

Multi-objective Optimization of Space-Air-Ground Integrated Network Slicing Relying on a Pair of Central and Distributed Learning Algorithms

Guorong Zhou, Liqiang Zhao, *Member, IEEE*, Gan Zheng, *Fellow, IEEE*, Shenghui Song, *Senior Member, IEEE*, Jiankang Zhang, *Senior Member, IEEE*, and Lajos Hanzo, *Life Fellow, IEEE*

Abstract—As an attractive enabling technology for next-generation wireless communications, network slicing supports diverse customized services in the global space-air-ground integrated network (SAGIN) with diverse resource constraints. In this paper, we dynamically consider three typical classes of radio access network (RAN) slices, namely high-throughput slices, low-delay slices and wide-coverage slices, under the same underlying physical SAGIN. The throughput, the service delay and the coverage area of these three classes of RAN slices are jointly optimized in a non-scalar form by considering the distinct channel features and service advantages of the terrestrial, aerial and satellite components of SAGINs. A joint central and distributed multi-agent deep deterministic policy gradient (CDMADDPG) algorithm is proposed for solving the above problem to obtain the Pareto optimal solutions. The algorithm first determines the optimal virtual unmanned aerial vehicle (vUAV) positions and the inter-slice sub-channel and power sharing by relying on a centralized unit. Then it optimizes the intra-slice sub-channel and power allocation, and the virtual base station (vBS)/vUAV/virtual low earth orbit (vLEO) satellite deployment in support of three

classes of slices by three separate distributed units. Simulation results verify that the proposed method approaches the Pareto-optimal exploitation of multiple RAN slices, and outperforms the benchmarks.

Index Terms—Radio access network slicing, space-air-ground integrated network, multi-objective optimization, non-scalarization, hierarchical and distributed deep reinforcement learning.

I. INTRODUCTION

Given the explosive proliferation of user equipment and service categories in the emerging sixth generation (6G) era, traditional terrestrial wireless communication networks are facing huge challenges [1], [2]. Thus, the space-air-ground integrated networking (SAGIN) [3]–[5] concept was proposed for providing users with ubiquitous global coverage and enhanced data transmission. Specifically, satellite networks are capable of providing seamless access to rural or remote areas, *e.g.*, mountains, deserts and oceans. As a complement, aerial networks [6] are capable of providing prompt emergency communications in disaster areas, or providing service enhancement for areas having high tele-traffic density. Finally, the densely deployed terrestrial networks can support high data rate transmission in urban areas. By integrating the complementary advantages of the above three network components, SAGINs constitute a heterogeneous three-dimensional wireless communication architecture capable of providing seamless connections.

However, the variety of SAGIN-oriented service scenarios is increasing continually [3], [7]. That means a diverse variety of different services share the underlying spectral resources, thus maintaining the target quality of service (QoS) of heterogeneous applications is hard to guarantee. These services have distinct traffic features and may significantly affect each other, especially when the network resources are limited [8]. For example, bandwidth-extensive services are very likely to consume most of the channel resources, hence resulting in high transmission delays for lightweight delay-sensitive services. Clearly, there is a paucity of research on how to support the service diversity encountered in SAGINs. As an enabling technology of next-generation wireless networks, radio access network (RAN) slicing [9], [10] is potentially capable of constructing a set of independent virtual logical sub-networks based upon the same physical network infrastructure and resources, where each logical sub-network is customized for

This work was supported in part by the Key Research and Development Program of Shaanxi under Grant 2022KWZ-09, in part by the Postdoctoral Research Program of Shaanxi Province, in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B0101120003, in part by the Fundamental Research Funds for the Central Universities under Grant QTZX23031, and in part by the 111 Project under Grant B08038. The work of Shenghui Song was partially supported by a grant from the NSFC/RGC Joint Research Scheme sponsored by the Research Grants Council of the Hong Kong Special Administrative Region, China and National Natural Science Foundation of China (Project No. N_HKUST656/22). L. Hanzo would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council projects EP/W016605/1, EP/X01228X/1 and EP/Y026721/1 as well as of the European Research Council's Advanced Fellow Grant QuantCom (Grant No. 789028). (*Corresponding author: Liqiang Zhao and Lajos Hanzo.*)

Guorong Zhou is with the State Key Laboratory of Integrated Service Networks at Xidian University, Xi'an 710071, China (e-mail: guor_zhou@163.com).

Liqiang Zhao is with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China, and also with the Guangzhou Institute of Technology, Xidian University, Guangzhou 510100, China (e-mail: lqzhao@mail.xidian.edu.cn).

Gan Zheng is with the School of Engineering, University of Warwick, Coventry, CV4 7AL, UK (e-mail: gan.zheng@warwick.ac.uk).

Shenghui Song is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: eeshsong@ust.hk).

Jiankang Zhang is with Department of Computing & Informatics, Bournemouth University, BH12 5BB, UK (e-mail: jzhang3@bournemouth.ac.uk).

Lajos Hanzo is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK (e-mail: lh@ecs.soton.ac.uk).

Copyright (c) 20xx IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

a particular type of service. However, these network slices are isolated from each other. Therefore, RAN slicing may be applied to support diverse customized services in SAGINs under various resource constraints [3].

To elaborate, RAN slicing is conceived for supporting various services, hence it is inappropriate to optimize a sliced network based on a single criterion [11], [12]. For instance, bandwidth-extensive slicing aims for providing users with high-throughput services, while delay-sensitive slicing aims for ensuring ultra-low delay. However, the objectives of maintaining a high throughput and ultra-low delay tend to conflict with each other. To circumvent this conflict, multi-objective optimization problems (MOOPs) have been conceived as a mathematical framework for jointly optimizing multiple criteria [13], which are eminently suitable for RAN slicing scenarios.

There have been some studies on MOOPs in RAN slicing, but only in scalar form [14]–[18]. Although the traditional scalar method has low computational complexity, its objective function (OF) needs to be designed by experience, and its solution has a strong dependence on its weight. A better way to solve MOOPs in RAN slicing is to use non-scalar methods, *e.g.*, the popular multi-agent deep reinforcement learning (DR-L) algorithm [19], [20]. Instead of the unique globally optimal solution, the solutions of the non-scalar MOOP are composed of a number of non-dominated Pareto optimal solutions [21]. A multi-agent DRL algorithm is capable of learning different optimization objectives at the same time in the context of SAGINs, which makes it easier to obtain multiple Pareto optimal solutions [22], [23].

In this paper, we simultaneously establish three typical RAN slices, *i.e.*, high-throughput slices, low-delay slices and wide-coverage slices, based upon the same underlying physical SAGIN. A non-scalar MOOP is formulated for jointly optimizing the throughput, the service delay and the coverage area. This is achieved by dynamically assigning the most appropriate network components, by finding the virtual unmanned aerial vehicles (vUAVs) having the optimal positions, and by allocating the most suitable subchannels and powers to the users relying on each slice type. In order to solve the above non-scalar problem and to approach the Pareto optimal solutions, we propose a pair of central and distributed multi-agent deep deterministic policy gradient (CDMADDPG) algorithms. More explicitly, our proposed method simultaneously optimizes three SAGIN slices, and finds near-Pareto optimal solutions. Moreover, it succeeds in striking a compelling trade-off among three types of RAN slices by exploiting the non-dominance relationship amongst the Pareto optimal solutions. Against the above background, the main contributions of this paper are as follows:

- We apply dynamic RAN slices in a global SAGIN scenario to support diverse customized services under specific resource constraints. The system model is built for jointly supporting the high-throughput slices, the low-delay slices and the wide-coverage slices based upon the same underlying 3-dimensional (3D) SAGIN. Then, the proposed model takes the distinct channel characteristics and service advantages of terrestrial, aerial and satellite

communications in SAGIN into account, which is also verified by our simulation results.

- We formulate a non-scalar MOOP for jointly optimizing the throughput, the service delay and the coverage area based on RAN slicing, whilst the existing literature has only considered scalar MOOPs. To elaborate, we dynamically deploy virtual base stations (vBSs)/vUAVs/virtual low earth orbit (vLEO) satellites to support the users, and then find the optimal vUAVs' positions, allocate optimal subchannels and power resources among the slices, as well as optimize both the intra-slice subchannels and power resources for each user.
- We propose a CDMADDPG algorithm for solving the associated non-scalar MOOP. The algorithm first determines the position of the most suitable vUAVs and the associated inter-slice resource sharing, as decided by the centralized supervisory unit; then it proceeds to intra-slice resource sharing. This is arranged by the vBSs/vUAVs/vLEO satellites relying on three separate reduced-dimensional distributed units, in order to find near-Pareto optimal solutions. Our main contributions in comparison to the salient literature are summarized in Table 1.

The rest of the paper is organized as follows. In the next section, we briefly describe the related work of SAGIN and of MOOP based on RAN slicing. In Section III, three different types of RAN slices of our SAGIN are established. In Section IV, we formulate the proposed non-scalar MOOP of SAGIN slicing. In Section V, we solve the MOOP by conceiving our CDMADDPG algorithm. In Section VI, we discuss our simulation results. Finally, the paper is concluded in Section VII.

II. RELATED WORK

In this section, we first introduce the State of the Art in SAGIN, and then illustrate the popular methods of solving the MOOP based on RAN slicing.

A. SAGIN

SAGIN has been considered as one of the most promising techniques for increasing connectivity and improving the spatial 3D spectral efficiency of 6G system [24]. Substantial SAGIN research efforts have been focused on mobility management, traffic offloading and performance analysis [25]–[30]. For example, Sun *et al.* [26] have presented an enhanced air-ground integrated vehicular network (AGVN) by introducing a surveillance plane for improving the management capability and have discussed the related networking, security, and application-oriented aspects. Wang *et al.* [27] have proposed a data offloading scheme with drones acting as relays for maximizing the system capacity by jointly optimizing the users' connection scheduling, power control and drone trajectory. An attractive transmission control strategy has been proposed in [28] for optimally scheduling the ground-air-space and ground-to-space transmission schemes according to the specific status of ground users. Mao *et al.* [29] have considered a space-aerial-assisted mixed cloud-edge computing offloading

Table 1: Contrasting our contributions to the salient literature.

Novelty	[11]	[12]	[15]	[26]	[27]	[32]	[33]	Proposed
Studying the SAGIN				✓	✓		✓	✓
Studying a dynamic sliced network	✓	✓	✓			✓		✓
Optimizing the vUAVs' positions				✓	✓			✓
Optimizing network components' assignment to users				✓	✓	✓	✓	✓
Optimization of inter-slice resources (<i>e.g.</i> , power, subchannels)	✓	✓						✓
Optimization of intra-slice resources between users (<i>e.g.</i> , power, subchannels)	✓	✓	✓		✓			✓
Jointly optimizing at least 3 metrics						✓		✓
Formulating a non-scalar MOOP						✓	✓	✓
Solving the non-scalar MOOP by the multi-agent DRL algorithm								✓
Solving the non-scalar MOOP by a pair of central and distributed algorithms								✓

framework through jointly scheduling the associated position deployment and resource allocation. Ye *et al.* [30] have constructed a cooperative channel model of SAGINs including the space-air, space-ground as well as air-ground links, and analyzed the outage performance attained.

However, while the variety of SAGIN-oriented service scenarios is increasing, the existing physical network architecture struggles to provide customized services for all users. The next-generation enabling technology, RAN slicing, is capable of constructing a set of independent virtual logical sub-networks based upon the same physical network, with the promise of meeting the above challenge. Nevertheless, at the time of writing, there are few studies on constructing network slices for SAGIN.

B. MOOP based on RAN Slicing

The key to solving a MOOP is to find Pareto optimal solutions, which are non-dominated by each other. Generally, there are several methods available on RAN slicing, *e.g.*, the traditional scalar method and the family of non-scalar intelligent algorithms.

The traditional scalar method mainly includes the weighted sum method, constraint method, etc [14]. This method aims to transform the MOOP into a single-objective optimization problem (SOOP), the optimal solution of which subjectively corresponds to one of the Pareto solutions. Specifically, Shi *et al.* [15] have simultaneously analyzed the energy efficiency (EE) and delay in wireless network virtualization, but only the EE was employed as the objective, while the delay was treated as a constraint. Afolabi *et al.* [16] have also used the constrained optimization method for minimizing the amount of computing resources subject to the constraint of a maximum tolerable mean response time on the end-to-end network slicing. Although both the throughput and delay are considered in [17], the authors have used the weighted sum to project the above pair of metrics into a unified utility function. Wang *et al.* [18] have integrated three optimization criteria into the slices' profits under a hybrid slicing framework. However, these scalar methods have some drawbacks. Explicitly, on the one hand, their OFs are subjectively designed based on experience, so it is hard to obtain a satisfactory solution in practice. On the other hand, the optimization result of each slice is highly dependent on its weight in the OF, which requires extensive prior knowledge.

The non-scalar intelligent algorithm mainly includes the multi-objective evolutionary algorithms (MOEAs) [31] and

the multi-agent DRL algorithms. Specifically, MOEAs apply evolutionary operations to a population of legitimate candidates for finally approaching the Pareto front of optimal solutions. For instance, Chantre *et al.* [32] employed MOEAs to solve the multi-criterion optimization problem built on 5G network slicing. In the context of aeronautical ad hoc networks, Cui *et al.* [33] formulated a twin-objective multi-hop routing problem and obtained multiple tradeoff solutions by MOEA. Nevertheless, the existing literature only solved static and discrete MOOPs by employing MOEAs.

A multi-agent deep deterministic policy gradient (MADDPG) algorithm [19] is capable of simultaneously learning different optimization objectives in a dynamic and successive SAGIN, but it may become computationally demanding in large-scale multi-user scenarios due to its single-layer and agent-coupled structure. This will result in excessive complexity once the action set becomes excessively large in the training. Hence the algorithm may fail to converge and the computational complexity becomes excessive. Therefore we decompose the coupled structure of the MADDPG algorithm, and propose the CDMADDPG algorithm, where the heavy intra-slice resource decisions and training can be performed locally in the distributed units, and only their decision results have to be transmitted to the centralized supervisory unit for global processing. Furthermore, by defining multiple agents within the distributed units as different types of RAN slices, the CDMADDPG algorithm succeeds in the simultaneous optimization of multiple objectives.

III. SYSTEM MODEL

Again, we focus our attention on analyzing three typical classes of RAN slices [34] based upon the SAGIN of Fig. 1. Firstly, on the left side of Fig. 1, we have a triple-layer physical RAN, constituted by the ground layer of M base stations (BSs), the aerial layer of V unmanned aerial vehicles (UAVs) and the space layer of a satellite constellation. The UAVs are used as flying BSs [35], [36] and the satellite constellation is constituted by a group of low earth orbit (LEO) satellites to provide the full coverage of planet Earth. The sets of M and V nodes are denoted by $\mathbf{M} = \{1, \dots, M\}$ and $\mathbf{V} = \{1, \dots, V\}$, respectively. A total of K terrestrial users may be served by different RAN slices, with the set of $\mathbf{K} = \{1, \dots, K\}$. We assume that all the terrestrial, airborne and spaceborne cellular downlink channels are operated in the C-band at 5 GHz, and the channels of different layers do not

overlap with each other to avoid cross-layer interference [37]. The bandwidth of each layer is set to B Hz, divided into N subchannels associated with $\mathbf{N} = \{1, \dots, N\}$, respectively. Hence, each subchannel has a bandwidth of B/N Hz, allowing the BSs/UAVs/LEOs to provide downlink communication for the users. To reduce the model complexity, this paper does not consider the communication between network components (such as between the satellite and UAVs) [8], [29]. The orthogonal frequency division multiple access (OFDMA) method is used for avoiding intra-cell interference. We assume that the available power of each physical BS, UAV and LEO satellite is P_B , P_V and P_L , respectively.

Then, with the aid of virtualization, we can support three typical classes of RAN slices, namely the high-throughput slices, the low-delay slices and the wide-coverage slices, which are referred to as class- s slices, $s \in \{1, 2, 3\}$, $S=3$. As shown at the right of Fig. 1, the network slicing process can be classified into three steps:

- 1) All communication resources and components are dynamically shared among the three classes of slices. Every physical BS/UAV/LEO is mapped into S vBSs/vUAVs/vLEOs, each of which is associated with a specific class- s slice [17]. Naturally, the locations of the vBSs/vUAVs/vLEOs supporting the class- s slices correspond to those of the physical BSs/UAVs/LEOs, thus we represent them by the same index $m, \forall m \in \mathbf{M}$ or $v, \forall v \in \mathbf{V}$ as those of the physical ones.
- 2) Both the spectral and power resources of our 3D system are abstracted into the shared virtual resources.
- 3) Depending on the different services' specific demands, the mobile network operator (MNO) orchestrates the flexible deployment of vBSs/vUAVs/vLEOs (also referred to as network components hereafter) as well as supports the real-time sharing and release of their associated communication resources among the different slices.

We assume that K_s users would like to request class- s slices, with $\mathbf{K}_s = \{1, \dots, K_s\}$, $\mathbf{K}_s \subset \mathbf{K}$. A dynamically adaptive multi-slicing network is considered, where the time slots (TSs) are defined as the time interval $[t, t+1)$, $t \in \{0, 1, 2, \dots, T-1\}$, where T is the total number of TSs and the TS duration is δ second. In our model, it is assumed that the positions of vUAVs and users can change in a certain area from TS to TS, but they are stationary within a TS. The users' positions follow a dynamic random distribution with a density ratio of 5:1 in densely populated and sparsely populated areas between TSs, respectively; the vUAVs' positions will be dynamically optimized in the proposed optimization problem.¹ Naturally, the positions of vBSs remain fixed. We also assume that the vLEO satellite index does not change within a total TS length T and that a single satellite can be relied upon in our scenario [27]. Therefore, we denote the 3D coordinates of the vLEO satellite, vUAVs, terrestrial vBSs and users as $[x^{vLEO}(t), y^{vLEO}(t), z^{vLEO}(t)]$, $[x_v^{vUAV}(t), y_v^{vUAV}(t), z_v^{vUAV}(t)]$, $[x_m^{vBS}, y_m^{vBS}, z_m^{vBS}]$ and $[x_{k,s}^{UE}(t), y_{k,s}^{UE}(t), z_{k,s}^{UE}(t)]$, respectively, where

¹The users' and vUAVs' motion trajectory models are beyond the scope of this paper's discussions [29].

$z_m^{vBS} = z_{k,s}^{UE}(t) = 0$. The main notations of this paper are summarized in Table 2.

A. Terrestrial Communication

In the terrestrial layer, it is assumed that the vBS provides customized services for multiple users. The channel coefficient $g_{k,s,m,n}^{vBS}(t)$ of user $k, \forall k \in \mathbf{K}_s$ on subchannel $n, \forall n \in \mathbf{N}$ associated with vBS m on the class- s slices at TS t is independent and identically distributed (*i.i.d.*) over time, which is modeled as:

$$g_{k,s,m,n}^{vBS}(t) = h_{k,s,m,n}^{vBS}(t) [d_{k,s,m}^{vBS}(t)]^{-\alpha}, \quad (1)$$

and

$$d_{k,s,m}^{vBS}(t) = \left[(x_m^{vBS} - x_{k,s}^{UE}(t))^2 + (y_m^{vBS} - y_{k,s}^{UE}(t))^2 \right]^{1/2}, \quad (2)$$

where $d_{k,s,m}^{vBS}(t)$ is the distance between the vBS m and user k of class- s slices. We adopt the classic Rayleigh fading channel model [8], [30], and the channel fading of $h_{k,s,m,n}^{vBS}(t)$ follows an exponential distribution with unity mean, while α is the path loss exponent.

Then, we define $p_{k,s,m,n}^{vBS}(t)$ as the power received from vBS m by user k associated with the subchannel n on the class- s slices at TS t . Then we have $\mathbf{P}_s^{vBS}(t) = \{p_{k,s,m,n}^{vBS}(t)\}$. Therefore, the data rate $r_{k,s,m,n}^{vBS}(t)$ of user k on subchannel n associated with vBS m on class- s slices at TS t is given by

$$r_{k,s,m,n}^{vBS}(t) = \frac{B}{N} \log_2 \left[1 + \frac{p_{k,s,m,n}^{vBS}(t) g_{k,s,m,n}^{vBS}(t)}{I_{k,s,m,n}^{vBS}(t) + \left(\frac{B}{N}\right) N_0} \right], \quad (3)$$

where

$$I_{k,s,m,n}^{vBS}(t) = \sum_{j \in \mathbf{M} \setminus \{m\}} p_{k,s,j,n}^{vBS}(t) g_{k,s,j,n}^{vBS}(t). \quad (4)$$

In (3), N_0 is the power spectral density of the additive white Gaussian noise (AWGN) and $I_{k,s,m,n}^{vBS}(t)$ is defined as the inter-cell interference of user k associated with vBS m and subchannel n on class- s slices at TS t . Moreover, $g_{k,s,j,n}^{vBS}(t)$ is the channel coefficient of the interfering vBS $j, \forall j \in \mathbf{M} \setminus \{m\}$; and $p_{k,s,j,n}^{vBS}(t)$ is the interference power of vBS j .

B. UAV Communication

In the aerial layer, UAVs can furnish users with various types of slices through their downlink channels. Similarly, the channel coefficient $g_{k,s,v,n}^{vUAV}(t)$ of user k on subchannel n associated with vUAV v on class- s slices at TS t is also *i.i.d.* vs. time, formulated as follows:

$$\begin{aligned} g_{k,s,v,n}^{vUAV}(t) &= h_{k,s,v,n}^{vUAV}(t) [d_{k,s,v}^{vUAV}(t)]^{-\alpha} \\ &= h_0 [d_{k,s,v}^{vUAV}(t)]^{-\alpha} \left(\frac{R}{R+1} \hat{h}_{k,s,v,n}^{vUAV}(t) + \frac{1}{R+1} \tilde{h}_{k,s,v,n}^{vUAV}(t) \right), \end{aligned} \quad (5)$$

and

$$\begin{aligned} d_{k,s,v}^{vUAV}(t) &= \left[(x_v^{vUAV}(t) - x_{k,s}^{UE}(t))^2 \right. \\ &\quad \left. + (y_v^{vUAV}(t) - y_{k,s}^{UE}(t))^2 + (z_v^{vUAV}(t))^2 \right]^{1/2}, \end{aligned} \quad (6)$$

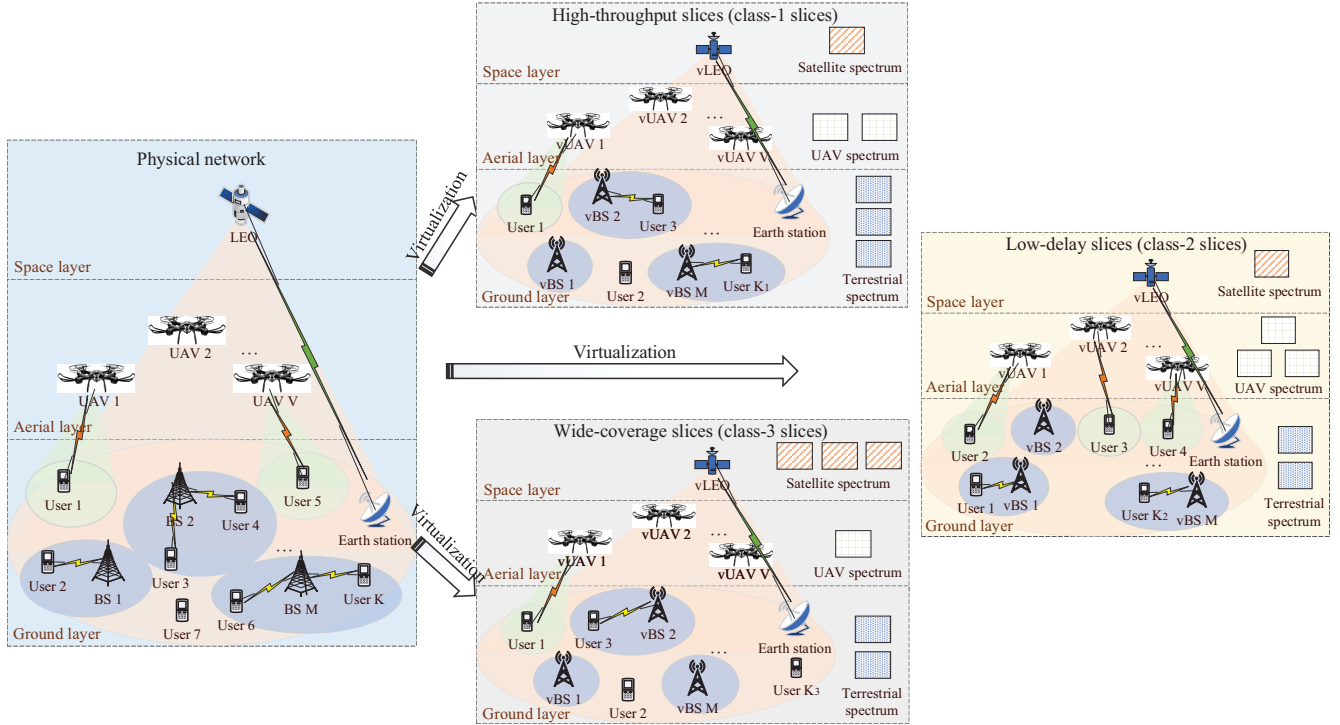


Fig. 1: System model of SAGIN slicing.

Table 2: Main notations of this paper.

Notation	Description
$\mathbf{M} = \{1, \dots, m, \dots, M\}$	Set of vBSs
$\mathbf{V} = \{1, \dots, v, \dots, V\}$	Set of vUAVs
$\mathbf{K} = \{1, \dots, k, \dots, K\}$	Set of users
P_B, P_V, P_L	Available power of each physical BS/UAV/LEO satellite
B	System bandwidth
$\mathbf{N} = \{1, \dots, n, \dots, N\}$	Set of subchannels
$\mathbf{K}_s = \{1, \dots, K_s\}, \mathbf{K}_s \subset \mathbf{K}$	Set of users requesting class- s slices
t, T, δ	TS; Total number of TSs; TS duration
$g_{k,s,m,n}^{vBS}(t), g_{k,s,v,n}^{vUAV}(t), g_{k,s,n}^{vLEO}(t)$	The channel coefficient of user k on subchannel n associated with vBS m /vUAV n /vLEO on class- s slices at TS t
$d_{k,s,m}^{vBS}(t), \mathbf{d}_s^{vUAV} = \{d_{k,s,v}^{vUAV}(t)\}, d_{k,s}^{vLEO}(t)$	The distance between vBS m /vUAV n /vLEO and user k
$\mathbf{P}_s = \{\mathbf{P}_s^{vBS}(t), \mathbf{P}_s^{vUAV}(t), \mathbf{P}_s^{vLEO}(t)\}$	Set of transmit powers
$\Xi_s(t) = \{\xi_{k,s,n}(t)\}$	Set of intra-slice subchannel allocation indicators
$\Phi_s(t) = \{\varphi_{k,s,m(v)}(t)\}, \varphi_{k,s,m(v)}(t) \in \{\varphi_{k,s,m}^{vBS}(t), \varphi_{k,s,v}^{vUAV}(t), \varphi_{k,s}^{vLEO}(t)\}$	Set of intra-slice network component association indicators
$r_{k,s,m,n}^{vBS}(t), R_{k,s}^{vBS}(t); r_{k,s,v,n}^{vUAV}(t), R_{k,s}^{vUAV}(t); r_{k,s,n}^{vLEO}(t), R_{k,s}^{vLEO}(t)$	Data rate and total data rate for user k associated with vBSs/vUAVs/vLEO satellite on class- s slices
$I_{k,s,m,n}^{vBS}(t), I_{k,s,v,n}^{vUAV}(t)$	Inter-cell interference of user k associated with vBSs/vUAVs
$R_{k,s}(t), R_1^{sum}(t)$	Total downlink data rate of user k on class- s slices; Throughput of class-1 slices at TS t
$D_{k,2}(t), D_2^{ave}(t)$	Service delay of user k , average delay of all users on class-2 slices
$SINR_{k,3}^{vBS}(t), SINR_{k,3}^{vUAV}(t), SINR_{k,3}^{vLEO}(t), SINR_3^{ave}(t)$	SINR of user k associated with vBSs/vUAVs/vLEO satellite on class-3 slices; average SINR on class-3 slices
$P_{m,s}^{vBS}, P_{v,s}^{vUAV}, P_s^{vLEO}$	Power consumption of each vBS/vUAV/vLEO satellite
β	Initial threshold of average delay objective
$\rho_s = \{\rho_s^{vBS}(t), \rho_s^{vUAV}(t), \rho_s^{vLEO}(t)\}$	Set of inter-slice power allocation factors
$\eta_s = \{\eta_s^{vBS}(t), \eta_s^{vUAV}(t), \eta_s^{vLEO}(t)\}$	Set of inter-slice subchannel allocation factors

where $d_{k,s,v}^{vUAV}(t)$ is the distance between the vUAV v and user k relying on class- s slices. We have $\mathbf{d}_s^{vUAV} = \{d_{k,s,v}^{vUAV}(t)\}$ as one of the decision variables, because it dynamically changes between TSs. The fading of $h_{k,s,v,n}^{vUAV}(t)$ is assumed to follow a Rician channel model [29], [38], in which h_0 denotes the reference channel gain when the distance is 1 meter; R is

the Rician fading factor; $\hat{h}_{k,s,v,n}^{vUAV}(t)$ represents the line-of-sight (LoS) component that satisfies $|\hat{h}_{k,s,v,n}^{vUAV}(t)| = 1$; and $\tilde{h}_{k,s,v,n}^{vUAV}(t)$ is the non-line-of-sight (NLoS) component that follows $\tilde{h}_{k,s,v,n}^{vUAV}(t) \sim \mathcal{CN}(0, 1)$.

Furthermore, $p_{k,s,v,n}^{vUAV}(t)$ is the power transmitted from vUAV v to user k associated with subchannel n on class- s

slices at TS t , with the set as $\mathbf{P}_s^{v\text{UAV}}(t) = \{p_{k,s,v,n}^{v\text{UAV}}(t)\}$. The data rate $r_{k,s,v,n}^{v\text{UAV}}(t)$ of user k on subchannel n associated with vUAV v on class- s slices at TS t is given by

$$r_{k,s,v,n}^{v\text{UAV}}(t) = \frac{B}{N} \log_2 \left[1 + \frac{p_{k,s,v,n}^{v\text{UAV}}(t) g_{k,s,v,n}^{v\text{UAV}}(t)}{I_{k,s,v,n}^{v\text{UAV}}(t) + \left(\frac{B}{N}\right) N_0} \right], \quad (7)$$

where

$$I_{k,s,v,n}^{v\text{UAV}}(t) = \sum_{i \in \mathbf{V} \setminus \{v\}} p_{k,s,i,n}^{v\text{UAV}}(t) g_{k,s,i,n}^{v\text{UAV}}(t). \quad (8)$$

Similarly, $I_{k,s,v,n}^{v\text{UAV}}(t)$ is the inter-cell interference of user k associated with vUAV v and subchannel n on class- s slices at TS t , where $g_{k,s,i,n}^{v\text{UAV}}(t)$ and $p_{k,s,i,n}^{v\text{UAV}}(t)$ are the channel coefficient and the power of the interference vUAV i , $\forall i \in \mathbf{V} \setminus \{v\}$, respectively.

C. Satellite Communication

With the aid of LEO satellite downlink channels, seamless coverage may be provided for users in our slicing network. In the space layer, the channel coefficient $g_{k,s,n}^{v\text{LEO}}(t)$ of user k on subchannel n associated with the vLEO satellite on class- s slices at TS t is defined as:

$$\begin{aligned} g_{k,s,n}^{v\text{LEO}}(t) &= h_{k,s,n}^{v\text{LEO}}(t) [d_{k,s}^{v\text{LEO}}(t)]^{-\alpha} \\ &= \left(\frac{c}{4\pi f_c} \right)^2 [d_{k,s}^{v\text{LEO}}(t)]^{-\alpha}, \end{aligned} \quad (9)$$

and

$$\begin{aligned} d_{k,s}^{v\text{LEO}}(t) &= \left[(x^{v\text{LEO}}(t) - x_{k,s}^{UE}(t))^2 \right. \\ &\quad \left. + (y^{v\text{LEO}}(t) - y_{k,s}^{UE}(t))^2 + (z^{v\text{LEO}}(t))^2 \right]^{1/2}, \end{aligned} \quad (10)$$

where $d_{k,s}^{v\text{LEO}}(t)$ is the distance between the vLEO satellite and user k of class- s slices. Furthermore, $h_{k,s,n}^{v\text{LEO}}(t)$ is the unit radio propagation loss of the satellite link caused by the free space loss [37], where f_c is the carrier frequency and c is the velocity of light.

We denote the transmit power of the vLEO satellite to user k associated with subchannel n on class- s slices at TS t by $p_{k,s,n}^{v\text{LEO}}(t)$, having $\mathbf{P}_s^{v\text{LEO}}(t) = \{p_{k,s,n}^{v\text{LEO}}(t)\}$. Thus, the data rate $r_{k,s,n}^{v\text{LEO}}(t)$ of user k on subchannel n associated with the vLEO satellite on class- s slices at TS t is given by

$$r_{k,s,n}^{v\text{LEO}}(t) = \frac{B}{N} \log_2 \left[1 + \frac{p_{k,s,n}^{v\text{LEO}}(t) g_{k,s,n}^{v\text{LEO}}(t)}{\left(\frac{B}{N}\right) N_0} \right]. \quad (11)$$

Subsequently, we define $\xi_{k,s,n}(t)$ as a binary factor of user k associated with subchannel n on class- s slices at TS t , and $\Xi_s(t) = \{\xi_{k,s,n}(t)\}$ is the set of subchannel allocation indicators. Furthermore, $\Phi_s(t) = \{\varphi_{k,s,m(v)}(t)\}$ represents the set of network component association indicators, where $\varphi_{k,s,m(v)}(t) \in \{\varphi_{k,s,m}^{v\text{BS}}(t), \varphi_{k,s,v}^{v\text{UAV}}(t), \varphi_{k,s}^{v\text{LEO}}(t)\}$ is a binary factor of user k associated with vBS m , vUAV v or the vLEO satellite on class- s slices at TS t . Their specific expressions are as follows:

$$\xi_{k,s,n}(t) = \begin{cases} 1, & \text{if } n \text{ is allocated to user } k \text{ of class-} s \text{ slices,} \\ 0, & \text{otherwise,} \end{cases} \quad (12)$$

and

$$\varphi_{k,s,m(v)}(t) = \begin{cases} 1, & \text{if user } k \text{ of class-} s \text{ slices is associated} \\ & \text{with } m, v \text{ or vLEO,} \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

Hence, the downlink data rate of user k associated with vBSs, vUAVs and the vLEO satellite on class- s slices at TS t are formulated as, respectively:

$$R_{k,s}^{v\text{BS}}(t) = \sum_{n \in \mathbf{N}} \sum_{m \in \mathbf{M}} \varphi_{k,s,m}^{v\text{BS}}(t) \xi_{k,s,n}(t) r_{k,s,m,n}^{v\text{BS}}(t), \quad (14)$$

$$R_{k,s}^{v\text{UAV}}(t) = \sum_{n \in \mathbf{N}} \sum_{v \in \mathbf{V}} \varphi_{k,s,v}^{v\text{UAV}}(t) \xi_{k,s,n}(t) r_{k,s,v,n}^{v\text{UAV}}(t), \quad (15)$$

and

$$R_{k,s}^{v\text{LEO}}(t) = \sum_{n \in \mathbf{N}} \varphi_{k,s}^{v\text{LEO}}(t) \xi_{k,s,n}(t) r_{k,s,n}^{v\text{LEO}}(t). \quad (16)$$

Upon considering our triple-layer RAN slicing system including the space, air and ground layers, we can get the total downlink data rate of user k on class- s slices at TS t , as follows:

$$R_{k,s}(t) = R_{k,s}^{v\text{BS}}(t) + R_{k,s}^{v\text{UAV}}(t) + R_{k,s}^{v\text{LEO}}(t). \quad (17)$$

D. SAGIN Slices

Let us now build the customized optimization objectives of three classes of slices, respectively. As for class-1 slices, such as a high-resolution video streaming service, MNO has to satisfy the users' requirements by transmitting high-throughput data through the wireless downlink. Hence, the throughput of the system is regarded as the customized optimization objective for class-1 slices. We express the throughput of class-1 slices at TS t as:

$$R_1^{sum}(t) = \sum_{k \in \mathbf{K}_1} R_{k,1}(t). \quad (18)$$

As for class-2 slices, such as mission critical services and emergency response, the primary target is to reduce the service delay in order to guarantee flawless lip-synchronization. We represent by $A_2(t) = \{A_{1,2}(t), A_{2,2}(t), \dots, A_{k,2}(t), \dots\}$ the process of random data arrivals on class-2 slices at TS t , which follows a Poisson arrival process having the average arrival rate of λ_2 . In this context, $A_2(t)$ is assumed to be independent among the users and *i.i.d.* across the TSs. Consequently, the service procedure at TS t can be modeled as an M/D/1 queue [39], [40] having the service rate of $R_{k,2}(t)$. The service delay of user k on class-2 slices at TS t is denoted by $D_{k,2}(t)$, which consists of the propagation delay, the transmission delay and the queuing delay, yielding:

$$D_{k,2}(t) = \frac{d_{k,2}(t)}{c} + \frac{A_{k,2}(t)}{R_{k,2}(t)} + \frac{\lambda_2 A_{k,2}(t)}{2 \left[\left(R_{k,2}(t) \right)^2 - \lambda_2 R_{k,2}(t) \right]}, \quad (19)$$

where we have:

$$\begin{aligned}
d_{k,2}(t) &= \sum_{n \in \mathbf{N}} \sum_{m \in \mathbf{M}} \varphi_{k,2,m}^{vBS}(t) \xi_{k,2,n}(t) d_{k,2,m}^{vBS}(t) \\
&+ \sum_{n \in \mathbf{N}} \sum_{v \in \mathbf{V}} \varphi_{k,2,v}^{vUAV}(t) \xi_{k,2,n}(t) d_{k,2,v}^{vUAV}(t) \\
&+ \sum_{n \in \mathbf{N}} \varphi_{k,2}^{vLEO}(t) \xi_{k,2,n}(t) d_{k,2}^{vLEO}(t).
\end{aligned} \quad (20)$$

The first term of (19) is the propagation delay of user k on class-2 slices at TS t , where $d_{k,2}(t)$ is the distance between the user k on class-2 slices and its associated network component. The second term of (19) is the transmission delay of user k on class-2 slices at TS t , and the last term of (19) is the queuing delay of user k on class-2 slices at TS t .

The average delay of all users on class-2 slices can be expressed as

$$D_2^{ave}(t) = \frac{1}{K_2} \sum_{k \in \mathbf{K}_2} D_{k,2}(t). \quad (21)$$

In order to meet the low-delay requirements of users on class-2 slices, we aim for minimizing $D_2^{ave}(t)$.

Class-3 slices provide the basic wide-area access services at a low user density, in rural areas supporting few people. Given the time-varying channel state and the heterogeneous 3D network architecture, the basic access services are also of variable-rate nature. Consequently, the main optimization objective of class-3 slices is to maximize the average signal to interference plus noise power ratio (SINR) of the network for ensuring a wide coverage area. The SINR of user k associated with vBSs, vUAVs and vLEO satellite on class-3 slices at TS t are defined respectively as:

$$\begin{aligned}
&SINR_{k,3}^{vBS}(t) \\
&= \sum_{n \in \mathbf{N}} \sum_{m \in \mathbf{M}} \varphi_{k,3,m}^{vBS}(t) \xi_{k,3,n}(t) \frac{p_{k,3,m,n}^{vBS}(t) g_{k,3,m,n}^{vBS}(t)}{I_{k,3,m,n}^{vBS}(t) + (\frac{B}{N})N_0}, \quad (22) \\
&SINR_{k,3}^{vUAV}(t) \\
&= \sum_{n \in \mathbf{N}} \sum_{v \in \mathbf{V}} \varphi_{k,3,v}^{vUAV}(t) \xi_{k,3,n}(t) \frac{p_{k,3,v,n}^{vUAV}(t) g_{k,3,v,n}^{vUAV}(t)}{I_{k,3,v,n}^{vUAV}(t) + (\frac{B}{N})N_0}, \quad (23)
\end{aligned}$$

and

$$SINR_{k,3}^{vLEO}(t) = \sum_{n \in \mathbf{N}} \varphi_{k,3}^{vLEO}(t) \xi_{k,3,n}(t) \frac{p_{k,3,n}^{vLEO}(t) g_{k,3,n}^{vLEO}(t)}{(\frac{B}{N})N_0}. \quad (24)$$

By jointly considering our SAGIN slicing network, we express the average SINR of class-3 slices as:

$$\begin{aligned}
SINR_3^{ave}(t) &= \frac{1}{K_3} \sum_{k \in \mathbf{K}_3} \{SINR_{k,3}^{vBS}(t) + SINR_{k,3}^{vUAV}(t) \\
&+ SINR_{k,3}^{vLEO}(t)\}.
\end{aligned} \quad (25)$$

IV. MULTI-OBJECTIVE PROBLEM FORMULATION

Before starting this section, we provide Fig. 2 to emphasize the flow of the mathematical analysis in the sequel. This diagram helps readers to understand the flow of the paper.

Explicitly, we formulate the power consumption of each vBS, vUAV and vLEO satellite on class- s slices respectively

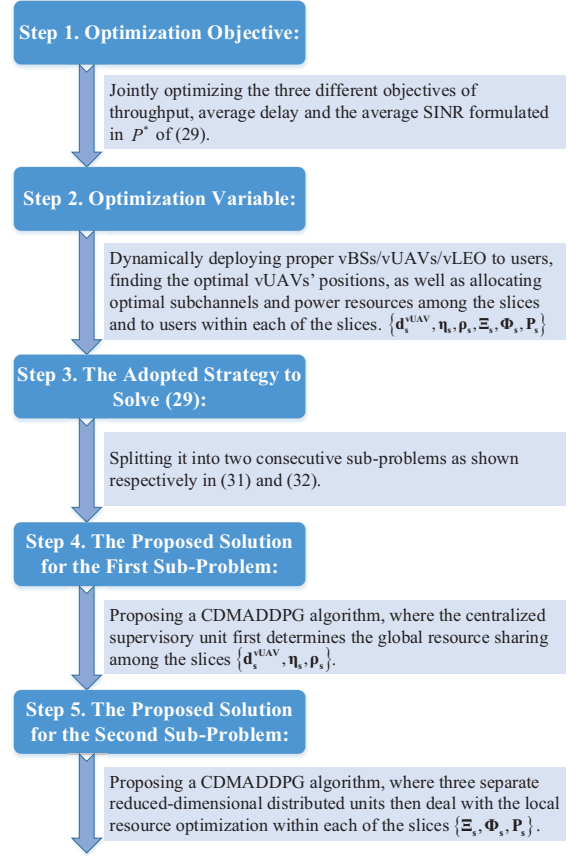


Fig. 2: Flow of the mathematical analysis.

as:

$$P_{m,s}^{vBS}(t) = \sum_{n \in \mathbf{N}} \sum_{k \in \mathbf{K}_s} \varphi_{k,s,m}^{vBS}(t) \xi_{k,s,n}(t) p_{k,s,m,n}^{vBS}(t), \quad (26)$$

$$P_{v,s}^{vUAV}(t) = \sum_{n \in \mathbf{N}} \sum_{k \in \mathbf{K}_s} \varphi_{k,s,v}^{vUAV}(t) \xi_{k,s,n}(t) p_{k,s,v,n}^{vUAV}(t), \quad (27)$$

and

$$P_s^{vLEO}(t) = \sum_{n \in \mathbf{N}} \sum_{k \in \mathbf{K}_s} \varphi_{k,s}^{vLEO}(t) \xi_{k,s,n}(t) p_{k,s,n}^{vLEO}(t). \quad (28)$$

Next, we define the set of inter-slice power allocation factors of the vBSs, vUAVs and vLEO satellite in class- s slices at TS t as $\rho_s = \{\rho_s^{vBS}(t), \rho_s^{vUAV}(t), \rho_s^{vLEO}(t)\}$. Furthermore, $\eta_s = \{\eta_s^{vBS}(t), \eta_s^{vUAV}(t), \eta_s^{vLEO}(t)\}$ represents the set of inter-slice subchannel allocation factors of the vBSs, vUAVs and vLEO satellite in class- s slices at TS t .

As for our dynamically adaptive multi-slicing SAGIN, it is more effective to adopt the long-term objectives to estimate the network performance instead of optimizing near-instantaneous objectives [41]. By (18), (21) and (25), we can jointly optimize the three different objectives of throughput, average delay and the average SINR relying on multi-objective optimization (MOO) algorithms in the non-scalar form, which is shown in P^* of (29) at the top of the next page.

In this context, we define:

$$\mathbf{P}_s = \{\mathbf{P}_s^{vBS}(t), \mathbf{P}_s^{vUAV}(t), \mathbf{P}_s^{vLEO}(t)\}. \quad (30)$$

Furthermore, in (29), β is the initial threshold used for guaranteeing the non-negativity of the delay objective. In (29),

$$\begin{aligned}
\mathbf{P}^* : & \max_{\left\{ \begin{array}{l} \Xi_s, \Phi_s, \mathbf{P}_s, \\ \eta_s, \rho_s, \mathbf{d}_s^{vUAV} \end{array} \right\}} \left\{ \begin{array}{l} \sum_{t=1}^T R_1^{sum}(t) \\ \sum_{t=1}^T [\beta - D_2^{ave}(t)] \\ \sum_{t=1}^T SINR_3^{ave}(t) \end{array} \right\}, \\
\text{s.t. } C1 : & \sum_{k \in \mathbf{K}_s} \xi_{k,s,n}(t) \varphi_{k,s,m(v)}(t) \leq 1, \forall n \in \mathbf{N}, m \in \mathbf{M}, v \in \mathbf{V}, \\
C2 : & \sum_{n \in \mathbf{N}} \sum_{\substack{m \in \mathbf{M} \\ (v \in \mathbf{V})}} \xi_{k,s,n}(t) \varphi_{k,s,m(v)}(t) \leq 1, \forall k \in \mathbf{K}, \\
C3 : & \sum_{m \in \mathbf{M}} \varphi_{k,s,m}^{vBS}(t) + \sum_{v \in \mathbf{V}} \varphi_{k,s,v}^{vUAV}(t) + \varphi_{k,s}^{vLEO}(t) \leq 1, \forall k \in \mathbf{K}, \\
C4 : & \rho_s, \eta_s \in (0, 1), \forall s \in \{1, 2, 3\}, \\
C5 : & p_{k,s,m,n}^{vBS}(t), p_{k,s,v,n}^{vUAV}(t), p_{k,s,n}^{vLEO}(t) \geq 0, \forall k \in \mathbf{K}, n \in \mathbf{N}, m \in \mathbf{M}, v \in \mathbf{V}, \\
C6 : & \xi_{k,s,n}(t), \varphi_{k,s,m(v)}(t) \in \{0, 1\}, \forall k \in \mathbf{K}, n \in \mathbf{N}, m \in \mathbf{M}, v \in \mathbf{V}, \\
C7 : & P_{m,s}^{vBS}(t) \leq \rho_s^{vBS}(t) P_B, \forall m \in \mathbf{M}, P_{v,s}^{vUAV}(t) \leq \rho_s^{vUAV}(t) P_V, \forall v \in \mathbf{V}, P_s^{vLEO}(t) \leq \rho_s^{vLEO}(t) P_L, \forall s \in \{1, 2, 3\}, \\
C8 : & \sum_{n \in \mathbf{N}} \sum_{k \in \mathbf{K}_s} \xi_{k,s,n}(t) \varphi_{k,s,m}^{vBS}(t) \leq \eta_s^{vBS}(t) N, \sum_{n \in \mathbf{N}} \sum_{k \in \mathbf{K}_s} \xi_{k,s,n}(t) \varphi_{k,s,v}^{vUAV}(t) \leq \eta_s^{vUAV}(t) N, \\
& \sum_{n \in \mathbf{N}} \sum_{k \in \mathbf{K}_s} \xi_{k,s,n}(t) \varphi_{k,s}^{vLEO}(t) \leq \eta_s^{vLEO}(t) N, \forall m \in \mathbf{M}, v \in \mathbf{V}, s \in \{1, 2, 3\}, \\
C9 : & (x_i^{vUAV}(t) - x_j^{vUAV}(t))^2 + (y_i^{vUAV}(t) - y_j^{vUAV}(t))^2 \geq (d_{\min}^{vUAV})^2, \forall i, j \in \mathbf{V}, i \neq j, \\
C10 : & \rho_1^{vBS(vUAV)(vLEO)}(t) + \rho_2^{vBS(vUAV)(vLEO)}(t) + \rho_3^{vBS(vUAV)(vLEO)}(t) \leq 1, \\
C11 : & \eta_1^{vBS(vUAV)(vLEO)}(t) + \eta_2^{vBS(vUAV)(vLEO)}(t) + \eta_3^{vBS(vUAV)(vLEO)}(t) \leq 1.
\end{aligned} \tag{29}$$

$C1$ exhibits that a subchannel is only allocated to at most one user in the same cell and slice, $C2$ shows that we can only assign at most one subchannel to one user in the same TS, and $C3$ represents that a user is only associated with one network component in the same TS. Still referring to (29), $C4$ represents the value range of the allocation factors $\{\eta_s, \rho_s\}$. Furthermore, $C5$ limits the non-negativity of the transmit power, while $C6$ limits the value range of binary variables $\xi_{k,s,n}(t)$ and $\varphi_{k,s,m(v)}(t)$. To elaborate further, $C7$ illustrates that the actual power consumed by each vLEO satellite, vUAV and vBS could not exceed the total available power on class- s slices, respectively, while $C8$ guarantees that the number of subchannels used in each network component on class- s slices at TS t is no more than the total subchannel numbers. In order to avoid collision, $C9$ limits the distance between two vUAVs to no less than d_{\min}^{vUAV} , where we assume that all UAVs in the network are at the same altitude. Finally, $C10$ and $C11$ illustrate that the power as well as subchannels assigned to different slice types do not overlap with each other in any of the TSs.

However, since we both consider the inter-cell interferences and introduce the binary variables $\{\Xi_s(t), \Phi_s(t)\}$ for our heterogeneous 3D network, (29) represents a non-convex and mixed-integer non-scalar MOOP, which is subject to multiple constraints. It is challenging to solve it by traditional mathematical optimization methods, and it is also ineffective

to directly utilize existing DRL algorithms.

V. PROBLEM TRANSFORMATION AND SOLUTION

In this section, firstly, we decompose (29) into two sub-problems in order to reduce its complexity. Then, based on the MADDPG algorithm [19], we develop a CDMADDPG model and propose the CDMADDPG algorithm to solve (29) as well as to obtain near-Pareto optimal solutions.

A. MOOP Decoupling

The proposed Eq. (29) includes 6 sets of decision variables and 3 optimization objectives, which makes the exploration of the full Pareto front excessively complex. MOOP decoupling can help mitigate this problem [42]. Furthermore, the RAN slices are independent with each other, so the users' resource allocation within a slice is not affected by other slices. Therefore, we can directly decouple (29) into the inter-slice optimization subproblem \mathbf{P}_1 and the intra-slice optimization subproblem \mathbf{P}_2 , as follows:

$$\mathbf{P}_1 : \max_{\left\{ \mathbf{d}_s^{vUAV}, \eta_s, \rho_s \right\}} \left\{ \begin{array}{l} \sum_{t=1}^T R_1^{sum}(t) \\ \sum_{t=1}^T [\beta - D_2^{ave}(t)] \\ \sum_{t=1}^T SINR_3^{ave}(t) \end{array} \right\}, \tag{31}$$

s.t. $C4, C9-C11$,

and

$$\mathbf{P}_2 : \max_{\{\Xi_s, \Phi_s, \mathbf{P}_s\}} \left\{ \begin{array}{l} \sum_{t=1}^T R_1^{sum}(t), s=1 \\ \sum_{t=1}^T [\beta - D_2^{ave}(t)], s=2 \\ \sum_{t=1}^T SINR_3^{ave}(t), s=3 \end{array} \right\}, \quad (32)$$

s.t. $C1-C3, C5-C8$.

Firstly, the subproblem \mathbf{P}_1 allocates the resources $\{\mathbf{d}_s^{vUAV}, \eta_s, \rho_s\}$ among slices in order to pursue Pareto-optimization of multiple performance metrics. Then, according to $\{\mathbf{d}_s^{vUAV}, \eta_s, \rho_s\}$, the subproblem \mathbf{P}_2 allocates the users' resources $\{\Xi_s, \Phi_s, \mathbf{P}_s\}$ within the class- s slices to optimize the respective performance of each slice, facilitating a beneficial performance enhancement. In this way, \mathbf{P}_1 and \mathbf{P}_2 are optimized iteratively for gradually approaching the Pareto-optimization of the original problem \mathbf{P}^* . When the number of iterations is large enough, the optimal solutions to the two subproblems can be regarded as optimal also for the original problem.

B. The CDMADDPG Model Structure

In this section, the CDMADDPG model structure is proposed for solving the above decoupled subproblems, as shown in Fig. 3. It is composed of three parts, namely the environment, the upper centralized unit and three underlying distributed units. Each centralized or distributed unit relies on DRL, including a main body, termed as Agent $i, \forall i \in \{1, d-1, d-2, d-3\}$, which is equivalent to a player in a game. Agent i can interact with the environment for formulating real-time state observations and reward functions, and may rely on different reward functions to facilitate MOO. In the CDMADDPG model, it is assumed that each agent's tasks exhibit Markovian attributes. Explicitly, a Markov decision process (MDP) consists of a four-tuple, namely $\langle \mathbf{o}_i^t, \mathbf{a}_i^t, r_i^t, \mathbf{o}_i^{t+1} \rangle, \forall i \in \{1, d-1, d-2, d-3\}$, the elements of which are the set of state observation, the action space, reward functions and the next state observation, respectively. In the MDP, the agent observes the current state of the environment and selects an action from the action space. By executing this action, the agent receives a reward from the environment and then goes to the next state. However, we cannot collect global action information for directly determining the optimal MDP, given the huge continuous action space of the CDMADDPG model. Hence, the agents have to train deep neural networks (DNNs) and learn the optimal decisions through their continuous interactions with the environment.

Bearing in mind that the dynamic resource sharing among slices based upon the same physical SAGIN is similar to the multi-agent cooperation scenario of the CDMADDPG model, the global MNO in the system may be viewed as Agent 1 of the centralized supervisory unit in the CDMADDPG model to carry out the inter-slice resource optimization for the entire SAGIN. Similarly, the three classes of SAGIN slices in the system may be transformed into Agent $d-s, s \in \{1, 2, 3\}$ of the underlying distributed units in the CDMADDPG model in order to process the resource allocation within a specific type of slices, respectively.

Then, the elements of the system model may also correspond to the MDP's four-tuples of $\langle \mathbf{o}_i^t, \mathbf{a}_i^t, r_i^t, \mathbf{o}_i^{t+1} \rangle$.

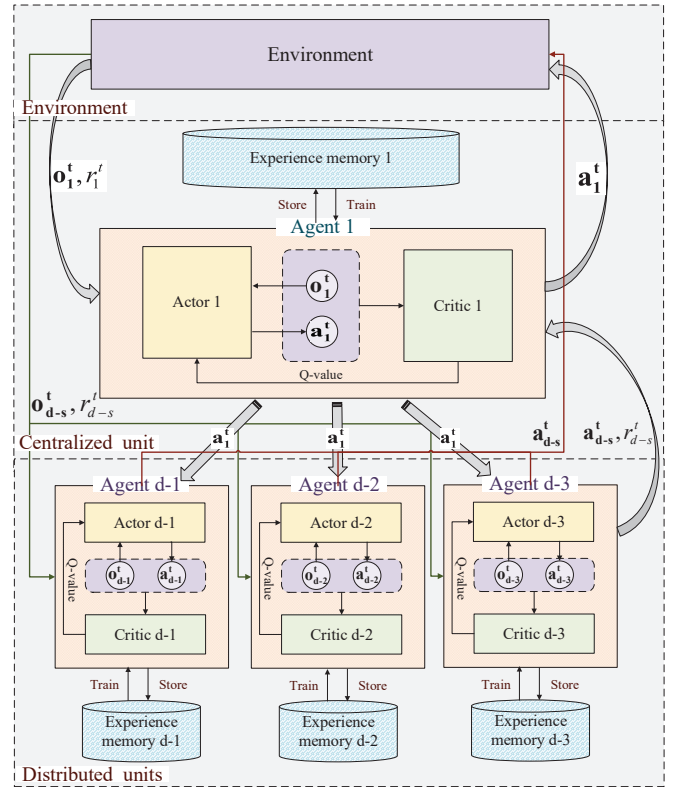


Fig. 3: CDMADDPG model structure.

The current and the next state of Agent 1: As for Agent 1, the state observation is held in a vector of the users' request and position dynamics in the entire system. Hence, at the time step t (equivalent to TS t), the state observation of Agent 1 is defined as:

$$\mathbf{o}_1^t = \{ \mathbf{A}(t), (\mathbf{x}^{UE}(t), \mathbf{y}^{UE}(t)) \}, \quad (33)$$

where $\mathbf{A}(t) = \{A_1(t), A_2(t), A_3(t)\}$ is the set of the data arrival process of the three types of slices in the system. Furthermore, $(\mathbf{x}^{UE}(t), \mathbf{y}^{UE}(t)) = \left\{ \left(x_{k,s}^{UE}(t), y_{k,s}^{UE}(t) \right) \right\}$ represents the coordinate set of all the terrestrial users.

Then, the next state observation of Agent 1 can be formulated as:

$$\mathbf{o}_1^{t+1} = \{ \mathbf{A}(t+1), (\mathbf{x}^{UE}(t+1), \mathbf{y}^{UE}(t+1)) \}, \quad (34)$$

where $\mathbf{A}(t+1)$ and $(\mathbf{x}^{UE}(t+1), \mathbf{y}^{UE}(t+1))$ represent the data arrival process of the three types of slices and all users' coordinates at the next time step.

Action of Agent 1: At the time step t , the MNO has to allocate optimal subchannels and power resources for each class of slices, and determine the optimal vUAVs' positions according to the state observation \mathbf{o}_1^t . Hence, the action set is denoted as:

$$\mathbf{a}_1^t = \{ \eta_s, \rho_s, \mathbf{d}_s^{vUAV} \}, \forall s \in \{1, 2, 3\}. \quad (35)$$

In order to further reduce the action space, we replace " \leq " in $C10$ and $C11$ of (31) with " $=$ ". In this way, Agent 1 only has to train the power and subchannel allocation factors of the first two kinds of slices. Note that the selected actions have to

satisfy the constraints of \mathbf{P}_1 , where those unsatisfied actions are removed.

State and the next state of Agent d -s: As for Agent d -s in the distributed units, the state observation is a vector of users' request on the class- s slices. Hence, at the time step t , the state observation of Agent d -s is defined as:

$$\mathbf{o}_{d-s}^t = \{A_{1,s}(t), A_{2,s}(t), \dots, A_{k,s}(t), \dots\}, \quad (36)$$

where $A_{k,s}(t)$ is the random data arrival of user k on class- s slices at time step t .

Naturally, the next state observation of Agent d -s can be formulated as:

$$\mathbf{o}_{d-s}^{t+1} = \{A_{1,s}(t+1), A_{2,s}(t+1), \dots, A_{k,s}(t+1), \dots\}. \quad (37)$$

Action of Agent d -s: At the time step t , the class- s slices are mapped to the suitable vBSs/vUAVs/vLEO satellite and rely on optimal subchannels and power resources for their users within the slices according to the state observation \mathbf{o}_{d-s}^t . Hence, the action set is denoted as:

$$\mathbf{a}_{d-s}^t = \{\Xi_s, \Phi_s, \mathbf{P}_s\}. \quad (38)$$

For facilitating our mathematical analysis, we relax binary variables of Ξ_s and Φ_s to continuous variables within the value range of $[0,1]$, which also conforms to the condition of continuous actions in our CDMADDPG algorithm. Similarly, the selected actions have to satisfy the constraints of \mathbf{P}_2 . Since sometimes resource contention occurs, a dual resource allocation mechanism is proposed for improving the algorithm's performance and the users' QoS. Specifically, within a time step t , the first resource allocation of Agent d -s is carried out to obtain \mathbf{a}_{d-s}^t . Then, the algorithm checks whether the action set satisfies the constraints, and collects all idle resources. If some of the actions violate the resource constraints and there are resources available in the system, a second resource allocation round shall be carried out immediately to resolve these conflicting actions in order to avoid that the related users fail to get the requested service due to collisions. However, if all available resources in the system are used up, then except for one of the conflicting actions, which retains its original value, all conflicting actions Ξ_s and Φ_s are set to 0 and \mathbf{P}_s are changed to a low value, in order to satisfy the constraints.

Reward of Agent d -s: Since the main performance metrics of the three classes of RAN slices are different, we define different reward functions for Agent d -s in the CDMADDPG model. According to (32), the reward space of Agent d -s can be formulated as follows:

$$r_{d-s}^t = \begin{cases} \overline{R_1^{sum}}(t), & s=1, \\ [1 - \overline{D_2^{ave}}(t)], & s=2, \\ \overline{SINR_3^{ave}}(t), & s=3, \end{cases} \quad (39)$$

where $\overline{R_1^{sum}}(t)$, $\overline{D_2^{ave}}(t)$ and $\overline{SINR_3^{ave}}(t)$ are the normalized objectives by 0-1 normalization. Then, the experience memory of the Agent d -s is summarized as \mathbf{D}_{d-s} , the elements of which are $\langle \mathbf{o}_{d-s}^t, \mathbf{a}_{d-s}^t, r_{d-s}^t, \mathbf{o}_{d-s}^{t+1} \rangle$.

Reward of Agent 1: According to (31), Agent 1 aims for finding Pareto optimal solutions of our MOOP. Hence, we define the reward function for Agent 1 by harnessing the

ensemble strategy of rank voting method [43]. Specifically, we first define the experience memory of Agent 1 as \mathbf{D}_1 to store four-tuples $\langle \mathbf{o}_1^t, \mathbf{a}_1^t, r_1^t, \mathbf{o}_1^{t+1} \rangle$ having $i = 1, d-1, d-2, d-3$ generated by the entire model. Then, we rank the four-tuples in \mathbf{D}_1 in ascending order according to the reward value $r_{d-s}^t, \forall s \in \{1, 2, 3\}$, respectively. Hence, the index χ_s^t is assigned to each four-tuple, which means the four-tuple $\langle \mathbf{o}_1^t, \mathbf{a}_1^t, r_1^t, \mathbf{o}_1^{t+1} \rangle$ is ranked χ_s^t -th position in the ascending sequence by r_{d-s}^t , as follows:

$$\chi_s^t = \text{asc}(\mathbf{D}_1; \langle \mathbf{o}_1^t, \mathbf{a}_1^t, r_1^t, \mathbf{o}_1^{t+1} \rangle), \quad (40)$$

where $i = 1, d-1, d-2, d-3$.

In (40), $\text{asc}(\mathbf{X}; y)$ is the function ordering the \mathbf{X} 's elements in ascending order, and then feeding back the ascending index of the element y to χ_s^t , where y is one of the elements in \mathbf{X} . By adding χ_s^t of three classes of slices for each time step, we can obtain the χ^t as the reward function r_1^t , as follows:

$$r_1^t = \chi^t = \chi_1^t + \chi_2^t + \chi_3^t. \quad (41)$$

In this way, we keep approaching the optimal four-tuple, as follows:

$$\langle \mathbf{o}_1^*, \mathbf{a}_1^*, r_1^*, \mathbf{o}_1'^* \rangle = \arg \max_t \chi^t, \quad (42)$$

where $\langle \mathbf{o}_1^*, \mathbf{a}_1^*, r_1^*, \mathbf{o}_1'^* \rangle$ is not dominated in the space of \mathbf{D}_1 , which conforms to the definition of Pareto optimal solutions (Please refer to Appendix A for the specific proof).

Basic structure of centralized and distributed units: In our CDMADDPG model, both the centralized and the three distributed DRL units rely on the same basic actor-critic (AC) structure [19], where each agent contains an actor and a critic, as seen from Fig. 3. Specifically, the actor selects an action by the current state observation; then the critic aims to evaluate this action and return the Q-value; and finally the actor modifies subsequent action selection policies based on the critic's Q-value. In this way, the learning process of the CDMADDPG model becomes more stable and converges faster than the DRL models operating without the AC structure. Among them, the actor $i, \forall i \in \{1, d-1, d-2, d-3\}$ is designated as the policy function $\mu_{\theta_i}(\mathbf{o}_i^t | \theta^{\mu_i})$ of the DRL unit i , while the critic i could be defined as the action-value function $Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i})$ of the DRL unit i . Thus, we utilize different DNNs for approximating the actor i and the critic i , respectively, and use the classic stochastic gradient method to train both θ^{μ_i} and θ^{Q_i} . Actor i 's input is set to the current state \mathbf{o}_i^t and its output is set to the deterministic action \mathbf{a}_i^t ; while critic i 's inputs are the state \mathbf{o}_i^t and the action \mathbf{a}_i^t generated by the actor, and its output is set to the estimated Q-value. Hence, we could obtain the optimal reward r_i^t by continuously updating both the actor and the critic of Agent i .

Updating the actor: The OF of agent i in the CDMADDPG algorithm can be defined as the expectation of long-term discounted cumulative reward, that is:

$$J_i(\mu) = E_{\mu} [r_i^0 + \gamma r_i^1 + \gamma^2 r_i^2 + \dots + \gamma^{T-1} r_i^{T-1}], \quad (43)$$

where γ is defined as the discount factor. The actor i aims to find the optimal deterministic policy $\mu_{\theta_i}^*$, which corresponds

to the maximization of $J_i(\mu)$, formulated as:

$$\mu_{\theta_i}^* (\mathbf{o}_i^t | \theta^{\mu_i}) = \arg \max_{\mu} J_i(\mu). \quad (44)$$

Hence, the deterministic action \mathbf{a}_i^t of agent i at time step t could be obtained through $\mu_{\theta_i}^*$, as follows:

$$\mathbf{a}_i^t = \mu_{\theta_i}^* (\mathbf{o}_i^t | \theta^{\mu_i}). \quad (45)$$

It has been proved in [44] that the gradient of $J_i(\mu)$ with respect to θ^{μ_i} is equivalent to the expected gradient of $Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i})$ with respect to θ^{μ_i} .

As for the action-value function $Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i})$ of agent i , based on MDP, it can be formally defined using the Bellman's equation, as follows:

$$Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i}) = r_i^t + \gamma \max_{\mathbf{a}} Q_{\theta_i}(\mathbf{o}_i^{t+1}, \mathbf{a}_i^{t+1} | \theta^{Q_i}), \quad (46)$$

where r_i^t is the current reward function, while \mathbf{o}_i^{t+1} and \mathbf{a}_i^{t+1} are the next state and the next action of agent i , respectively.

Subsequently, the update of $Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i})$ is as follows:

$$Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i}) \leftarrow Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i}) + \alpha \delta_t, \quad (47)$$

where

$$\delta_t = r_i^t + \gamma \max_{\mathbf{a}} Q'_{\theta_i}(\mathbf{o}_i^{t+1}, \mathbf{a}_i^{t+1} | \theta^{Q'_i}) - Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i}), \quad (48)$$

and $r_i^t + \gamma \max_{\mathbf{a}} Q'_{\theta_i}(\mathbf{o}_i^{t+1}, \mathbf{a}_i^{t+1} | \theta^{Q'_i})$ is defined as the target, which represents the actual reward of the prediction. Furthermore, δ_t is the error, which is used for estimating the action-value function, while α is the learning rate.

Therefore, the chain rule could be utilized to derive $J_i(\mu)$ and then to get the update method of the actor, as follows:

$$\begin{aligned} \nabla_{\theta^{\mu_i}} J_i &= E_{\mathbf{o}_i^t, \mathbf{a}_i^t \sim \mathbf{D}_i} [\nabla_{\theta^{\mu_i}} \mu_{\theta_i}(\mathbf{o}_i^t | \theta^{\mu_i}) \\ &\quad \cdot \nabla_{\mathbf{a}_i^t} Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i}) |_{\mathbf{a}_i^t = \mu_{\theta_i}(\mathbf{o}_i^t | \theta^{\mu_i})}]. \end{aligned} \quad (49)$$

Referring to the gradient ascent algorithm, we could update the parameter θ^{μ_i} in (49) and optimize the action \mathbf{a}_i^t along the direction of improving the action-value function $Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i})$.

Updating the critic: According to (47) and (48), we update the critic by minimizing the loss function, defined as follows:

$$L_i = E_{\mathbf{o}_i^t, \mathbf{a}_i^t, r_i^t, \mathbf{o}_i^{t+1} \sim \mathbf{D}_i} \left[\left(\text{Target } Q_i - Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i}) \right)^2 \right], \quad (50)$$

and

$$\text{Target } Q_i = r_i^t + \gamma Q'_{\theta_i}(\mathbf{o}_i^{t+1}, \mu'_{\theta_i}(\mathbf{o}_i^{t+1} | \theta^{\mu'_i}) | \theta^{Q'_i}), \quad (51)$$

where $\theta^{\mu'_i}$ and $\theta^{Q'_i}$ in Target Q_i are the parameters from the target actor and the target critic, respectively.

Therefore, the gradient of the critic can be expressed as shown below:

$$\begin{aligned} \nabla_{\theta^{Q_i}} L_i &= E_{\mathbf{o}_i^t, \mathbf{a}_i^t, r_i^t, \mathbf{o}_i^{t+1} \sim \mathbf{D}_i} \left[\left(\text{Target } Q_i - \right. \right. \\ &\quad \left. \left. Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i}) \right) \nabla_{\theta^{Q_i}} Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i}) \right]. \end{aligned} \quad (52)$$

The gradient descent method is used to update the parameter θ^{Q_i} and then to find the optimal value $Q_{\theta_i}^*$.

Online network and target network: In order to guarantee the stability of the learning process, the CDMADDPG algorithm creates two DNNs for each actor/critic, *i.e.* the online and target networks, as follows:

$$\text{actor} \begin{cases} \text{online} : \mu_{\theta_i}(\mathbf{o}_i^t | \theta^{\mu_i}), \text{ update } \theta^{\mu_i}, \\ \text{target} : \mu'_{\theta_i}(\mathbf{o}_i^t | \theta^{\mu'_i}), \text{ update } \theta^{\mu'_i}, \end{cases} \quad (53)$$

and

$$\text{critic} \begin{cases} \text{online} : Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i}), \text{ update } \theta^{Q_i}, \\ \text{target} : Q'_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q'_i}), \text{ update } \theta^{Q'_i}. \end{cases} \quad (54)$$

More explicitly, the CDMADDPG algorithm uses a total of 16 DNNs ((1[online network] + 1[target network]) \times 2[actor&critic] \times 4[agents]). Firstly, the agent i updates θ^{μ_i} and θ^{Q_i} of the online networks through the gradient ascent/descent method after finishing a mini-batch based data training. Then, $\theta^{\mu'_i}$ and $\theta^{Q'_i}$ of the target networks are updated using the following soft update method:

$$\begin{cases} \theta^{Q'_i} \leftarrow \tau \theta^{Q_i} + (1 - \tau) \theta^{Q'_i}, \\ \theta^{\mu'_i} \leftarrow \tau \theta^{\mu_i} + (1 - \tau) \theta^{\mu'_i}, \end{cases} \quad (55)$$

where the soft update parameter τ is 0.001.

DNN configuration: According to the defined state observation and action space of Agent i , we set different number of DNNs' inputs and outputs for respectively approximating Actors as well as Critics. Then, the fixed size of the experience memory is limited to $\mathbf{D}_1 = 10,000$ four-tuples and $\mathbf{D}_{d-s} = 2000$ four-tuples, and we sample $D_1^{mini} = 100$ mini-batches as well as $D_{d-s}^{mini} = 50$ mini-batches for training the centralized unit and distributed units at each time step, respectively. However, no known rule is used for determining the number of neurons and hidden layers, hence it is chosen by trial-and-error [45]. For instance, we set Relu activation function for the hidden layer and then set Sigmoid activation function for the output layer, because all the outputs are positive and between 0 and 1. The specific DNNs' configuration is summarized in Table 3.

C. The CDMADDPG Algorithm

Based on the above CDMADDPG model, the specific algorithmic process is formulated for approaching the Pareto optimal solutions of our MOOP. The CDMADDPG algorithm is executed by the centralized supervisory unit for global learning between different types of slices, and uses the slice-specific distributed units to make local resource decisions within the slices, respectively.

Firstly, Agent 1 in the centralized unit selects the action \mathbf{a}_1^t based on the current policy and exploration for inter-slice resource optimization, and delivers it to Agent $d-s$. In the distributed units, the three dedicated slice-specific agents decide their own actions \mathbf{a}_{d-s}^t for intra-slice resource optimization, and obtain the current reward r_{d-s}^t and the next state \mathbf{o}_{d-s}^{t+1} by their interaction with the environment in a parallel and asynchronous manner. The four-tuples of Agent $d-s$ generated are stored in \mathbf{D}_{d-s} . We randomly sample D_{d-s}^{mini} mini-batches selected from \mathbf{D}_{d-s} to critic $d-s$ and actor $d-s$ for online

Table 3: DNNs' configuration in this paper.

	Number of inputs	Number of outputs	Number of hidden layers	Number of neurons	Activation function	Learning rate	Fixed size of \mathbf{D}_1	D_i^{mini}
Actor 1	$(3+6K_s)$	$(12+2V)$	2	100	Relu+Sigmoid	0.0001	10,000	100
Critic 1	$(15+6K_s+2V)$	1	2	100	Relu+Sigmoid	0.001		
Actor $d-s$	K_s	$3K_s$	2	100	Relu+Sigmoid	0.0001	2000	50
Critic $d-s$	$4K_s$	1	2	100	Relu+Sigmoid	0.001		

Table 4: Specific values of the parameters in (56).

Parameter	N_{ag}	$L_{i,ac}$	$L_{i,cri}$	$n_{1,ac}^{(0)}$	$n_{1,ac}^{(1)}$	$n_{1,ac}^{(2)}$	$n_{1,ac}^{(3)}$	$n_{1,cri}^{(0)}$	$n_{1,cri}^{(1)}$	$n_{1,cri}^{(2)}$
Value	4	2	2	$(3+6K_s)$	100	100	$(12+2V)$	$(15+6K_s+2V)$	100	100
Parameter	$n_{1,cri}^{(3)}$	$n_{d-s,ac}^{(0)}$	$n_{d-s,ac}^{(1)}$	$n_{d-s,ac}^{(2)}$	$n_{d-s,ac}^{(3)}$	$n_{d-s,cri}^{(0)}$	$n_{d-s,cri}^{(1)}$	$n_{d-s,cri}^{(2)}$	$n_{d-s,cri}^{(3)}$	
Value	1	K_s	100	100	$3K_s$	$4K_s$	100	100	1	

and off-policy training. After delivering \mathbf{a}_{d-s}^t and r_{d-s}^t of Agent $d-s$ to Agent 1 and collecting the next state \mathbf{o}_1^{t+1} from the environment, all the four-tuples of the system are generated and stored in \mathbf{D}_1 . Next, we randomly sample D_1^{mini} mini-batches selected from \mathbf{D}_1 and find the non-dominated four-tuple, *i.e.*, calculating r_1^t by (41), for training the critic and actor of Agent 1. In this way, we alternately train the centralized unit and the distributed units until convergence is reached. Finally, it is hard to offer any proof of evidence that the Pareto optimal solutions have indeed been found [33], [46]. Hence we resort to collecting all the non-dominated four-tuples $\langle \mathbf{o}_1^*, \mathbf{a}_1^*, r_i^*, \mathbf{o}_1^* \rangle$ of a few of the final iterations as the near-Pareto optimal solutions of the MOOP formulated. We summarize the specific process in Algorithm 1. By activating Algorithm 1 several times, we can collect multiple near-Pareto optimal solutions so as to present an approximation of the Pareto boundary.

Theorem 1: Similar to the investigations in [47], the order of computational complexity of the learning procedure in our proposed algorithm is given by

$$\mathcal{O} \left(ET \sum_{i=1}^{N_{ag}} \left(\sum_{j=0}^{L_{i,ac}} n_{i,ac}^{(j)} n_{i,ac}^{(j+1)} + \sum_{j=0}^{L_{i,cri}} n_{i,cri}^{(j)} n_{i,cri}^{(j+1)} \right) \right), \quad (56)$$

where E and T refer to the number of episodes and time steps, respectively. Furthermore, N_{ag} represents the number of Agents in our CDMADDPG algorithm, while $n_{i,ac}^{(j)}$ and $n_{i,cri}^{(j)}$ are defined as the numbers of neurons in the j -th layer of actor i and critic i , respectively. Still referring to (56), $L_{i,ac}$ ($L_{i,cri}$) represents the number of hidden layers in actor (critic) i .

In our algorithm, the specific values of these parameters are shown in Table 4, which correspond to the DNNs' configuration in Table 3. We substitute the values of Table 4 into (56) to finally obtain the computational complexity order of the CDMADDPG algorithm as $\mathcal{O}((9K_s+V) \times 400ET)$.

Furthermore, the centralized Reward r_1^t is a monotonically increasing function of the distributed Reward r_{d-s}^t , as shown in (41) and (39), respectively. Therefore, the relationship between them can be characterized as a mixing network, as follows:

$$r_1^t = f_{mix} (r_{d-1}^t, r_{d-2}^t, r_{d-3}^t), \quad (57)$$

where f_{mix} denotes the Mixing network [48], [49], which is a non-negative non-linear monotonic mapping between r_1^t and r_{d-s}^t , $s \in \{1, 2, 3\}$. The convergence of this centralized and distributed algorithm to a locally optimal policy has been substantiated in [49], [50]. In the next section, we will continue our analysis by combining the computational complexity with the simulation of convergence.

VI. SIMULATION RESULTS AND DISCUSSIONS

In this section, we exhibit simulation results to verify the theoretical analysis and compare the performance attained to benchmarks. We adopt a 3D Euclidean coordinate system model and the terrestrial area is given by 3000×3000 (m^2), where K users are randomly distributed. We set $M=2$ and $V=3$, where the coordinates of the terrestrial vBSs are fixed as (1000, 1000, 0)(m) and (2000, 2000, 0)(m), respectively, and the altitudes of the vUAVs and of the vLEO satellite are fixed as $z_v^{vUAV}(t)=100\text{m}$ and $z_v^{vLEO}(t)=200,000\text{m}$, respectively. Due to the high altitude of satellite, the distance may be approximated as $d_{k,s}^{vLEO}(t) \approx \sqrt{(z_v^{vLEO}(t))^2}$. Furthermore, the maximum rate of the vLEO satellite is 100 Mbps. The available bandwidth of different layers is $B=30\text{MHz}$, and it is divided into $N=7$ subchannels, respectively. Unless otherwise specified, we use 20,000 iterations of $E=20$ episodes multiplying $T=1000$ time steps. The other parameters are summarized as follows: $P_B=10\text{dBW}$, $P_V=20\text{dBW}$, $P_L=30\text{dBW}$, $N_0=-130\text{dBm/Hz}$, $\delta=0.1\text{s}$, $\beta=0.2\text{s}$, $\alpha=1.5$, $R=6$, $h_0=-30\text{dB}$, $d_{\min}^{vUAV}=100\text{m}$, $\lambda_2=(10 \times K_2)\text{kbps}$.

In order to highlight the advantages of our proposed scheme, we include four benchmark schemes for comparison.

- Firstly, Benchmark 1 is the conventional single-layer and agent-coupled ‘‘MADDPG’’ algorithm relying on three parallel distributed Agents, where each Agent includes an Actor, which only needs local information for its policy decisions, and a Critic, which has to collect all agents' global policies for action value learning.
- Secondly, in Benchmark 2, we consider the traditional scalar method of solving our MOOP. From the perspective of MNO, we convert the key metrics on three classes of slices into the network-wide utility through the weighted sum method, defined as:

$$U(t) = \omega_1 R_1^{sum}(t) - \omega_2 D_2^{ave}(t) + \omega_3 SINR_3^{ave}(t), \quad (58)$$

Algorithm 1 The Specific Process of our CDMADDPG Algorithm solving (29).

- 1: **Initialization:**
 - 2: Randomly initialize the parameters θ^{Q_i} and θ^{μ_i} of critic $Q_{\theta_i}(\mathbf{o}_i^t, \mathbf{a}_i^t | \theta^{Q_i})$ and of actor $\mu_{\theta_i}(\mathbf{o}_i^t | \theta^{\mu_i})$ for Agent i .
 - 3: Initialize the target network Q'_{θ_i} as well as μ'_{θ_i} , and set their parameters as $\theta^{Q'_i} \leftarrow \theta^{Q_i}$ and $\theta^{\mu'_i} \leftarrow \theta^{\mu_i}$.
 - 4: Initialize the experience memory \mathbf{D}_i of Agent i .
 - 5: **For** episode = 1, ..., E **do**:
 - 6: Initialize a random process \mathbf{N}_i^t for action exploration.
 - 7: Obtain the initial set of states \mathbf{o}_i^0 .
 - 8: **For** time step = 0, ..., $T-1$ **do**:
 - 9: As for Agent 1, select action $\mathbf{a}_1^t = \mu_{\theta_1}(\mathbf{o}_1^t | \theta^{\mu_1}) + \mathbf{N}_1^t$ based on the current policy and exploration for inter-slice resource optimization.
 - 10: Transfer the action \mathbf{a}_1^t of Agent 1 to Agent $d-s$.
 - 11: As for Agent $d-s$, select action $\mathbf{a}_{d-s}^t = \mu_{\theta_{d-s}}(\mathbf{o}_{d-s}^t | \theta^{\mu_{d-s}}) + \mathbf{N}_{d-s}^t$ based on the current policy, on exploration and Agent 1's action \mathbf{a}_1^t for intra-slice resource optimization.
 - 12: Estimate whether \mathbf{a}_{d-s}^t meets the constraints of \mathbf{P}_2 , otherwise activate the dual resource allocation mechanism.
 - 13: Perform actions \mathbf{a}_{d-s}^t and record the rewards r_{d-s}^t as well as the new states \mathbf{o}_{d-s}^{t+1} .
 - 14: In the experience memory \mathbf{D}_{d-s} , store the four-tuples of $\langle \mathbf{o}_{d-s}^t, \mathbf{a}_{d-s}^t, r_{d-s}^t, \mathbf{o}_{d-s}^{t+1} \rangle$.
 - 15: Randomly sample D_{d-s}^{mini} mini-batches of $\langle \mathbf{o}_{d-s}^j, \mathbf{a}_{d-s}^j, r_{d-s}^j, \mathbf{o}_{d-s}^{j+1} \rangle$ selected from \mathbf{D}_{d-s} .
 - 16: Calculate Target Q_{d-s} by (51).
 - 17: Update critic $d-s$ by minimizing loss function of (52).
 - 18: Update actor $d-s$ using the gradient policy algorithm of (49).
 - 19: Update parameters of the target network for Agent $d-s$ by (55).
 - 20: Transfer the action \mathbf{a}_{d-s}^t and the reward r_{d-s}^t of Agent $d-s$ to Agent 1, and record new states \mathbf{o}_1^{t+1} of Agent 1.
 - 21: In the experience memory \mathbf{D}_1 , store the four-tuples of $\langle \mathbf{o}_1^t, \mathbf{a}_1^t, r_1^t, \mathbf{o}_1^{t+1} \rangle$ having $i=1, d-1, d-2, d-3$.
 - 22: Randomly sample D_1^{mini} mini-batches from \mathbf{D}_1 .
 - 23: Calculate r_1^t by (41) and Target Q_1 by (51).
 - 24: Update critic 1 by minimizing loss function of (52).
 - 25: Update actor 1 using the gradient policy algorithm of (49).
 - 26: Update the parameters of the target network for Agent 1 by (55).
 - 27: $\mathbf{o}_i^t \leftarrow \mathbf{o}_i^{t+1}$.
 - 28: **End for**
 - 29: **When** episode=($E-2$) or ($E-1$) or E **do**:
 - 30: Collect all the non-dominated four-tuples according to the value of rewards $r_{d-s}^t, \forall s \in \{1, 2, 3\}$ from \mathbf{D}_1 .
 - 31: **End for**
 - 32: Store the final training parameters $\theta^{\mu_i}, \theta^{Q_i}, \theta^{\mu'_i}$ and $\theta^{Q'_i}$ of DNNs.
-

where ω_1, ω_2 and ω_3 are the unit prices of throughput, delay and SINR gains charged by the MNO, respectively. Clearly, the influence of $R_1^{sum}(t), D_2^{ave}(t)$ and $SINR_3^{ave}(t)$ is directly dependent on the weights of ω_1, ω_2 and ω_3 , respectively, which are usually defined through experience. Therefore, we adjust their weight ratio into 1:1:1, 1:1:4 and 4:4:1, termed as ‘‘utility(1:1:1)’’, ‘‘utility(1:1:4)’’ and ‘‘utility(4:4:1)’’ schemes, respectively, and then we use the deep deterministic policy gradient (DDPG) algorithm to solve them.

- Thirdly, we consider the ‘‘single resource allocation mechanism’’ to be Benchmark 3 of the proposed dual resource allocation mechanism. In Benchmark 3, when some variables in a given action set have resource collision, only one of these variables retains its original value, while the other conflicting variables will be set either to 0 or to a low value to meet the constraints, which means that the corresponding users will not be served.
- Finally, in Benchmark 4, we utilize the ‘‘fixed vUAVs’ position based scheme’’, in which the vUAVs’ positions are fixed as (500, 500), (1500, 1500) and (2500, 2500), respectively; while in our scheme, the vUAVs’ positions are dynamically optimized in \mathbf{P}^* .

We also summarize the differences of the application of these schemes in Table 5.

Fig. 4 shows the convergence of the CDMADDPG algorithm in the training process. Among them, Rewards 1, 2 and 3 are three reward functions defined in (39), which also correspond to three normalized metrics in (29). A total of 20,000 iterations including 20 episodes and 1000 time steps is used. After about 15,000 training iterations, Reward 1, 2 and 3 converge to a relatively steady state, respectively. As for the proposed non-scalar MOOP, the three conflicting optimization objectives sometimes cannot be simultaneously optimized during the iterative training process. Once resource contention occurs, there may be situations where some rewards converge to their minimum in exchange for improving either one or both of the rewards. Additionally, the number of users K_s on each class of slices also affects the reward values. As shown in Fig. 4, Reward 1 increases while Reward 2 and 3 decrease with the increase of K_s from 9 users to 17 users, which corresponds to the increase of both the class-1 slices’ throughput and the class-2 slices’ average delay, as well as to the reduction of the class-3 slices’ average SINR, respectively.

Fig. 5 characterizes the convergence performance of the MADDPG algorithm under the same DNN hyperparameter configuration as our algorithm. We use a total of 40,000 iterations. Observe that all the Rewards 1, 2 and 3 can finally converge to a relatively stable state after 36,000 training iterations. Due to the single-layer and the agent-coupled structure of the MADDPG algorithm, each agent has to collect both local and global information at the same time. This means that upon increasing K_s , the sets of both the global state observations and of the action space will rapidly expand. This imposes a substantial computational load on the training process of the algorithm, hence making the training convergence a challenge. We also arrive at this conclusion from Theorem 1.

Table 5: Comparison between the proposed scheme and four benchmark schemes.

	Proposed scheme	Benchmark 1 [19]	Benchmark 2 [17]	Benchmark 3 [51]	Benchmark 4 [27]
Optimization objective	non-scalar \mathbf{P}^*	non-scalar \mathbf{P}^*	scalar (58)	non-scalar \mathbf{P}^*	non-scalar \mathbf{P}^*
Optimization algorithm	CDMADDPG	MADDPG	DDPG	CDMADDPG	CDMADDPG
Algorithm characteristics	Hierarchical; 4 Agents independent	Single layer; 3 Agents coupled	One Agent	Hierarchical; 4 Agents independent	Hierarchical; 4 Agents independent
Actor's input	$\langle \mathbf{o}_i^t \rangle$	$\langle \mathbf{o}_s^t \rangle$	$\langle \mathbf{o}^t \rangle$	$\langle \mathbf{o}_i^t \rangle$	$\langle \mathbf{o}_i^t \rangle$
Actor's output	$\langle \mathbf{a}_i^t \rangle$	$\langle \mathbf{a}_s^t \rangle$	$\langle \mathbf{a}^t \rangle$	$\langle \mathbf{a}_i^t \rangle$	$\langle \mathbf{a}_i^t \rangle$
Critic's input	$\langle \mathbf{o}_i^t, \mathbf{a}_i^t \rangle$	$\langle \mathbf{o}_s^t, \mathbf{a}_s^t, \mathbf{a}_{S \setminus s}^t \rangle$	$\langle \mathbf{o}^t, \mathbf{a}^t \rangle$	$\langle \mathbf{o}_i^t, \mathbf{a}_i^t \rangle$	$\langle \mathbf{o}_i^t, \mathbf{a}_i^t \rangle$
Critic's output	Q-value of Agent i	Q-value of Agent s	Q-value	Q-value of Agent i	Q-value of Agent i
Resource allocation mechanism	Dual resource allocation	Dual resource allocation	Dual resource allocation	Single resource allocation	Dual resource allocation
vUAVs' position	Dynamic optimization in \mathbf{P}^*	Dynamic optimization in \mathbf{P}^*	Dynamic optimization in \mathbf{P}^*	Dynamic optimization in \mathbf{P}^*	Fixed as (500, 500), (1500, 1500) and (2500, 2500)
Markers in Figures	"CDMADDPG"	"MADDPG"	"Utility(1:1:1)" "Utility(1:1:4)" "Utility(4:4:1)"	"Single resource allocation mechanism"	"Fixed vUAVs position scheme"

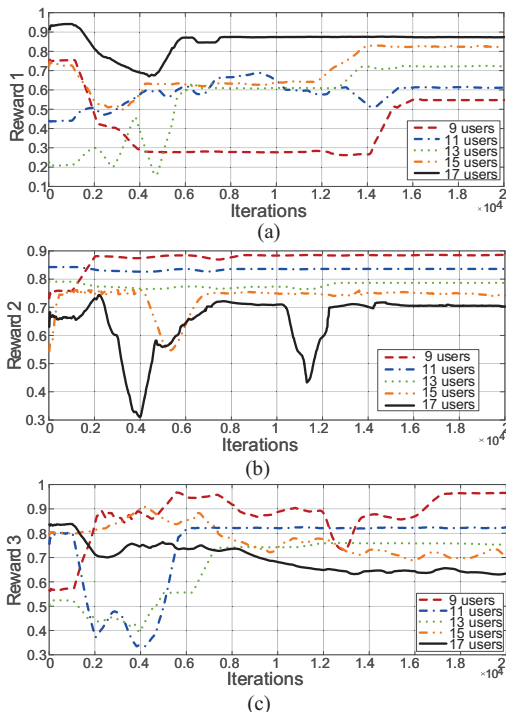


Fig. 4: Convergence of the CDMADDPG algorithm.

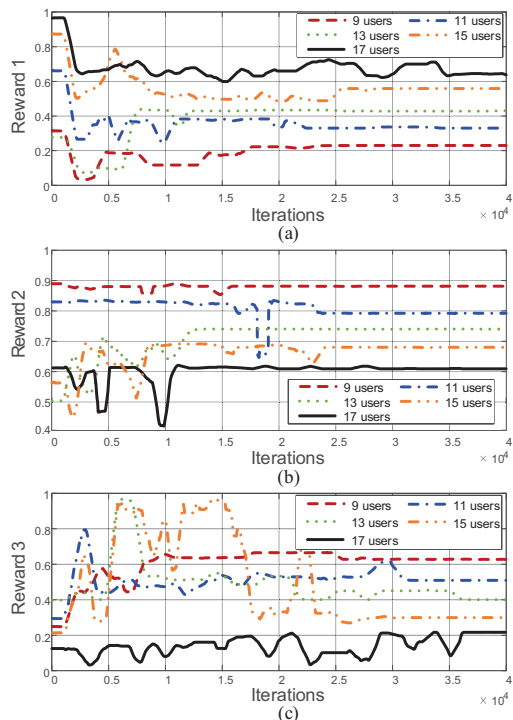


Fig. 5: Convergence of the MADDPG algorithm.

According to (56), the computational complexity order of the CDMADDPG algorithm is given by $\mathcal{O}((9K_s + V) \times 400ET)$ for the current parameter setting, while that of the MADDPG algorithm is $\mathcal{O}((9K_s + 4V) \times 600ET)$. We can see that the complexity difference between the proposed and MADDPG algorithm is expanding with the increase of K_s and iterations ET . Additionally, increasing K_s has the same influence on the performance of class- s slices as the proposed algorithm.

Fig. 6, Fig. 7 and Fig. 8 characterize the throughput, the average delay and average SINR versus K_1 , K_2 and K_3 for the proposed scheme and for the benchmarkers, respectively. In order to facilitate our comparisons, we show time-averaged values in this section, i.e., $\frac{1}{T} \sum_{t=1}^T R_1^{sum}(t)$, $\frac{1}{T} \sum_{t=1}^T Q_2^{ave}(t)$ and $\frac{1}{T} \sum_{t=1}^T SINR_3^{ave}(t)$, and give the mean of 10 experiment trials with the same hyperparameter configuration. Observe in Fig. 6, Fig. 7 and Fig. 8 that with the increase of K_s ,

both the throughput and the average delay of all schemes continue to increase, whilst the average SINR is decreasing. Compared to the MADDPG algorithm, our CDMADDPG algorithm has better performance at 20,000 iterations, thanks to its hierarchical and decoupled framework.

Observe for the three utility schemes of Fig. 6, Fig. 7 and Fig. 8 that the larger the weight assigned, the better the performance becomes. Therefore, the utility(4:4:1) scheme attains a better throughput and has the lowest average delay, but its average SINR is poor. By contrast, the SINR advantage of the utility(1:1:4) scheme is more obvious, but as a penalty, both the throughput and the delay become inferior. Therefore, the influence of the metrics in the utility scheme is highly dependent on their weights, which are subjectively designed in advance. However, it is worth noting that sometimes even if we increase the weight of a specific metric, we may not

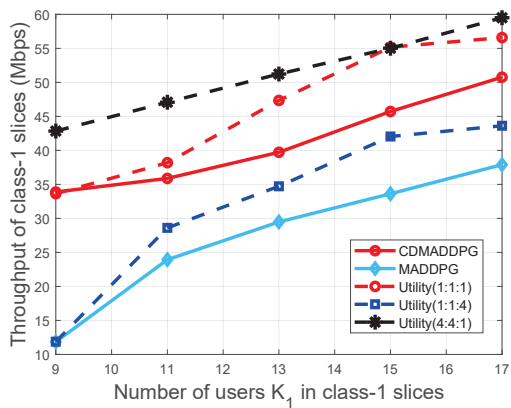


Fig. 6: Throughput versus the number of users K_1 of class-1 slices.

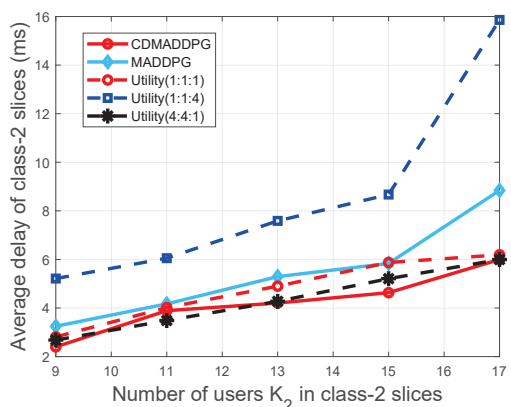


Fig. 7: Average delay versus the number of users K_2 of class-2 slices.

witness any striking advantage. For example, although the SINR of the utility(1:1:4) scheme is somewhat higher than that of the utility(1:1:1) scheme, its throughput and delay is gravely deteriorated. That is to say, even though the traditional scalar method has the benefit of a priori knowledge, it still struggles to satisfy the users' QoS.

By contrast, our scheme is eminently suitable for different types of RAN slices even without designing the utility function and weights in advance. It finds numerous near-Pareto optimal solutions, and these solutions are non-dominated by each other. Taking into account all the above discussions regarding Fig. 6, Fig. 7 and Fig. 8, the CDMADDPG scheme is capable of striking the best tradeoff (pursuing Pareto optimal) among the three classes of slices considered, even if it does not always achieve the best result for all scenarios.

Fig. 9 portrays the average delay versus the data arrival rate λ_2 of class-2 slices for all schemes when $K_2=11$. Upon increasing λ_2 , the average delay of each scheme continues to increase. We can always attain a lower average delay by the CDMADDPG algorithm than that of the MADDPG algorithm.

Fig. 10 compares the two-dimensional coordinates of vUAVs in our proposed scheme to that in the fixed vUAVs' position scheme. In order to facilitate the analysis, we define the area by the coordinates (0, 0) to (1500, 1500)m as an urban area with a higher user density and define an area between (1500, 1,500) and (3000, 3000)m as a rural area having a small

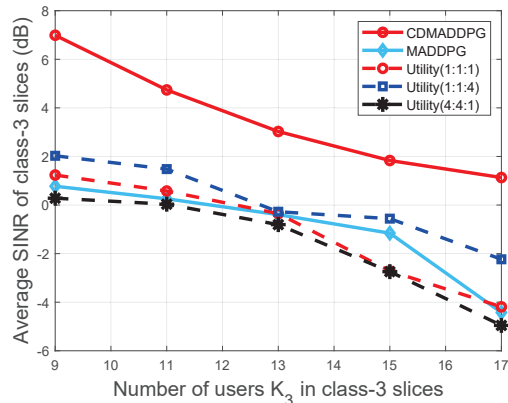


Fig. 8: Average SINR versus the number of users K_3 of class-3 slices.

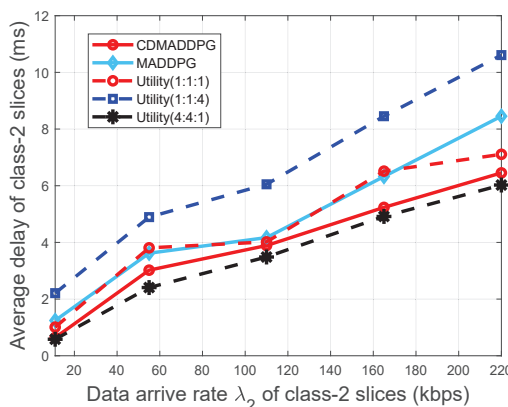


Fig. 9: Average delay versus the data arrival rate λ_2 of class-2 slices.

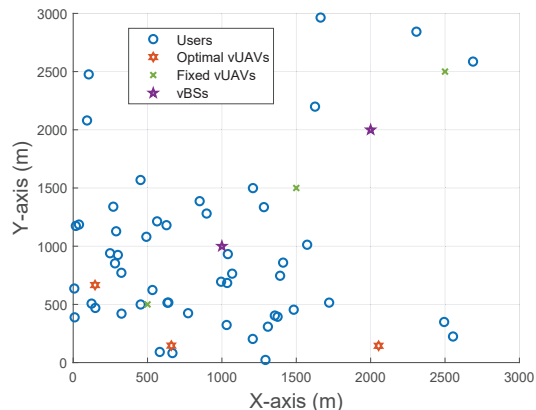


Fig. 10: Coordinate distribution of users, optimal vUAVs, fixed vUAVs and vBSs ($K_1 = K_2 = K_3 = 17$).

number of users, as shown in Fig. 10. Naturally, the optimal vUAVs will actively approach the high-density areas to reduce the transmission distance, but the fixed vUAVs' positions cannot be adjusted in real time. By jointly considering the position distribution of vBSs and optimal vUAVs in Fig. 10, our scheme is seen to provide a reasonable network component deployment.

Fig. 11, Fig. 12 and Fig. 13 portray our performance comparison of the CDMADDPG scheme both with the fixed vUAVs' position based scheme and with the single resource

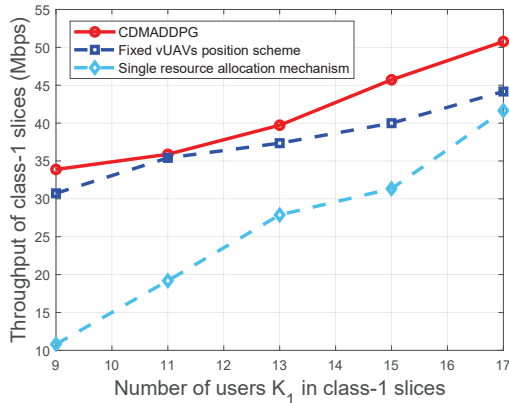


Fig. 11: Throughput comparison with the benchmarks.

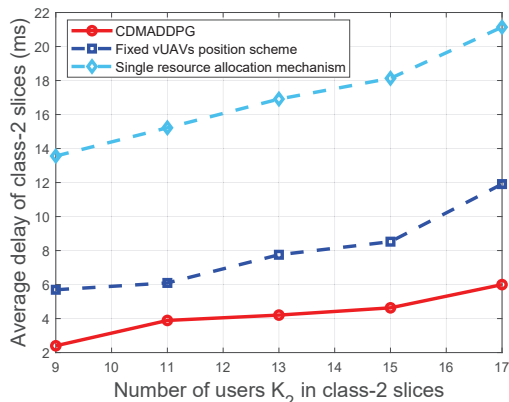
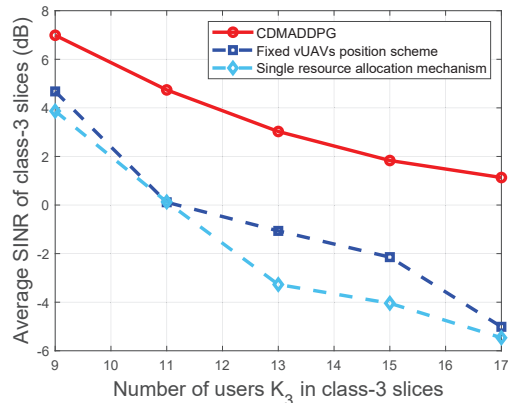


Fig. 12: Average delay comparison with the benchmarks.

allocation mechanism. Firstly, it can be seen that all the performance metrics of the proposed scheme are better than those of the fixed vUAVs' position based scheme. Explicitly, the throughput difference between both schemes is relatively small, while their average delay as well as average SINR are quite different. It transpires that the communication distance has a grave impact on both the average delay and on the average SINR. As expected, since we apply a dual resource allocation mechanism, the proposed scheme exhibits a higher performance advantage than the single resource allocation mechanism. During the DRL training, a large action space leads to a high collision probability, which has a grave impact both on the algorithm's performance and on the users' QoS. In this case, having a dual resource allocation mechanism is indispensable.

In heterogeneous SAGINs, terrestrial, aerial and satellite communications have their own service advantages, which is also reflected in our simulation results. We collect the value of inter-slice sub-channel allocation factors $\eta_s^{vBS}(t)$, $\eta_s^{vUAV}(t)$ and $\eta_s^{vLEO}(t)$ when Algorithm 1 converges, as shown in Table 6. It can be seen that class-1 slices tend to provide access by both vBSs and vUAVs; class-2 slices have a large proportion of vUAVs channel resources; finally, vLEO channel resources are mainly allocated to class-3 slices. This is in line with the service characteristics of these three communication methods. Therefore, the proposed model is eminently suitable for our 3D network having three distinct space-air-ground channel features, and our CDMADDPG algorithm is eminently

Fig. 13: Average SINR comparison with the benchmarks. Table 6: Average value of $\eta_s^{vBS}(t)$, $\eta_s^{vUAV}(t)$ and $\eta_s^{vLEO}(t)$ of ten simulations.

	$\eta_s^{vBS}(t)$	$\eta_s^{vUAV}(t)$	$\eta_s^{vLEO}(t)$
Class-1 slices	0.428	0.333	0.286
Class-2 slices	0.286	0.445	0.143
Class-3 slices	0.286	0.222	0.571

suitable for the customized services of users on the different SAGIN slices.

Fig. 14(a) portrays the three-dimensional scatter diagram of near-Pareto optimal solutions for (29), which is obtained by iteratively activating Algorithm 1 ten times when $K_s = 11$. Let us briefly consider a pair of extreme points at the top left and bottom right corners represented by the approximate throughputs of [18,68]Mbps, delays of [18,2]ms and SINR of [10,1]dB. Furthermore, Fig. 14(b) represents the approximate Pareto boundary derived by using interpolation among the points of Fig. 14(a). For the convenience of observation, we also provide the front view and side view of Fig. 14(b), as shown in Fig. 14(c) and Fig. 14(d) respectively. Since all solutions on the Pareto boundary are non-dominated by each other, the metric's improvement on one slice is accompanied by metrics' degradation on the other one or two slices. Explicitly, the throughput, average delay and SINR cannot be simultaneously improved, indicating their tradeoff. For example, when the average SINR of class-3 slices is higher than 7 dB, the average delay of class-2 slices will also exceed 6 ms, and the throughput of class-1 slices will be reduced below 36Mbps. In practical applications, the MNO may activate one of the near-Pareto optimal solutions by referring to slices' priority, where each solution involves a specific resource allocation. Suppose that the low-delay slices own a higher priority, MNO may activate a near-Pareto optimal solution having the lowest delay, while guaranteeing basic target requirements of the high-throughput slices and the wide-coverage slices.

VII. CONCLUSIONS

In this paper, we simultaneously support high-throughput slices, low-delay slices and wide-coverage slices under the same 3D physical SAGIN. In order to jointly optimize the throughput, average delay and SINR, we have formulated a non-scalar MOOP involving the optimal vUAVs' position

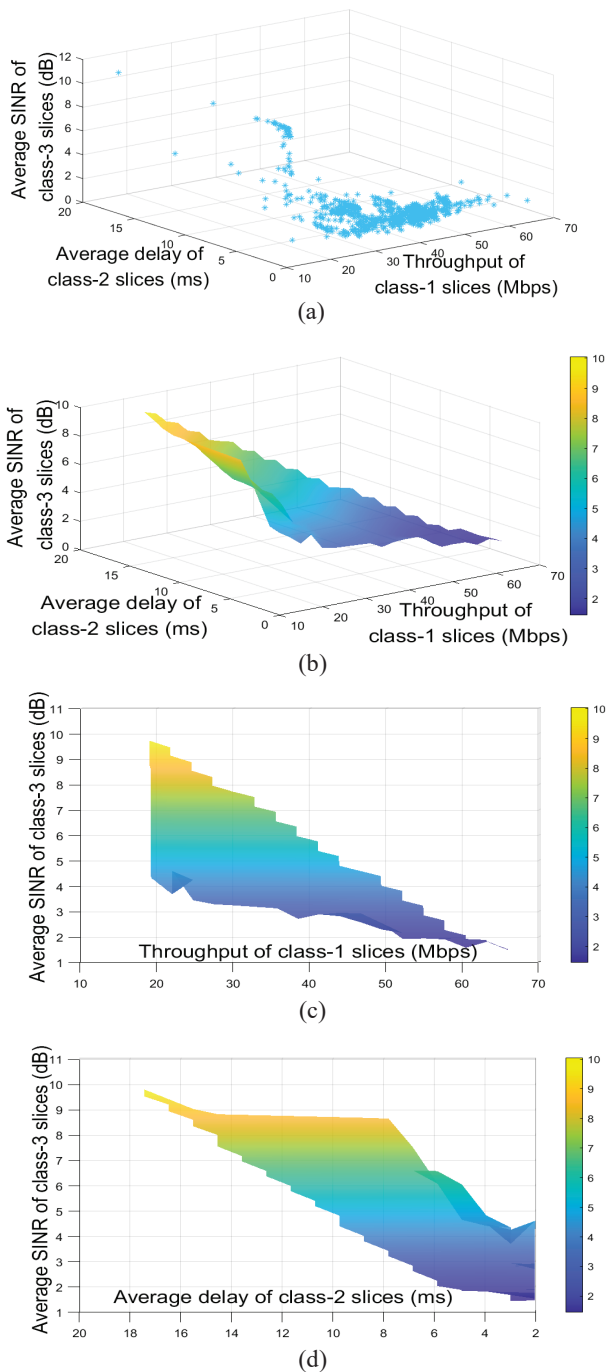


Fig. 14: (a) Near-Pareto optimal scatter diagram; (b) Interpolated near-Pareto boundary; (c) Front view of (b); (d) Side view of (b) ($K_1 = K_2 = K_3 = 11$).

decisions, dynamic network components' deployment and subchannel as well as power allocation. The CDMADDPG algorithm has been proposed for solving the non-scalar MOOP and for approaching the Pareto optimal solutions, where the centralized unit was used for the global resource allocation policies among the three classes of slices, while three distributed slice-specific units were constructed for making local resource decisions within the slices. Moreover, a dual resource allocation mechanism was designed for improving the algorithm's performance and the users' QoS. The proposed

algorithm exhibited better performance than the traditional MADDPG algorithm in multi-user scenarios despite its lower computational complexity. The proposed scheme has also exhibited prominent performance advantages over the single resource allocation mechanism and the fixed vUAV position based scheme. Compared to the traditional scalar method, our scheme has the distinct benefit of operating without designing the utility function and weights in advance, while still finding numerous near-Pareto optimal solutions, characterizing the set of tradeoff amongst the different slices.

APPENDIX A NON-DOMINANCE PROOF OF (42)

Firstly, we assume that the four-tuple found by maximal r_1^t (χ^*) can be dominated by other four-tuples in the space of \mathbf{D}_1 . By *Definition 1* of Pareto optimal solutions in [52], there must be another solution of $\langle \mathbf{o}_1^j, \mathbf{a}_1^j, r_i^j, \mathbf{o}'_1^j \rangle$ associated with $r_{d-s}^j \geq r_{d-s}^*$, $s = 1, 2, 3$. In this way, the index χ_s^j by r_{d-s}^j must be higher than the index χ_s^* by r_{d-s}^* , and then the sum of the three agents' indices χ^j must be higher than the sum of the three agents' indices χ^* .

However, we know that χ^* is the maximum sum of indices. That is, there exists the contradiction between χ^* and χ^j , which indicates the assumption is false. Therefore, the four-tuple $\langle \mathbf{o}_1^*, \mathbf{a}_1^*, r_i^*, \mathbf{o}'_1^* \rangle$ found by maximal r_1^t is not dominated by any other four-tuples in the space of \mathbf{D}_1 .

REFERENCES

- [1] D. C. Nguyen *et al.*, "6G Internet of Things: A Comprehensive Survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359-383, 1 Jan., 2022.
- [2] Q. Wu *et al.*, "A Comprehensive Overview on 5G-and-Beyond Networks With UAVs: From Communications to Sensing and Intelligence," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 2912-2945, Oct. 2021.
- [3] X. You, C.X. Wang, J. Huang *et al.*, "Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts," *Sci. China Inf. Sci.*, vol. 64, pp. 110301, Jan. 2021.
- [4] Z. Jia, M. Sheng, J. Li and Z. Han, "Toward Data Collection and Transmission in 6G Space-Air-Ground Integrated Networks: Cooperative HAP and LEO Satellite Schemes," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 10516-10528, 1 July, 2022.
- [5] J. Liu, X. Du, J. Cui, M. Pan and D. Wei, "Task-Oriented Intelligent Networking Architecture for the Space-Air-Ground-Aqua Integrated Network," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5345-5358, June 2020.
- [6] J. Zhang, *et al.*, "Aeronautical ad-hoc networking for the Internet-above-the-clouds," *Proceedings of the IEEE*, vol. 107, no. 5, pp. 868-911, May 2019.
- [7] Y. Wang, Z. Su, J. Ni, N. Zhang and X. Shen, "Blockchain-Empowered Space-Air-Ground Integrated Networks: Opportunities, Challenges, and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 160-209, Firstquarter 2022.
- [8] F. Lyu *et al.*, "Service-Oriented Dynamic Resource Slicing and Optimization for Space-Air-Ground Integrated Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7469-7483, July 2022.
- [9] H. Cao *et al.*, "Toward Tailored Resource Allocation of Slices in 6G Networks With Softwarization and Virtualization," *IEEE Internet of Things Journal*, vol. 9, no. 9, pp. 6623-6637, 1 May, 2022.
- [10] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini and H. Flinck, "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429-2453, thirdquarter 2018.
- [11] D. Wu, Z. Zhang, S. Wu, J. Yang and R. Wang, "Biologically Inspired Resource Allocation for Network Slices in 5G-Enabled Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9266-9279, Dec. 2019.

- [12] P. Yang, X. Xi, T. Q. S. Quek, J. Chen, X. Cao and D. Wu, "RAN Slicing for Massive IoT and Bursty URLLC Service Multiplexing: Analysis and Optimization," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14258-14275, 15 Sept., 2021.
- [13] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen and L. Hanzo, "A Survey of Multi-Objective Optimization in Wireless Sensor Networks: Metrics, Algorithms, and Open Problems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 550-586, Firstquarter 2017.
- [14] E. Bjornson, E. Jorswieck, M. Debbah, *et al.*, "Multi-Objective Signal Processing Optimization: The Way to Balance Conflicting Metrics in 5G systems," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 14-23, 2014.
- [15] Q. Shi, L. Zhao, Y. Zhang, G. Zheng, F. R. Yu and H. Chen, "Energy-Efficiency Versus Delay Tradeoff in Wireless Networks Virtualization," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 837-841, Jan. 2018.
- [16] I. Afolabi, J. Prados-Garzon, M. Bagaa, T. Taleb and P. Ameigeiras, "Dynamic Resource Provisioning of a Scalable E2E Network Slicing Orchestration System," *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2594-2608, 1 Nov. 2020.
- [17] G. Zhou, L. Zhao, K. Liang, G. Zheng and L. Hanzo, "Utility Analysis of Radio Access Network Slicing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1163-1167, Jan. 2020.
- [18] G. Wang, G. Feng, T. Q. S. Quek, S. Qin, R. Wen and W. Tan, "Reconfiguration in Network Slicing-Optimizing the Profit and Performance," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 591-605, June 2019.
- [19] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mor-datch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, pp. 6379-6390, 2017.
- [20] J. Cui, S. X. Ng, D. Liu, J. Zhang, A. Nallanathan and L. Hanzo, "Multiobjective Optimization for Integrated Ground-Air-Space Networks: Current Research and Future Challenges," *IEEE Vehicular Technology Magazine*, vol. 16, no. 3, pp. 88-98, Sept. 2021.
- [21] J. Wang, C. Jiang, H. Zhang, Y. Ren, K. -C. Chen and L. Hanzo, "Thirty Years of Machine Learning: The Road to Pareto-Optimal Wireless Networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1472-1514, thirdquarter 2020.
- [22] B. Gu, X. Yang, Z. Lin, W. Hu, M. Alazab and R. Kharal, "Multiagent Actor-Critic Network-Based Incentive Mechanism for Mobile Crowdsensing in Industrial Systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 6182-6191, Sept. 2021.
- [23] Y. Zhang, Z. Zhuang, F. Gao, J. Wang and Z. Han, "Multi-Agent Deep Reinforcement Learning for Secure UAV Communications," 2020 IEEE Wireless Communications and Networking Conference (WCNC), 2020, pp. 1-5.
- [24] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, "6G: Opening New Horizons for Integration of Comfort, Security, and Intelligence," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 126-132, Oct. 2020.
- [25] J. Liu, Y. Shi, Z. M. Fadlullah and N. Kato, "Space-Air-Ground Integrated Network: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714-2741, Fourthquarter 2018.
- [26] J. Sun, F. Liu, Y. Zhou, G. Gui, T. Ohtsuki, S. Guo, and F. Adachi, "Surveillance Plane Aided Air-Ground Integrated Vehicular Networks: Architectures, Applications, and Potential," *IEEE Wireless Communications*, vol. 27, no. 6, pp. 122-128, Dec. 2020.
- [27] Y. Wang *et al.*, "Joint Resource Allocation and UAV Trajectory Optimization for Space-Air-Ground Internet of Remote Things Networks," *IEEE Systems Journal*, vol. 15, no. 4, pp. 4745-4755, Dec. 2021.
- [28] X. Cao, B. Yang, C. Yuen and Z. Han, "HAP-Reserved Communications in Space-Air-Ground Integrated Networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 8286-8291, Aug. 2021.
- [29] S. Mao, S. He and J. Wu, "Joint UAV Position Optimization and Resource Scheduling in Space-Air-Ground Integrated Networks With Mixed Cloud-Edge Computing," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3992-4002, Sept. 2021.
- [30] J. Ye, S. Dang, B. Shihada and M. -S. Alouini, "Space-Air-Ground Integrated Networks: Outage Performance Analysis," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 7897-7912, Dec. 2020.
- [31] D. A. V. Veldhuizen and G. B. Lamont, "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art," *Evolutionary Computation*, vol. 8, no. 2, pp. 125-147, June 2000.
- [32] H. D. Chantre and N. L. Saldanha da Fonseca, "The Location Problem for the Provisioning of Protected Slices in NFV-Based MEC Infrastructure," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1505-1514, July 2020.
- [33] J. Cui, H. Yetgin, D. Liu, J. Zhang, S. X. Ng and L. Hanzo, "Twin-Component Near-Pareto Routing Optimization for AANETs in the North-Atlantic Region Relying on Real Flight Statistics," *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 346-364, 2021.
- [34] X. Foukas, G. Patounas, A. Elmokashfi and M. K. Marina, "Network Slicing in 5G: Survey and Challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94-100, May 2017.
- [35] S. Zhang, H. Zhang, B. Di and L. Song, "Cellular UAV-to-X Communications: Design and Optimization for Multi-UAV Networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 2, pp. 1346-1359, Feb. 2019.
- [36] R. d. Silva and S. Rajasinghege, "Optimal Desired Trajectories of UAVs in Private UAV Networks," 2018 International Conference on Advanced Technologies for Communications (ATC), 2018, pp. 310-314.
- [37] Y. Shi, Y. Xia and Y. Gao, "Joint Gateway Selection and Resource Allocation for Cross-Tier Communication in Space-Air-Ground Integrated IoT Networks," *IEEE Access*, vol. 9, pp. 4303-4314, 2021.
- [38] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen and A. Ghryeb, "UAV Trajectory Planning for Data Collection from Time-Constrained IoT Devices," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 34-46, Jan. 2020.
- [39] S. Asmussen, *Applied Probability and Queues*. New York, USA: Springer-Verlag, 2003.
- [40] G. Wang, S. Zhou and Z. Niu, "Radio Resource Allocation for Bidirectional Offloading in Space-Air-Ground Integrated Vehicular Network," *Journal of Communications and Information Networks*, vol. 4, no. 4, pp. 24-31, Dec. 2019.
- [41] Y. Cao, S. -Y. Lien, Y. -C. Liang, K. -C. Chen and X. Shen, "User Access Control in Open Radio Access Networks: A Federated Deep Reinforcement Learning Approach," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 3721-3736, June 2022.
- [42] D. Liu, J. Zhang, J. Cui, S.-X. Ng, R. G. Maunder, and L. Hanzo, "Deep Learning Aided Routing for Space-Air-Ground Integrated Networks Relying on Real Satellite, Flight, and Shipping Data," *IEEE Wireless Communications*, vol. 29, no. 2, pp. 177-184, Apr. 2022.
- [43] T. Brys, *et al.*, "Multi-objectivization and Ensembles of Shapings in Reinforcement Learning," *Neurocomputing*, vol. 263, pp. 48-59, Nov. 2017.
- [44] D. Silver, J. Schrittwieser, K. Simonyan, *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, pp. 354-359, Oct. 2017.
- [45] G. Sun, Z. T. Gebrekidan, G. O. Boateng, D. Ayepah-Mensah and W. Jiang, "Dynamic Reservation and Deep Reinforcement Learning Based Autonomous Resource Slicing for Virtualized Radio Access Networks," *IEEE Access*, vol. 7, pp. 45758-45772, 2019.
- [46] J. Zhang, D. Liu, S. Chen, S. X. Ng, R. G. Maunder and L. Hanzo, "Multiple-Objective Packet Routing Optimization for Aeronautical Ad-Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 1002-1016, Jan. 2023.
- [47] J. Mei, X. Wang, K. Zheng, G. Boudreau, A. B. Sediq and H. Abou-Zeid, "Intelligent Radio Access Network Slicing for Service Provisioning in 6G: A Hierarchical Deep Reinforcement Learning Approach," *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 6063-6078, Sept. 2021.
- [48] Y. Gao, S. Yang, F. Li, S. Trajanovski, P. Zhou, P. Hui and X. Fu, "Video Content Placement at the Network Edge: Centralized and Distributed Algorithms," *IEEE Transactions on Mobile Computing*, DOI 10.1109/TMC.2022.3200401.
- [49] J. Su, S. Adams and P. A. Beling, "Value-Decomposition Multi-Agent Actor-Critics," Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI21), May 2021.
- [50] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," *Advances in neural information processing systems*, 1008-1014, 2000.
- [51] C. Zhou, *et al.*, "Deep Reinforcement Learning for Delay-Oriented IoT Task Scheduling in SAGIN," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 911-925, Feb. 2021.
- [52] G. Zhou, L. Zhao, G. Zheng, Z. Xie, S. Song and K. -C. Chen, "Joint Multi-Objective Optimization for Radio Access Network Slicing Using Multi-Agent Deep Reinforcement Learning," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 9, pp. 11828-11843, Sept. 2023.