pp. 9155–9169, Nov. 2022.

[28] A. Nouruzi et al., "Toward a smart resource allocation policy via artificial intelligence in 6G networks: Centralized or decentralized?" 2022, *arXiv:2202.09093*.

[29] C. Ssengonzi, O. P. Kogeda, and T. O. Olwal, "A survey of deep reinforcement learning application in 5G and beyond network slicing and virtualization," *Array*, vol. 14, Jul. 2022, Art. no. 100142.

[30] G. Qu, H. Wu, R. Li, and P. Jiao, "DMRO: A deep meta reinforcement learning-based task offloading framework for edge-cloud computing," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 3448–3459, Sep. 2021.

[31] R. Du, C. Liu, Y. Gao, P. Hao, and Z. Wang, "Collaborative cloud-edge-end task offloading in NOMA-enabled mobile edge computing using deep learning," *J. Grid Comput.*, vol. 20, no. 2, p. 14, 2022.

[32] F. Wei, G. Feng, Y. Sun, Y. Wang, S. Qin, and Y.-C. Liang, "Network slice reconfiguration by exploiting deep reinforcement learning with large action space," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2197–2211, 2020.

[33] Z. Gao, A. Liu, C. Han, and X. Liang, "Non-orthogonal multiple access-based average age of information minimization in LEO satellite-terrestrial integrated networks," *IEEE Trans. Green Commun. Network.*, vol. 6, no. 3, pp. 1793–1805, Sep. 2022.

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

[35] N. Nouri, J. Abouei, A. R. Sepasian, M. Jaseemuddin, A. Anpalagan, and K. N. Plataniotis, "Three-dimensional multi-UAV placement and resource allocation for energy-efficient IoT communication," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 2134–2152, Feb. 2022.

[36] R. Han, J. Wang, L. Bai, J. Liu, and J. Choi, "Age of information and performance analysis for UAV-aided IoT systems," *IEEE Internet Things J.*, vol. 8, no. 19, pp. 14447–14457, Oct. 2021.

[37] H. Yang, R. Ruby, Q.-V. Pham, and K. Wu, "Aiding a disaster spot via multi-UAV-based IoT networks: Energy and mission completion time-aware trajectory optimization," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5853–5867, Apr. 2021.

[38] Q. Zhang, Y. Jiang, X. Ge, Y. Huang, and Y. Liu, "Distributed data flow scheduling optimization in Industrial Internet of Things based on optimal transport theory," *IEEE Internet Things J.*, vol. 10, no. 14, pp. 12961–12974, Jul. 2023.

[39] Y. Zhao, J. Hu, K. Yang, and X. Wei, "A joint communication and control system for URLLC in Industrial IoT," *IEEE Trans. Veh. Technol.*, vol. 72, no. 11, pp. 15074–15079, Nov. 2023.

[40] V. D. Tuong, W. Noh, and S. Cho, "Sparse CNN and deep reinforcement learning-based D2D scheduling in UAV-assisted Industrial IoT networks," *IEEE Trans. Ind. Informat.*, vol. 20, no. 1, pp. 213–223, Jan. 2024.

[41] X. Tang, H. Zhang, R. Zhang, D. Zhou, Y. Zhang, and Z. Han, "Robust trajectory and offloading for energy-efficient UAV edge computing in Industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 20, no. 1, pp. 38–49, Jan. 2024.

[42] J.-H. Lee, J. Park, M. Bennis, and Y.-C. Ko, "Integrating LEO satellites and multi-UAV reinforcement learning for hybrid FSO/RF non-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 3647–3662, Mar. 2023.

[43] M. Elsayed and M. Erol-Kantarci, "Deep reinforcement learning for reducing latency in mission critical services," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2018, pp. 1–6.

[44] W. Ahsan, W. Yi, Z. Qin, Y. Liu, and A. Nallanathan, "Resource allocation in uplink NOMA-IoT networks: A reinforcement-learning approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5083–5098, Aug. 2021.

[45] N. Huang, T. Wang, Y. Wu, S. Bi, L. Qian, and B. Lin, "Delay minimization for intelligent reflecting surface assisted federated learning," *China Commun.*, vol. 19, no. 4, pp. 216–229, 2022.

[46] C. Qi, J. Wang, L. Lyu, L. Tan, J. Zhang, and G. Y. Li, "Key issues in wireless transmission for NTN-assisted Internet of Things," *IEEE Internet Things Mag.*, vol. 7, no. 1, pp. 40–46, Jan. 2024.

[47] M. Khalid, J. Ali, and B.-H. Roh, "Artificial intelligence and machine learning technologies for integration of terrestrial in non-terrestrial networks," *IEEE Internet Things Mag.*, vol. 7, no. 1, pp. 28–33, Jan. 2024.

[48] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5031–5044, May 2019.

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

[50] R. Duan, J. Wang, C. Jiang, H. Yao, Y. Ren, and Y. Qian, "Resource allocation for multi-UAV aided IoT NOMA uplink transmission systems," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7025–7037, Aug. 2019.

[51] Z. Yang, Z. Ding, P. Fan, and N. Al-Dhahir, "A general power allocation scheme to guarantee quality of service in downlink and uplink NOMA systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7244–7257, Nov. 2016.

[52] E. Bertran and A. Sànchez-Cerdà, "On the tradeoff between electrical power consumption and flight performance in fixed-wing UAV autopilots," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 8832–8840, Nov. 2016.

[53] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[54] M. A. B. S. Abir, M. Z. Chowdhury, and Y. M. Jang, "A software-defined UAV network using queueing model," *IEEE Access*, vol. 11, pp. 91423–91440, 2023.

[55] A. Khalili, E. M. Monfared, S. Zargari, M. R. Javan, N. M. Yamchi, and E. A. Jorswieck, "Resource management for transmit power minimization in UAV-assisted RIS HetNets supported by dual connectivity," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1806–1822, Mar. 2022.

[56] X. Yang, T. Cui, H. Wang, and Y. Ye, "Multiagent deep reinforcement learning for electric vehicle fast charging station pricing game in electricity-transportation Nexus," *IEEE Trans. Ind. Informat.*, early access, Jan. 4, 2024, doi: 10.1109/TII.2023.3345457.

[57] H. Lee and S.-W. Kim, "Task-oriented edge networks: Decentralized learning over wireless fronthaul," *IEEE Internet Things J.*, early access, Dec. 25, 2023, doi: 10.1109/JIOT.2023.3347234.

[58] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 12, 1999, pp. 1009–1014.

[59] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[60] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[61] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020.

[62] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Auton. Agents Multi-Agent Syst.*, vol. 11, no. 3, pp. 387–434, 2005.

[63] Y. Liu, J. Yan, and X. Zhao, "Deep-reinforcement-learning-based optimal transmission policies for opportunistic UAV-aided wireless sensor network," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13823–13836, Aug. 2022.

[64] X. Zhou, X. Zhang, H. Zhao, J. Xiong, and J. Wei, "Constrained soft actor-critic for energy-aware trajectory design in UAV-aided IoT networks," *IEEE Wireless Commun. Letters*, vol. 11, no. 7, pp. 1414–1418, Jul. 2022.

[65] H. Peng and L.-C. Wang, "Energy harvesting reconfigurable intelligent surface for UAV Based on robust deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 22, no. 10, pp. 6826–6838, Oct. 2023.

[66] C. J. Watkins and P. Dayan, "Q-Learning," *Mach. Learn.*, vol. 8, no. 3, pp. 279–292, 1992.

[67] Y. Shi et al., "Machine learning for large-scale optimization in 6G wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2088–2132, 4th Quart., 2023.

[68] J. Pei, P. Hong, M. Pan, J. Liu, and J. Zhou, "Optimal VNF placement via deep reinforcement learning in SDN/NFV-enabled networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 263–278, Feb. 2019.

[69] Y. Xue and W. Chen, "Multi-agent deep reinforcement learning for UAVs navigation in unknown complex environment," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 2290–2303, Jan. 2024.

[70] L. Liu, J. Feng, X. Mu, Q. Pei, D. Lan, and M. Xiao, "Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing," *IEEE Trans. Intell. Veh.*, vol. 24, no. 12, pp. 15513–15526, Dec. 2023.

[71] A. Zhu, H. Lu, S. Guo, Z. Zeng, and Z. Zhou, "CollOR: Distributed collaborative offloading and routing for tasks with QoS demands in multi-robot system," *Ad Hoc Netw.*, vol. 152, Jan. 2024, Art. no. 103311.

[72] K. Pan, B. Zhou, W. Zhang, and C. Ju, "Joint beamforming and phase shifts design for RIS-aided multi-user full-duplex systems in smart cities," *Sensors*, vol. 24, no. 1, p. 121, 2024.

[73] A. Gharehgoli, A. Nouruzi, N. Mokari, P. Azmi, M. R. Javan, and E. A. Jorswieck, "AI-based resource allocation in end-to-end network slicing under demand and CSI uncertainties," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 3, pp. 3630–3651, Sep. 2023.

[74] F. H. Panahi, F. H. Panahi, M. Ghaderzadeh, and A. Mohammadisarab, "M2M communications as a promising technique to support green powered base stations," in *Proc. 27th Iran. Conf. Electr. Eng. (ICEE)*, 2019, pp. 1654–1658.

[75] Y. Zhang, Z. Mou, F. Gao, J. Jiang, R. Ding, and Z. Han, "UAV-enabled secure communications by multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11599–11611, Oct. 2020.

[76] A. Mohammadisarab, A. Nouruzi, N. Mokari, B. Abbasi Arand, A. Khalili, and E. A. Jorswieck, 2023, "Source code of 6G disaster relief paper," Dataset, dataport. [Online]. Available: https://dx.doi.org/10.21227/xsb6-ah82

[77] Y. Hao, Z. Song, Z. Zheng, Q. Zhang, and Z. Miao, "Joint communication, computing, and caching resource allocation in LEO satellite MEC networks," *IEEE Access*, vol. 11, pp. 6708–6716, 2023.

[78] L. X. Nguyen, Y. K. Tun, T. N. Dang, Y. M. Park, Z. Han, and C. S. Hong, "Dependency tasks offloading and communication resource allocation in collaborative UAV networks: A metaheuristic approach," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 9062–9076, May 2023.

[79] S. Aboagye, T. M. Ngatched, O. A. Dobre, and H. V. Poor, "Energy-efficient resource allocation for aggregated RF/VLC systems," *IEEE Trans. Wireless Commun.*, vol. 22, no. 10, pp. 6624–6640, Oct. 2023.

[80] A. B. Rahman, J. Patrizi, P. Charatsaris, E. E. Tsiropoulou, and S. Papavassiliou, "Bioinspired dynamic spectrum management in 3D networks," in *Proc. 19th Int. Conf. Distrib. Comput. Smart Syst. Internet Things (DCOSS-IoT)*, 2023, pp. 166–170.

[81] Y. Sun, D. W. K. Ng, Z. Ding, and R. Schober, "Optimal joint power and subcarrier allocation for full-duplex multicarrier non-orthogonal multiple access systems," *IEEE Trans. Commun.*, vol. 65, no. 3, pp. 1077–1091, Mar. 2017.

AMIR MOHAMMADISARAB received the B.Sc. degree (with Hons.) in electronic engineering and telecommunication engineering from the University of Kurdistan in 2020, and the M.Sc. degree (with Hons.) in electronic engineering and telecommunication engineering from Tarbiat Modares University in 2023. He is currently employed as a Researcher with UWICORE in UMH de Elche, Spain. His research interests encompass wireless communication networks (6G), intelligent reflecting surface, nonterrestrial networks communications, Internet of Things, deep reinforcement learning, biomedical image processing, resource allocation, and optimization theory.

ALI NOURUZI received the M.Sc. degree in telecommunication engineering from Tarbiat Modares University, Tehran, Iran, where he has been with the Department of Electrical and Computer Engineering Since 2020. His research interests include wireless communication networks, networking, network function virtualization, software-defined network, machine learning, resource allocation, and optimization theory. He is a Reviewer of IEEE TRANSACTION ON COMMUNICATIONS and was selected as an exemplary reviewer in 2022.

ATA KHALILI (Member, IEEE) received the B.Sc. and M.Sc. degrees (Hons.) in electronic engineering and telecommunication engineering from Shahed University in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Institute for Digital Communications, Friedrich-Alexander University of Erlangen–Nürnberg, Erlangen, Germany. From 2018 to 2019, he was a Visiting Researcher with the Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran. From October 2019 to November 2021, he was a Research Assistant with the Department of Electrical and Computer Engineering, Tarbiat Modares University, Tehran. He also worked as a Research Assistant with the Electronics Research Institute, Sharif University of Technology, Tehran, from March 2020 to May 2021. His research interests include intelligent reflecting surface, integrated sensing and communication, unmanned aerial vehicle communications, resource allocation in wireless communication, green communication, mobile-edge computing, and optimization theory. He has been serving as a member of the Technical Program Committees for the IEEE GLOBECOM, IEEE WCNC, and IEEE ICC conferences, since 2020. He served as a Session Chair for IEEE GLOBECOM 2021 and IEEE ICC 2022. He is a Reviewer of several IEEE journals, such as IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He received the Best Paper Award from IEEE WPMC 2022 and recognition as an Exemplary Reviewer for IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE WIRELESS COMMUNICATION LETTERS.

NADER MOKARI (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Tarbiat Modares University, Tehran, Iran, in 2014. His thesis received the IEEE outstanding Ph.D. Thesis Award. He joined the Department of Electrical and Computer Engineering, Tarbiat Modares University as an Assistant Professor in October 2015. He has been elected as an IEEE Exemplary Reviewer in 2016 by IEEE Communications Society. He is currently an Associated Professor with the Department of Electrical and Computer Engineering, Tarbiat Modares University. His research interests cover many aspects of wireless technologies with a special emphasis on wireless networks. He is on the editorial board of the IEEE TRANSACTIONS ON COMMUNICATIONS. In recent years, his research has been funded by Iranian Mobile Telecommunication Companies, Iranian National Science Foundation (INSF). He received the Best Paper Award at ITU K-2020. He was also involved in a number of large scale network design and consulting projects in the telecom industry.

BIJAN ABBASI ARAND (Senior Member, IEEE) received the B.Sc. degree from Shiraz University, Shiraz, Iran, in 1995, and the M.S. and Ph.D. degrees in telecommunication engineering from Tarbiat Modares University, Tehran, Iran, in 1997 and 2003, respectively. From 2003 to 2005, he was a Researcher with the Electromagnetic Propagation Department, Iran Telecommunication Research Center, Tehran. In 2005, he joined the Satellite Communication Laboratory, Tarbiat Modares University, as a Postdoctoral Researcher, where he has been a Faculty Member with the Department of Electrical and Computer Engineering since 2010. He is currently conducting research in the areas of antennas and propagation, mobile network communications, and metasurfaces as an Associate Professor.

EDUARD A. JORSWIECK (Fellow, IEEE) received the Ph.D. degree in electrical engineering and computer science from TU Berlin in 2004. From 2006 to 2008, he was with the Signal Processing Group, KTH Stockholm, as a Postdoctoral Fellow and an Assistant Professor. From 2008 to 2019, he was the Chair of Communication Theory with TU Dresden. He is currently the Managing Director of the Institute of Communications Technology, the Head of the Chair for Communications Systems, and a Full Professor with Technische Universität Braunschweig, Brunswick, Germany. He has published more than 180 journal articles, 15 book chapters, one book, three monographs, and some 300 conference papers. His main research interests are in the broad area of communications. He was a recipient of the IEEE Signal Processing Society Best Paper Award. He and his colleagues were also recipients of the Best Paper and Best Student Paper Awards at the IEEE CAMSAP 2011, IEEE WCSP 2012, IEEE SPAWC 2012, IEEE ICUFN 2018, PETS 2019, and ISWCS 2019. Since 2017, he has been the Editor-in-Chief of the *EURASIP Journal on Wireless Communications and Networking*. He is currently serving on the editorial boards of the IEEE TRANSACTIONS ON INFORMATION THEORY and IEEE TRANSACTIONS ON COMMUNICATIONS. He was on the editorial boards of the IEEE SIGNAL PROCESSING LETTERS, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.