Multi-UAV Placement and User Association in Uplink MIMO Ultra-Dense Wireless Networks

Nima Nouri, *Student Member, IEEE*, Fahimeh Fazel, *Student Member, IEEE*, Jamshid Abouei, *Senior Member, IEEE*, and Konstantinos N. Plataniotis, *Fellow, IEEE*

Abstract—This paper investigates an Unmanned Aerial Vehicle (UAV)-enabled network consisting of smart mobile devices and multiple UAVs as aerial base stations in a Multiple-Input Multiple-Output (MIMO) architecture. Mobile devices are partitioned into several clusters and offload their tasks to the UAV servers via the Non-Orthogonal Multiple Access (NOMA) protocol. The main goal of the paper is to jointly maximize the number of served terrestrial users and their scheduling. Moreover, the number of UAV servers and their 3D placement are optimized. To this end, we formulate an optimization problem subject to some Quality of Service (QoS) constraints. The resulting problem is non-convex and intractable to solve. Therefore, we break the problem into two subproblems. We propose an efficient algorithm based on machine learning to solve the first subproblem, i.e., optimizing the number of UAVs and their 3D placements, and the user association. Different from existing literature, our proposed algorithm can achieve low computational complexity and fast convergence. The second subproblem, the user scheduling, is non-convex too. We utilize the ℓ_p-norm concept to find a convex upper bound for the subproblem and optimize the user scheduling by applying the Successive Convex Approximation (SCA) algorithm. The aforementioned process is performed iteratively until the overall algorithm converges and a near-optimal solution is achieved for the optimization problem. Moreover, the computational complexity of the proposed scheme is analyzed. Finally, we evaluate the performance of our proposed algorithm via the simulation results. Regarding fast convergence and low computational complexity of the proposed algorithm, its superior performance is confirmed through numerical results.

Index Terms—3D UAV placement, mobile edge computing, MIMO, machine learning

1 Introduction

With the substantial growth of the Internet of Things (IoT) smart devices for sensing real-time information in 5G networks and beyond 6G, a considerable amount of traffic data is generated in the Ultra-Dense Heterogeneous Networks (UDHNs). Furthermore, many emerging applications are receiving more and more popularity but most of these applications are delay-sensitive and computational intensive, which makes the IoT Mobile Devices (IMDs) very hard to execute because of limited battery and computation resources such as CPU and memory. Recently, Mobile Edge Computing (MEC) has been proposed as a way to enable IMDs to offload their delay-sensitive and computationally intensive tasks to the edge cloud. In this way, IMDs not only prolong the battery lifetime but also enhance their computational capabilities [1]. One of the main drawbacks of conventional terrestrial MEC in a 5G network is the resource constraints that may not be able to meet the requirements of all the IMDs' requests at the same time. Further, MEC with fixed location lacks flexibility and may not be suitable to the cases where the number and the requirement of the user equipment (UE) keep changing. Unmanned Aerial Vehicles (UAVs), due to the features of low cost, high flexibility, and ease of deployment, have recently attracted much attention in wireless communication. In fact, UAVs can be employed in temporary adventures such as when the terrestrial Base Stations (BSs) cannot serve all the IMDs (e.g., due to congestion, or when the ground BSs are broken due to vandalism, bad weather conditions, and transmission problems). Recently, UAV-enabled MEC has been proposed by integrating MEC servers into UAVs to provide computing resources to IMDs. Such new technology assists terrestrial BSs to supply high data rate coverage wherever and whenever there is an extreme requirement and barricade temporary congestion in areas such as stadiums. Particularly, in situations when this demand occurs in a relatively difficult to predict manner. MEC-enabled UAVs can also support IMDs in remote or sparsely populated areas that there is no coverage by terrestrial networks [2]. Machine Learning (ML) techniques have been recognized as current applications of Artificial Intelligence (AI) that pave the way for processing and analyzing such devices' data. In fact, the integration of AI and IoT concepts leads to increasing the connectivity among smart devices and boosting their capabilities, such as self-correcting, self-healing, and decision-making[3]. In addition, big data pertinent to IoT Mobile Devices (IMDs) is transmitted to the cloud network as a data processor through the internet in the traditional Mobile Edge Computing (MEC) architecture to process data in real-time

- *Nima Nouri and Jamshid Abouei are with the Department of Electrical Engineering, Yazd University, Yazd 89158-18411, Iran. E-mail: nimanouri68@gmail.com, abouei@yazd.ac.ir.*
- *Fahimeh Fazel is with the Department of Electrical Engineering, University of Tehran, Tehran 14174, Iran. E-mail: f.fazel@ut.ac.ir.*
- *Konstantinos N. Plataniotis is with the Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S, Canada. E-mail: kostas@ece.utoronto.ca.*

Manuscript received 31 Mar. 2021; revised 24 Aug. 2021; accepted 26 Aug. 2021. Date of publication 1 Sept. 2021; date of current version 3 Feb. 2023.
(Corresponding author: Jamshid Abouei.)
Digital Object Identifier no. 10.1109/TMC.2021.3108960

1536-1233 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

networks. However, conventional MECs cannot provide services for IMDs located at inaccessible positions due to their dependency on terrestrial infrastructure. Recently, new MEC systems equipped with UAVs are utilized for extending the communication coverage to meet the high demands of IoT devices by establishing reconfigurable communications between UAVs and ground IMDs[4]. Moreover, fast deployment, cost and latency reduction, and the ability to navigate long distances for data transmission without additional infrastructure are other advantages of UAV-equipped MEC networks to serve IoT data. In other words, UAVs approach and access ground IMDs, collect big data, and finally transmit information to the data center, providing internet connectivity between devices efficiently. Despite all the benefits above, UAVs operate by using power-limited batteries, giving rise to constraining their execution time. Therefore, increasing the number of devices served by UAVs in limited endurance time is of utmost importance and remains a challenge. Besides, 3D placement of UAVs in MEC networks can be considered another challenge, which should be optimally determined to maximize the network's efficiency.

Orthogonal Multiple Access (OMA) protocols, where the users can periodically access the orthogonal time/frequency resources, are not efficient for 5G ultra-dense wireless networks and beyond from the spectrum efficiency and delay (fairness) points of view, while in NOMA counterpart, multiple users can utilize all of non-orthogonal time/frequency resources concurrently and permanently by yielding a high spectral efficiency. From this point of view, the users' fairness is included in the NOMA case by the simultaneous transmissions of all users in the network over a shared frequency band. Other advantages of NOMA over OMA are summarized as providing massive connectivity, high spectral and energy efficiency, significant achievable data rate, low latency, exceptional user fairness, large throughput, ultrahigh reliability, and upholding the different quality of services (QoS). In addition, we should consider overhead as a significant factor influencing the network's performance. By applying the NOMA technology in the proposed algorithm, user pairing and power allocation are conducted over each subband, and the overhead signaling increases linearly with the number of subbands. In fact, we balance spectral efficiency and user fairness via power allocation policies and sharing the same bandwidth among several users at the expense of signaling overhead costs in NOMA-UAV networks. In addition, the successive interference cancellation (SIC) process imposes extra latency at the receiver. Therefore, considerations on signaling and time overhead, and performance tradeoffs need to be taken into account in the design of NOMA. However, in the uplink-NOMA transmission applied in the proposed scheme, the user separation process is implemented at UAV servers. Therefore, the signaling overhead and time delay are not changed significantly.

1.1 Related Works

AI-Based MEC Networks. MEC is a distributed architecture utilized in IoT networks to deal with their latency-critical and computation-intensive tasks by deploying edge servers adjacent to smart devices. However, the resource management of MECs needs dynamic solutions to meet the fundamental requirements of future networks. The interplay between AI algorithms and MECs is considered a promising approach to address the issue above [5], [6], [7], [8], [9]. The first study related to AI-based edge servers was framed in [5], where the resource management of MECs was modeled as a Markov decision process. Then, the optimal decision for the MEC resource management was obtained by the use of reinforcement learning. In [6], the K-means clustering algorithm, as an unsupervised ML method, was applied in MECs to determine edge servers' deployments. Besides, wireless sensor networks, a component of AI-driven MECs, were investigated in [7] to obtain in-depth information about users and their environments. Given the limited size of IMDs with low transmission powers, employing AI algorithms in IoT devices has created an essential challenge in UDHNs. Therefore, the line of the work was extended in [8] by employing deep learning algorithms, as an improved AI algorithm applied in high-dimensional situations. In [9], the challenges related to applying deep learning in MEC networks were investigated by employing adaptive resource management and considering users' high mobility.

UAV-Enabled IoT Networks. Owing to the ease of deployment and establishing reconfigurable communications, the emerge of UAV, as a novel type of node in 5G and beyond 6G, draws researchers' attention to UAV-equipped MEC networks consisting of IMDs with transmit power constraints[10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21]. The first study pertinent to applying UAVs in IoT networks was framed in [10], where the authors presented a comprehensive survey related to UAVs' applications. They introduced an architecture for the delivery of data offloading from IMDs to a UAV. The authors in [12] investigated the OMA-based UAV applications for collecting data from IMDs. They optimized the UAVs' association and their 3D trajectories, given the position and uplink power of ground IMDs in time-varying IoT networks. UAV 3D placement and resource allocation were investigated in UAV-enabled MEC networks. The authors in [13] employed a polynomial-time algorithm to determine UAVs' deployment so that the number of required UAVs for the coverage of IMDs was minimized. In [14], the downlink coverage probability of UAVs was analyzed based on their height and antenna gain. In other words, the authors determined UAVs' locations so as to maximize the coverage area related to each UAV. The authors in [15] utilized a ML algorithm as an AI-based method to determine UAVs' optimal deployment by predicting network congestion. Moreover, the overall power consumption of IMDs was degraded by applying this method. Reference [22] proposed a long-term proactive optimization algorithm to minimize the outage probability of a UAV-enabled network subject to UAV power budget, speed, and acceleration. The authors in [23] jointly optimized the power control and trajectories of a UAV base station and a UAV jammer to maximize the achievable secrecy rate in the presence of illegal eavesdroppers. Applying the UAV jammer causes a significant improvement in the secrecy rate value compared to not using the UAV jammer. Authors in [24] investigated a UAV-enabled cloud network under partial computation offloading scenario to maximize

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

energy efficiency by minimizing the number of needed drones while minimizing the cost associated with serving the users under some realistic quality of service constraints. In [16], the authors proposed a ML-based algorithm to predict the movements of IMDs, resulting in determining the trajectory of UAVs. In [17], the author aimed to jointly optimize the throughput of ground users, user scheduling, UAV placement, and power control by solving a mixed-integer non-convex optimization problem. This line of work was extended to a general case in [18], where the throughput related to the uplink and downlink transmission links of a UDHN consisting of multiple UAVs was maximized. Unlike previous works, the authors in [19], [20] jointly optimized the transmit power of IMDs, resource allocation, and the height of multiple UAVs under the NOMA framework to maximize the network coverage. In addition, NOMA-based user scheduling in a UAV-enabled network was investigated in [21], and an iterative algorithm was employed to maximize the effective throughput of users. Authors in [25] investigated an optimization problem based on energy efficiency in mmWave UAV-enabled networks. By decomposing the problem into several subproblems, the authors approximate the non-convex UAV placement problem with a convex one. Then, hybrid precoding with user clustering is performed to obtain better multi-antenna gain. Reference [26] proposed an iterative algorithm to optimize UAVs' optimal placement, beam pattern, and charging time by applying unconstrained sequential convex optimization techniques. The authors in [27] extended this line of work by jointly optimizing the above issues along with the UAV trajectory through the branch and bound method. In [28], the authors maximized the sum rate of a UAV-NOMA-enabled network. The problem was divided into the optimization of the user's location and transmit power by employing the successive convex approximation algorithm.

AI-Based Algorithms in UAV-Enabled IoT Networks. To the best of our knowledge, a few studies have been focused on AI-based algorithms in UAV-assisted IoT networks[29], [30], [31]. Reference [29] considered UAVs as mobile edge servers in IoT networks, in which the task offloading of IMDs to UAVs was modeled as a semi-Markov Decision Process (MDP) problem. In addition, the deep learning method was utilized to maximize the IMDs' throughput. In [30], the UAVs' flight resource allocation was modeled as a partial observable MDP problem, where each state was composed of UAVs' battery level and their adopted heights, as well as the queue of IMDs. A deep reinforcement learning algorithm, called Q-learning, was employed to make optimal decisions for each state and solve the online flight resource allocation problem. Since Q-learning algorithms cannot make efficient decisions in IoT networks due to the massive number of actions related to the IMDs, this line of work was extended in [31] by proposing deep Q-learning onboard flight resource allocation. In fact, the proposed scheme combined Q-learning with deep reinforcement learning to address the issue above which considerably minimized the data loss.

To the best of our knowledge, this paper is the first attempt to investigate a new multi-UAV-enabled MEC system, with jointly optimizing the number of UAVs, their 3D placement, the user association, and the user scheduling in an uplink MIMO-NOMA network. This paper addresses this gap by proposing an ML-based algorithm and utilizing the ℓ_p-norm concept. Furthermore, in contrast to many existing UAV placement works, in which the users' density has not been considered at all in clustering, this assumption has also been taken into account, which has made the model more efficient. Thus, the proposed algorithm can be employed with low time and computational complexity for users with different densifications.

1.2 Contributions

This paper considers the integration of NOMA and multi-UAV-enabled MEC system in an urban environment with different users' densities. The UAVs provide MEC services to support overloaded/damaged ground base stations. We aim to maximize the number of served users, while they offload their tasks to the UAV servers. For such a network, the key contributions of this paper are briefly summarized as follows:

- The main goal is to optimize the number of UAVs and their 3D placement based on the traffic requirements of the network subject to energy and delay limitations. We mathematically formulate the optimization problem, which is non-convex and it is intractable to solve. Therefore, it is broken into two sub-problems: 3D UAVs' placement along with the user association, and the user scheduling.
- In the first step, we focus on solving the 3D UAVs' placement in the plane and user association, which is complicated to solve. To tackle the issue, we propose an efficient ML-based algorithm. Unlike existing researches, the proposed algorithm displays a superior performance, fast convergence, and low computational complexity indicated by the complexity order calculation. The optimum number of UAVs and their optimal 3D placement are obtained by applying the algorithm, while the network capacity and coverage constraints are guaranteed.
- In contrast to much-existing literature in which a certain number of UAVs has been considered in their system models, we propose an efficient algorithm to serve the IMDs by optimizing the (minimum) number of UAVs, by effectively eliminating some redundant UAVs, so that the QoS is preserved.
- In the next step, we focus on the user scheduling subproblem. We mathematically formulate the subproblem by employing the l_p-norm concept. Then, we propose an algorithm to solve the second subproblem by leveraging the Successive Convex Approximation (SCA). This procedure is performed iteratively until the optimum number of UAVs is determined.
- Finally, we evaluate the performance of our multi-UAV-enabled MEC offloading system with different users' densities through extensive simulation results. The results validate the efficiency and superiority of the proposed schemes as compared to other benchmark methods.

Notations. Throughout this paper, we utilize boldface lower and upper case letters as the vectors and matrices,

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

respectively. The operator $\langle \, , \, \rangle$ is defined as $\langle \mathbf{X}, \mathbf{Y} \rangle \triangleq \mathrm{Re}\{\mathrm{tr}(\mathbf{X}^H \mathbf{Y})\}$, where $\mathbf{X}$ and $\mathbf{Y}$ are the matrices. In addition, operation $\mathbb{E}[\bullet]$ and $\mathbb{I}(\bullet)$ denote the expectation operator and indicator function, respectively, where $\mathbb{I}(v) = 1$ if $v \geq 0$, otherwise $\mathbb{I}(\bullet)$ returns 0. The vector norm $\parallel \mathbf{z} \parallel_p$ of $\mathbf{z} = [z_1, \ldots, z_n]^T$, $p = 1, 2, ..,$ is defined as $\parallel \mathbf{z} \parallel_p = \left(\sum_{i=1}^{n} \mid z_i \mid^p\right)^{1/p}$.

1.3 Article Outline

The rest of this paper is organized as follows. In Section 2, we describe our proposed model in the uplink transmission by applying the NOMA technology. In Section 3, we formulate our proposed scheme. Due to the non-convexity of the problem, we break the optimization problem into two subproblems and propose two algorithms to solve each subproblem, and the process is iteratively run to obtain a near-optimal solution for the problem. Then, the complexity of the overall algorithm is calculated. In Section 5, the simulation results are shown to verify the performance of our proposed scheme and confirm our analysis. Finally, we summarize our results and findings in Section 6.

2 System Model and Assumptions

In this paper, we consider an UDHN network consisting of M UAVs, indexed by the set $\mathbb{U} = \{u_1, \ldots, u_M\}$, and N IMDs, which are distributed randomly throughout an area with size $\mathcal{A}$ m^2 (see Fig. 1). The UAVs provide MEC services to support overloaded/damaged ground base stations. In this way, unlike the conventional fixed terrestrial MEC servers, where all processes were done in a centralized manner, the processing tasks can be executed in a decentralized manner in the MEC-enabled UAVs. It is assumed that the network area is partitioned into N_A subareas, indexed by $\{\mathcal{A}_1, \ldots, \mathcal{A}_{N_A}\}$, so that $\cup_{i=1}^{N_A} \mathcal{A}_i = \mathcal{A}$. Each subarea has a particular user density function, denoted by $\mathcal{D}_i$, $1 \leq i \leq N_A$. We use index j_m referring to j^{th} user overlaid by UAV $m \in \mathbb{M} = \{1, \ldots, M\}$ with A_m antennas. Thus, the whole IMDs served by M UAVs are denoted as the set $\mathcal{I} \triangleq \{j_m : j = 1, \ldots, N_m, m = 1, \ldots, M\}$, so that $\sum_{m=1}^{M} \sum_{j=1}^{N_m} j_m = N$. Besides, each IMD j_m is equipped with A_{j_m} antennas. Let define $\mathcal{C}_m$ as the set of IMDs covered by m^{th} UAV with cardinality N_m, so that $\mathbb{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_M\}, \mathcal{C}_m \cap \mathcal{C}_n = \emptyset, \ \forall m, n \in \mathbb{M}$ and $\cup_{m \in \mathbb{M}} \mathcal{C}_m = \mathcal{I}$. Based on the constraint $\mathcal{C}_m \cap \mathcal{C}_n = \emptyset$, it is assumed that each user[1] is covered only by one UAV. The 3D Cartesian coordinates of m^{th} UAV and IMD j_m are denoted by $\mathbf{p}_{j_m} = (x_{j_m}, y_{i_m}, 0)$ and $\mathbf{p}_{u_m} = (x_{u_m}, y_{u_m}, h_{u_m})$, respectively, where each IMD's location is assumed to be fixed and known to its corresponding UAV. Therefore, the euclidean distance between IMD j_m and UAV u_m is obtained as $d_{j_m,u_m} \triangleq \sqrt{(x_{u_m} - x_{j_m})^2 + (y_{u_m} - y_{j_m})^2 + h_{u_m}^2}$.

It is assumed that each user j_m has a computational-intensive task $\mathcal{APP}_{j_m} = \{v_{j_m}, \beta_{j_m}\}$ that should be partially offloaded to its corresponding UAV, where v_{j_m} represents the computing intensity related to the IMD j_m in the unit of CPU cycles per bit. In addition, β_{j_m} denotes the input data size in the unit of bits containing the program code and the input parameters. Moreover, IMDs prefer to offload their tasks on UAV-equipped MEC servers compared to executing them on their own devices due to their constrained computation powers and limited sizes. To improve the computation offloading of multiple users, IMDs can access the Power Domain NOMA (PD-NOMA) to offload their tasks on UAV servers. In addition, the Superposition Coding (SC) technique is employed at IMDs, while the SIC technique is utilized at each UAV to separate its desired signal.

2.1 Channel Model

Due to the signal distortion caused by the obstacles, the Air-to-Ground (AG) channel consisting of either Line of Sight (LoS) or Non-LoS (NLoS) signals is considered between IMDs and the corresponding UAV [32], [33], [34]. Let define $P_{j_m}^{LoS}$ and $P_{j_m}^{NLoS}$ as the LoS and the NLoS connection probabilities related to user j_m, respectively. Then, $P_{j_m}^{LoS}$ is obtained as [12], [24]

$$P_{j_m}^{LoS} = \frac{1}{1 + a\exp\left(-b\left(\frac{180}{\pi}\theta_{j_m} - a\right)\right)}, \tag{1}$$

so that $P_{j_m}^{LoS} + P_{j_m}^{NLoS} = 1$. In addition, a and b are constant values pertinent to the type of environment and carrier frequency. Moreover, θ_{j_m} is the elevation angle between the AG link pertinent to user j_m and UAV u_m, and their corresponding horizontal distance, expressed as

$$\theta_{j_m} = \tan^{-1}\left(\frac{h_{u_m}}{r_{j_m,u_m}}\right), \tag{2}$$

where $r_{j_m,u_m} \triangleq \sqrt{\left(x_{u_m} - x_{j_m}\right)^2 + \left(y_{u_m} - y_{j_m}\right)^2}$ is the horizontal distance between user j_m and UAV u_m. Thus, the average path loss expression related to LoS and NLoS connections is defined as

$$\ell_{j_m} = P_{j_m}^{LoS} \ell_{j_m}^{LoS} + P_{j_m}^{NLoS} \ell_{j_m}^{NLoS}, \tag{3}$$

where $\ell_{j_m}^{LoS}$ and $\ell_{j_m}^{NLOS}$ denote the path loss expression of LoS and NLoS connections for the link (j_m, u_m), respectively, expressed as [24], [33]

$$\ell_{j_m}^{LoS} = 10n\log\left(\frac{4\pi f_c d_{j_m,u_m}}{c}\right) + \xi^{LoS} \quad , \forall j_m \tag{4}$$

$$\ell_{j_m}^{NLoS} = 10n\log\left(\frac{4\pi f_c r_{j_m,u_m}}{c}\right) + \xi^{NLoS} \ , \forall j_m, \tag{5}$$

where n denotes the path loss exponent, f_c is the carrier frequency, and c is the light speed. Besides, ξ^{LoS} and ξ^{NLoS} are the mean additional loss related to LoS and NLoS connections, respectively. Using (4) and (5) in Eq. (3), the closed-form expression for the average path loss ℓ_{j_m} is obtained as

$$\ell_{j_m} = 10n\log\left(\frac{4\pi f_c d_{j_m,u_m}}{c}\right) + P_{j_m}^{LoS}\xi^{LoS} + P_{j_m}^{NLoS}\xi^{NLoS}.$$

Accordingly, the channel coefficients matrix between UAV u_m and user j_m can be represented as [35]

$$\mathbf{H}_{j_m,u_m} = \frac{\tilde{\mathbf{H}}_{j_m,u_m}}{\sqrt{\ell_{j_m}}}, \tag{6}$$

[1] Throughout of the paper, we occasionally use the term "user" instead of IMD.

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

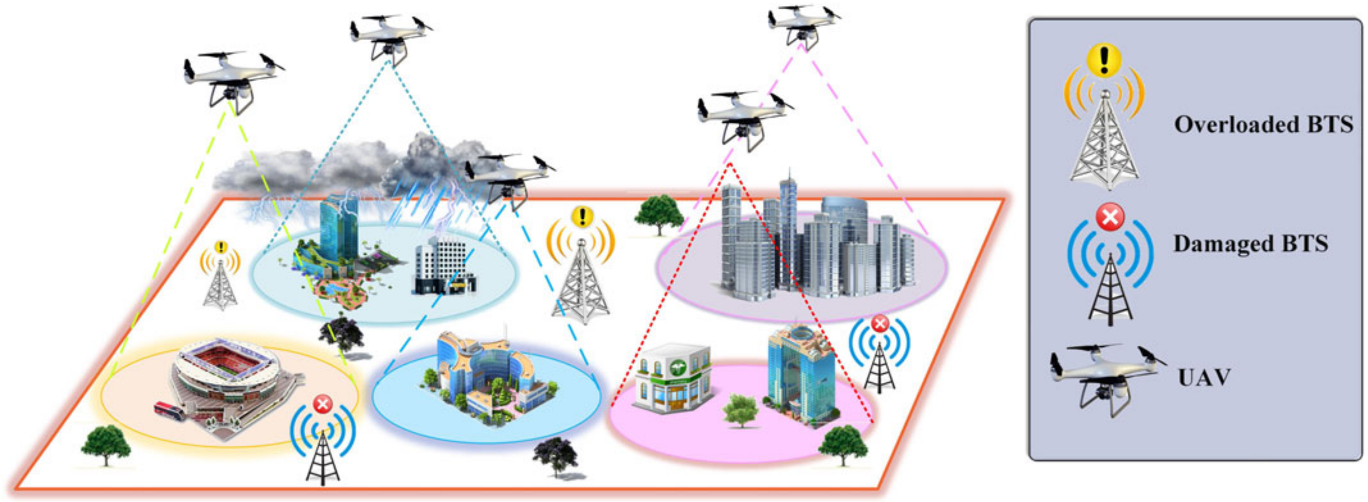


Fig. 1. Graphical illustration of UAV servers employed for serving the areas under the coverage of the overloaded or damaged terrestrial BSs. Solid-black ellipses show the coverage regions served by UAV servers.

where $\tilde{\mathbf{H}}_{j_m,u_m} \in \mathbb{C}^{A_m \times A_{j_m}}$ is modeled as a Rician fading channel consisting of two components: A term of LoS assumed to be constant and deterministic, and a term of Rayleigh fading representing the multipath reflections, expressed as [35]

$$\tilde{\mathbf{H}}_{j_m,u_m} = \sqrt{\frac{K}{K+1}} e^{i\varphi} \tilde{\mathbf{H}}^{LoS}_{j_m,u_m} + \sqrt{\frac{1}{K+1}} e^{i\varphi} \tilde{\mathbf{H}}^{NLoS}_{j_m,u_m}, \tag{7}$$

where K and φ represent the Rician factor as well as the phase shift of the signal between the transmitter and the receiver antennas, respectively. Moreover, $\tilde{\mathbf{H}}^{LoS}_{i_m}$ and $\tilde{\mathbf{H}}^{NLoS}_{i_m}$ refer to the LoS and NLoS components of the channel fading, respectively. The Rician factor K is obtained such that the following equality is satisfied:

$$P^{LoS}_{j_m} = \frac{K}{K+1}. \tag{8}$$

2.2 Signal Model

Employing NOMA technology, all IMDs belonging to m^{th} cluster utilize a superposition coding to transmit the associated tasks to UAV m over the same frequency band. Therefore, the transmitted signal by IMD j_m can be expressed as $\mathbf{x}_{j_m} = \mathbf{\Phi}_{j_m} \mathbf{s}_{j_m}$, $\forall j_m \in \mathcal{I}$, where $\mathbf{\Phi}_{j_m} \in \mathbb{C}^{A_{j_m} \times A_{j_m}}$ is the linear precoding matrix applied at the user j_m. Furthermore, $\mathbf{s}_{j_m} \in \mathbb{C}^{A_{j_m} \times 1}$ is the signal message satisfying $\mathbb{E}\left[\mathbf{s}^H_{j_m} \mathbf{s}_{j_m}\right] = 1$. In addition, the received signal at m^{th} UAV is obtained as

$$\hat{\mathbf{r}}_{u_m} = \sum_{j=1}^{N_m} \mathbf{H}_{j_m,u_m} \mathbf{x}_{j_m} + \hat{\mathbf{z}}_m = \sum_{j=1}^{N_m} \frac{\tilde{\mathbf{H}}_{j_m,u_m}}{\sqrt{\ell_{j_m}}} \mathbf{\Phi}_{j_m} \mathbf{s}_{j_m} + \hat{\mathbf{z}}_m, \tag{9}$$

where $\hat{\mathbf{z}}_m \in \mathbb{C}^{A_m \times 1}$ indicates a zero mean Additive White Gaussian Noise (AWGN) vector at m^{th} UAV. Let $w_m > 0$ denotes the fraction of uplink channel bandwidth assigned to user j_m, so that $\sum_{m \in \mathbb{M}} w_m \leq \mathcal{W}$, where $\mathcal{W}$ is the total available bandwidth. Prior to decoding, the received signal at m^{th} UAV is processed by employing a unitary detection matrix denoted by $\mathbf{V}_m \in \mathbb{C}^{A_m \times A_m}$. Then the received signal at m^{th} UAV after applying $\mathbf{V}_m$ can be rewritten as follows:

$$\begin{aligned} \mathbf{r}_{u_m} &= \mathbf{V}_m \hat{\mathbf{r}}_{u_m} = \mathbf{V}_m \sum_{j=1}^{N_m} \mathbf{H}_{j_m,u_m} \mathbf{x}_{j_m} + \mathbf{V}_m \hat{\mathbf{z}}_m \\ &= \sum_{j=1}^{N_m} \mathbf{V}_m \mathbf{H}_{j_m,u_m} \mathbf{\Phi}_{j_m} \mathbf{s}_{j_m} + \mathbf{z}_m, \end{aligned} \tag{10}$$

where $\mathbf{z}_m \triangleq \mathbf{V}_m \hat{\mathbf{z}}_m \sim \mathcal{CN}(\mathbf{0}, N_0 w_m \mathbf{I}_{A_m})$ and N_0 represents the power spectral density of the noise.

2.3 Uplink Transmission Model

In the uplink UAV-enabled NOMA network, the users' signals are transmitted using superposition coding at the IMDs. In fact, the higher the channel gain is between the IMD and the UAV server, the lower proportion of IMD's power is dedicated to the same UAV. Then, the superposition of users' signals is transmitted to the UAV server, and the SIC technique is employed at the UAV such that the users with better channel gains are decoded first at the UAV in a successive procedure. Accordingly, "user pairing schemes" and "power control techniques," and generally, the "information exchange" are implemented at the UAV for the existing SIC-based uplink NOMA schemes. Hence, UAV needs to send the information about user pairing, channel state information of users' interferes, and transmit power to the NOMA users, which results in an additional signaling overhead at the UAV servers. To address this issue in our proposed scheme, the IMDs can either process their information locally or offload their tasks on the UAV servers by applying the NOMA technology, which is a tradeoff between users' energy consumption and additional signal overhead on the UAV servers.

In the uplink transmission, we aim to optimize $\mathbf{Q} \triangleq \left[\mathbf{Q}_{j_m}\right]_{j_m \in \mathcal{I}}$ as the users' transmit covariance matrices, in

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

which $\mathbf{Q}_{j_m} = \mathbb{E}\left[\mathbf{x}_{j_m}\mathbf{x}_{j_m}^H\right]$, subject to the power budget limitation, expressed as

$$\mathcal{Q}_{j_m} \triangleq \left\{\mathbf{Q}_{j_m} \in \mathbb{C}^{A_{j_m}\times A_{j_m}} : \mathbf{Q}_{j_m} \succeq 0,\ \operatorname{tr}\left(\mathbf{Q}_{j_m}\right) \leqslant p_{j_m}^{max}\right\},$$

where $p_{j_m}^{max}$ denotes the maximum available transmit power budget per symbol related to user j_m. Without loss of generality, we assume that the IMDs belonging to m^{th} cluster are indexed in descending order such that the following inequality is satisfied

$$\left\langle \bar{\mathbf{H}}_{1_m,u_m}, \bar{\mathbf{H}}_{1_m,u_m}\right\rangle \geq \cdots \geq \left\langle \bar{\mathbf{H}}_{(N_m)_m,u_m}, \bar{\mathbf{H}}_{(N_m)_m,u_m}\right\rangle,\ \forall m \in \mathbb{M}, \tag{11}$$

where $\bar{\mathbf{H}}_{j_m,u_m} = \mathbf{V}_m\mathbf{H}_{j_m,u_m},\ 1 \leq j \leq N_m$, and the links $(1_m, u_m)$ and $((N_m)_m, u_m)$ have the weakest and the strongest channel conditions, respectively. By employing the NOMA technology, the SIC is performed at the receiver side to mitigate the interference effect imposed by other IMDs transmitting their data on the same resource [36], [37]. Specifically, user j_m transmits data towards UAV m. Then, UAV m decodes and removes the message of user $i_m,\ \forall i_m \in \mathcal{I},\ i > j$, from the received signal. With the above decoding scheme, the maximum achievable transmission rate pertinent to the link (j_m, u_m) is obtained as

$$\begin{aligned} &r_{j_m}(\mathbf{Q}, w_m) \\ &= w_m\log_2 \det\left(\mathbf{I}_{A_{j_m}} + \lambda\bar{\mathbf{H}}_{j_m,u_m}^H\mathbf{R}_m{}^{-1}(\mathbf{Q}_{-j_m}, w_m)\bar{\mathbf{H}}_{j_m,u_m}\mathbf{Q}_{j_m}\right), \end{aligned} \tag{12}$$

where $\mathbf{R}_m(\mathbf{Q}_{-j_m}, w_m) \triangleq \mathbf{I} + \lambda\sum_{j'=1}^{j-1}\bar{\mathbf{H}}_{j'_m,u_m}\mathbf{Q}_{j'_m}\bar{\mathbf{H}}_{j'_m,u_m}^H$ denotes the normalized covariance matrix of the noise plus the inter-cell interference at m^{th} UAV, in which $\lambda \triangleq \frac{1}{w_m N_0}$ and $\mathbf{Q}_{-j_m} \triangleq \left(\mathbf{Q}_{j'_m}\right)_{j'=1}^{j-1}$ (see *Appendix A* of supplementary material for the proof, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2021.3108960). Moreover, $\bar{\mathbf{H}}_{j'_m,u_m}$ refers to the cross-channel matrix between the interferer user j'_m belonging to m^{th} cluster and the corresponding UAV multiplied by the detection vector $\mathbf{V}_m$. Besides, $t_{j_m}^{ul}(\mathbf{Q}, w_m)$ is defined as the time required for IMD j_m to send the input bits to UAV m in the uplink transmission, obtained as

$$t_{j_m}^{ul}(\mathbf{Q}, w_m) = \frac{\beta_{j_m}}{r_{j_m}(\mathbf{Q}, w_m)}. \tag{13}$$

Then, the total energy consumed by user j_m related to offloading is computed as

$$E_{j_m}^{ul}(\mathbf{Q}, w_m) = \operatorname{tr}\left(\mathbf{Q}_{j_m}\right)t_{j_m}^{ul}(\mathbf{Q}, w_m) = \frac{\beta_{j_m}\operatorname{tr}\left(\mathbf{Q}_{j_m}\right)}{r_{j_m}(\mathbf{Q}, w_m)}. \tag{14}$$

2.4 Local Execution Model

In the proposed network model, the IMD can either process its applications in its own device or offload them to the UAV servers. If the user j_m decides to locally process its application in its device, its energy consumption can be calculated as

$$E_{j_m}^{loc} = \kappa v_{j_m} f_{j_m}^2, \tag{15}$$

where f_{j_m} and $\kappa = 10^{-26}$ represent the device chip clock frequency and a coefficient depending on the chip architecture, respectively [38]. Specifically, the energy per operation for CMOS circuits is proportional to the square of the supply voltage of the chip. If the supply voltage is low, the clock frequency of the chip is a linear function of the voltage supply. Thus, the clock frequency can be calculated as [39]

$$f_{j_m} = \frac{v_{j_m}}{t_{j_m}}. \tag{16}$$

Substituting (16) in (15), the energy consumption can be rewritten as

$$E_{j_m}^{loc} = k\frac{v_{j_m}^3}{t_{j_m}^2}. \tag{17}$$

3 Problem Formulation

As mentioned before, users can either offload their tasks to the UAV servers or process them on their own devices. However, if the energy consumed for local processing is more than the energy required for offloading the task to the UAV servers, i.e., $E_{j_m}^{loc} \geqslant E_{j_m}^{ul}$, the IMD j_m prefers to offload its task. For simplicity, we utilize two auxiliary slack variables entitled $\left(\nu_{j_m}^t, \nu_{j_m}^e\right)$ for each IMD j_m, which are employed to measure the violations of the delay and energy constraints, respectively. Designing such a system ultimately leads to maximizing the number of IMDs offloading their tasks, while they satisfy latency limitations and guarantee energy savings of local execution. In other words, we aim to maximize the number of IMDs who do not violate these two mentioned limitations, i.e., $\nu_{j_m}^t = 0$ and $\nu_{j_m}^e = 0$ during offloading their tasks. To this end, we minimize ℓ_0-norm, written as

$$\left\|\boldsymbol{\nu}^t\right\|_0 + \left\|\boldsymbol{\nu}^e\right\|_0 = \sum_{j_m\in\boldsymbol{\chi}}\mathbb{I}\left(\nu_{j_m}^t > 0\right) + \sum_{j_m\in\boldsymbol{\chi}}\mathbb{I}\left(\nu_{j_m}^e > 0\right), \tag{18}$$

where $\boldsymbol{\nu}^t \triangleq \left[\nu_{j_m}^t\right]_{j_m\in\boldsymbol{\chi}}$, $\boldsymbol{\nu}^e \triangleq \left[\nu_{j_m}^e\right]_{j_m\in\boldsymbol{\chi}}$, and $\boldsymbol{\chi}$ is the set of all users violating latency and energy constraints. Furthermore, $\mathbb{I}(\bullet)$ represents the indicator function such that if $\nu_{j_m}^t, \nu_{j_m}^e > 0$, then $\mathbb{I}(\bullet)$ returns 1, otherwise, it returns 0. The number of users violating delay and energy constraints can be counted by summing the ℓ_0-norm of the slack vectors $\boldsymbol{\nu}^t$ and $\boldsymbol{\nu}^e$ in (18). Besides, $\rho_{m,s}$ denotes a parameter related to the capacity constraint, expressed as follows [40]:

$$\rho_{m,s} = \frac{\sigma_{m,s}}{\mathcal{A}_m},\ 1 \leq m \leq M,\ 1 \leq s \leq N_A, \tag{19}$$

where $\sigma_{m,s}$ denotes the mutual area between UAV m and subarea s, and $\mathcal{A}_m$ is the total area covered by UAV m. Accordingly, the value of $\rho_{m,s}$ is between zero and one.

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

More precisely, if the mutual area $\sigma_{m,s}$ of UAV m is totally included in the subarea $\mathcal{A}_m$, then, $\rho_{m,s} = 1$. However, if the mutual area $\sigma_{m,s}$ is partially included in the subarea $\mathcal{A}_m$, $\rho_{m,s}$ will be less than 1. In addition, $\rho_{m,s} = 0$ refers to the case that the mutual area $\rho_{m,s}$ and $\mathcal{A}_m$ are disjoint. Therefore, the following inequality must be established in order to satisfy the capacity constraint [40]:

$$\sum_{m=1}^{\hat{M}} N^{UAV} \rho_{m,s} \geqslant \mathcal{D}_s \mathcal{A}_s, \ s = 1, \ldots, N_A, \tag{20}$$

where $\hat{M}$ and N^{UAV} represent the initial estimation of the number of UAVs employed in the proposed network, as will be discussed in Section 4.1.1, and the maximum number of IMDs served by a UAV, respectively.

Taking the above considerations into account, the main goal of this paper is to serve the most number of users by employing the optimum minimum number of UAVs. Accordingly, the 3D placement optimization problem and user scheduling can be expressed as

$$\mathscr{P}1) \ \min_{\mathcal{S}} \ \sum_{m=1}^{\hat{M}} \varphi_m + \left\|\boldsymbol{\nu}^t\right\|_0 + \left\|\boldsymbol{\nu}^e\right\|_0$$

$$\textbf{s.t.}$$

$$\textbf{C1}. \ \frac{\beta_{j_m}}{r_{j_m}(\mathbf{Q}, \ w_m)} - T_{j_m} \leqslant \nu^t_{j_m}, \ \forall j_m \in \boldsymbol{\chi},$$

$$\textbf{C2}. \ E^{ul}_{j_m}(\mathbf{Q}, w_m) - E^{loc}_{j_m} \leqslant \nu^e_{j_m}, \ \forall j_m \in \boldsymbol{\chi},$$

$$\textbf{C3}. \ r_{j_m}(\mathbf{Q}, w_m) \geqslant r^{min}_{j_m}, \ \forall j_m \in \boldsymbol{\chi},$$

$$\textbf{C4}. \ \nu^t_{j_m} \geqslant 0, \ \nu^e_{j_m} \geqslant 0, \ \forall j_m \in \boldsymbol{\chi},$$

$$\textbf{C5}. \ w_m \geqslant 0, \ \forall m \in \mathbb{M},$$

$$\textbf{C6}. \sum_{m=1}^{\hat{M}} w_m \leqslant \mathcal{W},$$

$$\textbf{C7}. \ \mathbf{Q}_{j_m} \in \mathcal{Q}_{j_m}, \ \forall j_m \in \boldsymbol{\chi},$$

$$\textbf{C8}. \sum_{m=1}^{\hat{M}} N^{UAV} \rho_{m,s} \geqslant \mathcal{D}_s \mathcal{A}_s, \ s = 1, \ldots, N_A,$$

$$\textbf{C9}. \sum_{j_m \in \mathcal{I}'} \eta_{j_m} \geqslant \ \xi N^{\upsilon},$$

where $\mathcal{S}$ denotes the set of all optimization variables, i.e., $\mathcal{S} \overset{\Delta}{=} \left(\mathbf{Q}, \boldsymbol{\nu}^t, \boldsymbol{\nu}^e, \boldsymbol{\varphi}, \mathbf{w}, \mathbf{x}_u, \mathbf{y}_u, \mathbf{h}_u\right)$, so that $\mathbf{Q} \triangleq \left[\mathbf{Q}_{j_m}\right]_{j_m \in \mathcal{I}'}$, $\boldsymbol{\nu}^t \triangleq \left[\nu^t_{j_m}\right]_{j_m \in \boldsymbol{\chi}}$, $\boldsymbol{\nu}^e \triangleq \left[\nu^e_{j_m}\right]_{j_m \in \boldsymbol{\chi}}$, $\mathbf{w} \overset{\Delta}{=} [w_m]_{m \in \mathbb{M}}$, $\mathbf{x}_u \overset{\Delta}{=} [x_{u_m}]_{m \in \mathbb{M}}$, $\mathbf{y}_u \overset{\Delta}{=} [y_{u_m}]_{m \in \mathbb{M}}$, and $\mathbf{h}_u \triangleq [h_{u_m}]_{m \in \mathbb{M}}$. In addition, $\boldsymbol{\varphi} \overset{\Delta}{=} [\varphi_m]_{m \in \mathbb{M}}$, where φ_m is the indicator function expressing the status of UAV m. In fact, $\varphi_m = 1$ if UAV m is deployed, while, $\varphi_m = 0$ if UAV m is redundant. For the optimization problem $\mathscr{P}1$, constraints **C1** and **C2** represent the delay and energy restrictions related to the data offloading which should be less than or equal to the maximum tolerable delay and energy budget, respectively. Constraints **C5** and **C6** indicate the restrictions related to the network bandwidth. In addition, constraints **C7** and **C8** guarantee that the power budget limitation on the radio interface and the capacity constraint are satisfied, respectively. Moreover, η_{j_m} in constraint **C9** is another indicator function. In fact, if IMD j_m is served by at least one UAV, $\eta_{j_m} = 1$; otherwise, $\eta_{j_m} = 0$. Accordingly, constraint **C9** refers to the coverage limitations and guarantees that at least ξ percent of all users belonging to the set $\mathcal{I}'$ are covered by UAVs, where N^{υ} is the cardinality of the set $\mathcal{I}'$, as will be discussed in Algorithm 1. Consider the worst-case scenario in which none of IMDs will be able to receive the service from the server due to poor channel conditions, or in another perspective, constraints **C1** and **C2** of problem $\mathscr{P}1$ are satisfied for each IMD. In this case, the number of IMDs in the network, i.e., N, would be zero, followed by the zero value for the initial estimation of the number of UAVs. This means that even in the worst-case scenario, the search space has at least one answer and is non-empty.

It is worth mentioning that problem $\mathscr{P}1$ cannot be solved directly by mathematical methods such as gradient-based algorithms which are only effective in solving small-scale problems and are not efficient and cost-effective for sophisticated problems. Therefore, in the next section, we solve this problem effectively by utilizing the decomposition method, which is a conventional scheme in the existing literature and has many applications [41], [42], [43], [44]. In this way, we are able to solve each sub-problem by the multi-variable fixed iterative algorithm.

4 Proposed Solution for Problem $\mathscr{P}1$

To tackle problem $\mathscr{P}1$, we first decompose the optimization problem into two subproblems, including UAV positioning and user scheduling along with the resource association. Then, we employ an iterative algorithm to find a near-optimal solution for the optimization problem. We solve these sub-problems by applying the proposed modified 3D placement of UAVs algorithm, and the proposed user scheduling based on successive convex optimization and successive methods, respectively.

4.1 Subproblem I (3D Placement of UAVs)

For any given $\mathcal{S}_2 \triangleq (\mathbf{Q}, \boldsymbol{\nu}^t, \boldsymbol{\nu}^e, \mathbf{w})$, the 3D UAV placement optimization problem is simplified as

$$\mathscr{P}2) \ \min_{\mathcal{S}_1} \ \sum_{m=1}^{\tilde{M}} \varphi_m$$

$$\textbf{s.t.}$$

$$\textbf{C3} \text{ and } \textbf{C8} \backsim \textbf{C9} \text{ of } \mathscr{P}1,$$

where $\mathcal{S}_1 \overset{\Delta}{=} \left(\boldsymbol{\varphi}, \mathbf{x}_u, \mathbf{y}_u, \mathbf{h}_u\right)$. In order to solve the subproblem $\mathscr{P}2$, we first estimate the number of required UAVs in the network. Then, we propose an algorithm to optimally determine their 3D placement.

4.1.1 Initial Estimation of the Number of Required UAVs

In the first stage, we present an initial estimation of the number of UAVs serving all IMDs in the network so that the requirements of the achievable rate and the capacity pertinent to each UAV are satisfied. The goal behind determining the initial estimation of the number of UAVs is to reduce the complexity of the proposed algorithm, which can be utilized in a dynamic network topology with mobile

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

devices. Moreover, it has been proven that each UAV covers a very large geographical area by adjusting its altitude in the optimum height for coverage constraint in our proposed algorithm, while it is not feasible in existing UDHNs due to the high data traffic requirement of IMDs [32]. The initial estimation method is performed in two steps. Denoting ψ as the capacity constraint of each UAV, we first find N^{UAV} as the maximum number of IMDs that a UAV can serve, calculated as $N^{UAV} = \left\lfloor \frac{\psi}{r^{min}_{j_m}} \right\rfloor$, where $r^{min}_{j_m}$ is the minimum rate of each IMD overlaid by the corresponding UAV. In the second step, the number of UAVs is estimated by $\hat{M} = \min\{\tilde{M}, M\}$, where $\tilde{M} = \left\lceil \frac{N}{N^{UAV}} \right\rceil$.

Other methods can be used for obtaining the minimum number of UAVs by maintaining the QoS, such as i) exhaustive search started from one UAV to the number of available UAVs, i.e., M, regardless of the initial estimation, and ii) random search. However, one drawback of these schemes is their high computational complexity in obtaining the best value of M. Therefore, using the initial estimation of the number of UAVs, we could approach the optimal value with the least possible iterations.

4.1.2 Optimal 3D Placement of UAVs

In the second stage, we classify all IMDs according to their horizontal coordinations by solving problem $\mathscr{P}\mathbf{2}$. To deal with the optimization problem, we propose an efficient algorithm based on an unsupervised ML-based method. In this regard, we find the best solution for UAVs' locations so that the following utility function $\mathscr{U}_1$ is minimized

$$\mathscr{U}_1 = \sum_{s=1}^{N_A} \left\{ \sum_{m=1}^{\hat{M}} \left\{ N^{UAV} \rho_{m,s} \right\} - D_s \mathcal{A}_s \right\}. \tag{21}$$

Note that the obtained solution for the utility function $\mathscr{U}_1$ is considered as an initial solution for UAVs' placement employed in utility function $\mathscr{U}_2$, expressed as

$$\mathscr{U}_2 = \begin{cases} -\sum_{j_m \in \mathcal{I}'} \eta_{j_m} & \text{if (20) is satisfied by IMD } j_m, \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

Moreover, the solution of the optimization problem $\mathscr{P}\mathbf{2}$ is improved by minimizing the utility function $\mathscr{U}_1$. Eventually, to guarantee that the constraint **C3** is satisfied, we utilize the utility function $\mathscr{U}_3$, expressed as

$$\mathscr{U}_3 = \begin{cases} -N^{\upsilon} + r^{min}_{j_m} - r_{j_m} & \text{if (20) is satisfied by each IMD,} \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

In our framework, Algorithm 1 generates matrix $\mathbf{G} = [\mathbf{g}_j]_{1 \leq j \leq \Omega}$ including the random positions of all UAVs within the interest area, where $\mathbf{g}_j \in \mathbb{C}^{1 \times 3\hat{M}}$ denotes j^{th} row of the matrix and Ω is the total number of the initial population. Accordingly, matrix $\mathbf{G}$ is expressed as

$$\mathbf{G} \triangleq \begin{bmatrix} \mathbf{g}_1 & g_1^1 & \cdot & \cdot & \cdot & g_1^d & \cdot & \cdot & \cdot & g_1^{3\hat{M}} \\ \cdot & \cdot & & & & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot & & & & \cdot \\ \mathbf{g}_i & g_i^1 & \cdot & \cdot & \cdot & g_i^d & \cdot & \cdot & \cdot & g_i^{3\hat{M}} \\ \cdot & \cdot & & & & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot & & & & \cdot \\ \mathbf{g}_\Omega & g_\Omega^1 & \cdot & \cdot & \cdot & g_\Omega^d & \cdot & \cdot & \cdot & g_\Omega^{3\hat{M}} \end{bmatrix}, \tag{24}$$

where g_j^d denotes d^{th} element of j^{th} member. To describe the varying increments of g_i in each iteration of the algorithm, let define

$$\mathcal{U}^{\text{worst}} = \max\{\mathcal{U}^i,\ i = 1, \ldots, \Omega\}, \tag{25}$$

$$\mathcal{U}^{\text{best}} = \min\{\mathcal{U}^i,\ i = 1, \ldots, \Omega\}, \tag{26}$$

where $\mathcal{U}^i$ denotes the fitness function pertinent to i^{th} member of matrix $\mathbf{G}$. In fact, $\mathcal{U}^{\text{worst}}$ and $\mathcal{U}^{\text{best}}$ refer to the maximum and minimum of fitness value in the current generation. Moreover, the normalized fitness of i^{th} member, denoted by $\mathcal{U}_n^i$, is computed as

$$\mathcal{U}_n^i = \frac{\mathcal{U}^i - \mathcal{U}^{\text{worst}}}{\sum_{j=1}^{\Omega} \left(\mathcal{U}^j - \mathcal{U}^{\text{worst}} \right)}. \tag{27}$$

In designing AI algorithms to solve optimization problems, it is essential to consider two main concepts of i) Exploration meaning that the population of each generation can explore well the feasible region of the problem to obtain the best solution, and ii) Exploitation meaning the good utilization of the experience gained in the previous generation to improve the obtained solution. Based on the above arguments, in the proposed Algorithm 1, the population is updated in three phases. In the first and second phases, we use the information of the worst and best solutions ever achieved by i^{th} member of matrix $\mathbf{G}$ to update this solution. To be more precise, in the first phase, $\mathbf{g}_i$ is updated as follows:

$$\mathbf{g_i} = \begin{cases} \mathbf{g}_i + \mathbf{v}_i, & \text{if } \mathcal{U}^{i,\text{update}} \leq \mathcal{U}^i, \\ \mathbf{g}_i, & \text{otherwise,} \end{cases} \tag{28}$$

where $\mathcal{U}^{i,\text{update}}$ denotes the fitness value related to i^{th} updated member of matrix $\mathbf{G}$, i.e., $\mathbf{g}_i + \mathbf{v}_i$, and $\mathbf{v}_i \triangleq [\mathrm{v}_i^d]_{1 \leq d \leq 3\hat{M}}$, in which v_i^d is expressed as

$$\mathrm{v}_i^d = \zeta \left(\left(2 - \frac{\mathcal{U}_n^i}{\mathcal{U}_n^{\text{best}}} \right) \times g_i^d - g_{\text{worst}}^d \right), \tag{29}$$

where ζ is a random variable belonging to the interval (0,1). In the second phase, $\mathbf{g}_i$ is updated as follows:

$$\mathbf{g}_i = \begin{cases} \mathbf{g}_i + \mathbf{d}_i, & \text{if } \mathcal{U}^{i,\text{update}} \leq \mathcal{U}^i, \\ \mathbf{g}_i, & \text{otherwise,} \end{cases} \tag{30}$$

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

where $\mathbf{d}_i \triangleq [d_i^d]_{1 \leq d \leq 3\hat{M}}$, in which

$$d_i^d = \zeta\left(g_{\text{best}}^d - \left(2 - \frac{\mathcal{U}_n^i}{\mathcal{U}_n^{\text{best}}}\right) \times g_i^d\right). \tag{31}$$

More precisely, phases 1 and 2 (i.e., Eqs. (28) and (30)) are considered to determine the search direction based on information obtained by the best and worst members of the population. Another important point in the design of such intelligent algorithms is to employ a strategy to avoid intelligent agents stuck in the local optimal solution, which is considered in the third phase of the proposed algorithm. This approach is widely used in the design of many well-known intelligent algorithms such as GA [45], TLBO [46], and GWO [47]. According to the above arguments, if there is not enough improvement in the solution, then we employ the third phase of updating $\mathbf{g}_i$ to further improve the solution as follows:

$$\mathbf{g}_i = \begin{cases} 0.6\ \mathbf{g}_i + 0.4\ \mathbf{g}_{\text{best}}, & \textbf{if } \mathcal{U}_n^{\text{best}} - \mathcal{U}_n^i \geq 0.9\ \mathcal{U}_n^{\text{best}}, \\ \mathbf{g}_i, & \text{otherwise.} \end{cases} \tag{32}$$

This algorithm is run iteratively to find a feasible solution for 3D UAV placement so as to satisfy the traffic constraints. After obtaining a desirable solution, we aim to reduce the number of UAVs by omitting the ones whose elimination cannot influence the network quality. To this end, a UAV can be removed at each iteration. Then, the constraints are checked, and this process should be iteratively operated until the optimal solution is achieved. Details of the proposed scheme are illustrated in a pseudo-code in Algorithm 1.

Remark 1. One of the strengths of the proposed Algorithm 1 compared to many existing schemes, e.g., [48], [49], is that it avoids using any adjustment parameters in achieving the optimum solution. In fact, the existence of such a setting parameter in an algorithm requires adjusting its optimal value in the first step, which is a complex task and leads to reducing the performance of the algorithm, considerably. Accordingly, the absence of such parameters in proposed scheme results in the performance improvement.

4.2 Subproblem II (User Scheduling)

For any given $\mathcal{S}_1 \overset{\Delta}{=} (\boldsymbol{\varphi}, \mathbf{x}_u, \mathbf{y}_u, \mathbf{h}_u)$, obtained by solving $\mathscr{P}\mathbf{2}$ from *Algorithm 1*, the user scheduling optimization problem can be simplified as

$$\mathscr{P}\mathbf{3})\ \min_{\mathcal{S}_2} \quad \|\boldsymbol{\nu}^t\|_0 + \|\boldsymbol{\nu}^e\|_0$$
$$\textbf{s.t.}$$
$$\mathbf{C1} \backsim \mathbf{C7} \text{ of } \mathscr{P}\mathbf{1},$$

where $\mathcal{S}_2 \overset{\Delta}{=} (\mathbf{Q}, \boldsymbol{\nu}^t, \boldsymbol{\nu}^e, \mathbf{w})$. Considering the combinatorial nature of the objective function and non-convex constraints, we can conclude that subproblem $\mathscr{P}\mathbf{3}$ is non-convex. We will handle this intractable optimization problem by employing smoothed ℓ_p-norm function along with the SCA method.

4.3 Convex and Smooth Approximation for Objective Function

To find a smooth approximation for the objective function, we utilize a non-convex ℓ_p-norm minimization $(0 < p \leqslant 1)$ method to make the original ℓ_0-norm minimization problem more tractable and find a tight surrogate of the ℓ_0-norm in the objective functions [50], expressed as

$$f(\boldsymbol{\nu}^t, \boldsymbol{\nu}^e) = \|\boldsymbol{\nu}^t\|_p^p + \|\boldsymbol{\nu}^e\|_p^p \overset{(a)}{=} \lim_{p \to 0}\left(\sum_{j_m \in \boldsymbol{\chi}} \left|\nu_{j_m}^t\right|^p + \sum_{j_m \in \boldsymbol{\chi}} \left|\nu_{j_m}^e\right|^p\right), \tag{33}$$

where (a) comes from the following expression:

$$\|\boldsymbol{y}\|_0 = \lim_{p \to 0} \|\boldsymbol{y}\|_p^p = \lim_{p \to 0} \sum |y_i|^p. \tag{34}$$

Since Eq. (33) is still non-smooth, we provide a more efficient and tractable algorithm using the following smoothed version of ℓ_p-norm, expressed as

$$f(\boldsymbol{\nu}^t, \boldsymbol{\nu}^e; \eta) = \sum_{j_m \in \boldsymbol{\chi}} \left(\left(\nu_{j_m}^t\right)^2 + \eta^2\right)^{p/2} + \sum_{j_m \in \boldsymbol{\chi}} \left(\left(\nu_{j_m}^e\right)^2 + \eta^2\right)^{p/2}, \tag{35}$$

where $\eta > 0$ represents a small adjustment regularization parameter. In the following, we first prove that if η is small enough, any optimal answer obtained for smoothed ℓ_p-norm function $f(\boldsymbol{\nu}^t, \boldsymbol{\nu}^e; \eta)$ is also optimal for the original non-smooth ℓ_p-norm function $f(\boldsymbol{\nu}^t, \boldsymbol{\nu}^e; 0)$. To investigate the optimality of the smoothed ℓ_p-norm function (35), we define the set of Karush-Kuhn-Tucker (KKT) points of this problem, expressed as follows:

$$\mathcal{P}(\eta) = \{\boldsymbol{\nu} \in \mathcal{Z} : 0 \in \nabla_{\boldsymbol{\nu}} f(\boldsymbol{\nu}; \eta) + \mathcal{L}_{\mathcal{Z}}(\boldsymbol{\nu})\}, \tag{36}$$

where $\mathcal{Z}$ is an arbitrary convex set and $\mathcal{L}_{\mathcal{Z}}(\boldsymbol{\nu})$ can be defined as follows:

$$\mathcal{L}_{\mathcal{Z}}(\boldsymbol{\nu}) := \{\boldsymbol{j} : \langle \boldsymbol{x} - \boldsymbol{\nu}, \boldsymbol{j} \rangle \leqslant 0, \forall \boldsymbol{x} \in \mathcal{Z}\}. \tag{37}$$

The deviation of a specified set $\mathcal{V}_1$ from other set $\mathcal{V}_2$ can be calculated as [51]

$$\mathbb{D}(\mathcal{V}_1, \mathcal{V}_2) = \sup_{\boldsymbol{\nu}_1 \in \mathcal{V}_1}\left(\inf_{\boldsymbol{\nu}_2 \in \mathcal{V}_2} \|\boldsymbol{\nu}_1 - \boldsymbol{\nu}_2\|\right). \tag{38}$$

To show the relation between smoothed ℓ_p-norm function $f(\boldsymbol{\nu}^t, \boldsymbol{\nu}^e; \eta)$ and the original non-smooth ℓ_p-norm function $f(\boldsymbol{\nu}^t, \boldsymbol{\nu}^e; 0)$, we employ a theory, written as follows:

$$\lim_{\eta \searrow 0} \mathbb{D}(\mathcal{P}(\eta), \mathcal{P}(0)) = 0. \tag{39}$$

Proof. Proof is available in *Appendix B* of the supplementary material, available online. □

Now, we focus on obtaining a KKT point for minimizing $f(\boldsymbol{\nu}^t, \boldsymbol{\nu}^e; \eta)$, yielding suitable approximations for the KKT points of the non-smooth ℓ_p-norm problem $f(\boldsymbol{\nu}^t, \boldsymbol{\nu}^e; 0)$. Then, inspired by the expectation-maximization method [52], [53],

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

we substitute a convex upper bound for the smoothed ℓ_p-norm problem according to the following proposition.

Algorithm 1. Proposed Algorithm for 3D Placement of UAVs

```
Input: Generate an initial population, i.e., $\mathbf{G}(0)$, composed of $\Omega$ random members as $\mathbf{g}_i(0), i = 1, \ldots, \Omega$, of size $3\hat{M}$.
$t \leftarrow 1$, $\mathscr{U} \leftarrow \mathscr{U}_1$
$\mathcal{U}^{\text{best}} \leftarrow \min\{\mathcal{U}^i(0), i = 1, \ldots, \Omega\}$,
$\mathcal{U}^{\text{worst}} = \max\{\mathcal{U}^i(0), i = 1, \ldots, \Omega\}$.
while $\mathcal{U}^{\text{best}} > -\xi N^{\upsilon}$ do
  Calculate $\mathcal{U}^{\text{worst}}$ and $\mathcal{U}^{\text{best}}$ based on (25) and (26).
  for $i = 1, \ldots, \Omega$
    Calculate $\mathcal{U}_n^i(t)$ based on (27).
    Calculate $\mathbf{v}_i(t)$ based on (29).
    Calculate $\mathbf{d}_i(t)$ based on (31).
    Calculate $\mathcal{U}^i(t)$ based on (28), (30), (32).
    if $\mathcal{U}^i(t) < \mathcal{U}^{\text{best}}$ then
      $\mathcal{U}^{\text{best}} \leftarrow \mathcal{U}^i(t)$.
    end if
  end for
  Update $\mathcal{U}^{\text{worst}}$ based on (25).
  if $\mathcal{U}^{\text{best}} \leq 0$ then
    $\mathscr{U} \leftarrow \mathscr{U}_2$
  end if
  if $\mathcal{U}^{\text{best}} \leq -\xi N^{\upsilon}$ then
    $\mathscr{U} \leftarrow \mathscr{U}_3$
  end if
  $t \leftarrow t + 1$
end while
Output: $\mathcal{S}_1^*$.
```

Proposition 1. *For any given $\nu(\upsilon)$ in υ^{th} iteration, a convex upper bound for the smoothed ℓ_p-norm objective function (35) is obtained as follows:*

$$\mathcal{D}(\boldsymbol{\nu}; \xi(\upsilon)) = \sum_{j_m \in \boldsymbol{\chi}} \xi_{j_m}^t(\upsilon)\left(\boldsymbol{\nu}_{j_m}^t\right)^2 + \sum_{j_m \in \boldsymbol{\chi}} \xi_{j_m}^e(\upsilon)\left(\boldsymbol{\nu}_{j_m}^e\right)^2, \tag{40}$$

where $\xi_{j_m}^t(\upsilon)$ and $\xi_{j_m}^e(\upsilon)$ are defined as

$$\xi_{j_m}^t(\upsilon) = \frac{p}{2}\left[\left(\boldsymbol{\nu}_{j_m}^t(\upsilon)\right)^2 + \eta^2\right]^{\frac{p}{2}-1}. \tag{41}$$

$$\xi_{j_m}^e(\upsilon) = \frac{p}{2}\left[\left(\boldsymbol{\nu}_{j_m}^e(\upsilon)\right)^2 + \eta^2\right]^{\frac{p}{2}-1}. \tag{42}$$

Proof. Proof is available in *Appendix C* of the supplementary material, available online. □

Proposition 2. *By minimizing the upper bound (40), the smoothed ℓ_p-norm objective function (35) can be minimized successively. Besides, it is proved that there exists at least a locally optimal solution.*

Proof. Proof is available in *Appendix D* of the supplementary material, available online. □

Besides, $\overline{f}(\mathcal{S}_2; \mathcal{S}_2(\upsilon))$ is defined as [54]

$$\begin{aligned}\overline{f}(\mathcal{S}_2; \mathcal{S}_2(\upsilon)) \triangleq\, & \frac{\gamma_{\boldsymbol{\nu}^t}}{2}\left\|\boldsymbol{\nu}^t - \boldsymbol{\nu}^t(\upsilon)\right\|^2 + \frac{\gamma_{\boldsymbol{\nu}^e}}{2}\left\|\boldsymbol{\nu}^e - \boldsymbol{\nu}^e(\upsilon)\right\|^2 \\ & + \frac{\gamma_{\mathbf{Q}}}{2}\sum_{m \in \mathbb{M}}\left\|\mathbf{Q} - \mathbf{Q}(\upsilon)\right\|^2 + \frac{\gamma_{\boldsymbol{w}}}{2}\left\|\mathbf{w} - \mathbf{w}(\upsilon)\right\|^2.\end{aligned} \tag{43}$$

By adding $\overline{f}(\mathcal{S}_2; \mathcal{S}_2(\upsilon))$ to (40), the final objective function is defined as follows:

$$\begin{aligned}\tilde{f}(\mathcal{S}_2; \mathcal{S}_2(\upsilon)) = & \sum_{j_m \in \boldsymbol{\chi}} \xi_{j_m}^t(\upsilon)\left(\nu_{j_m}^t\right)^2 + \sum_{j_m \in \boldsymbol{\chi}} \xi_{j_m}^e(\upsilon)\left(\nu_{j_m}^e\right)^2 \\ & + \bar{f}(\mathcal{S}_2; \mathcal{S}_2(\upsilon)).\end{aligned} \tag{44}$$

In fact, $\overline{f}(\mathcal{S}_2; \mathcal{S}_2(\upsilon))$ guarantees that the obtained upper bound is strongly convex. In addition, constraints **C1** $\backsim$ **C3** of $\mathscr{P}$**3** are still non-convex. In the sequel, we employ the SCA method to convexify the non-convex constraints efficiently. It should be noted that the objective function (44) satisfies the criteria A1-A3 described in [55, Section 2].

4.4 Convex Approximation for Constraints C1 $\backsim$ C3

Since constraint **C1** is not convex, we obtain a desirable convex approximation for this constraint through applying the formula $\log_2\left(\det\left(\mathbf{A}\mathbf{B}^{-1}\right)\right) = \log_2(\det(\mathbf{A})) - \log_2(\det(\mathbf{B}))$. Therefore, the achievable rate is rewritten as (45),

$$\begin{aligned} & r_{j_m}(\mathbf{Q}, w_m) \\ & = \underbrace{w_m \log_2\left(\det\left(\mathrm{R}_m\left(\mathbf{Q}_{-j_m}, w_m\right) + \lambda \bar{\mathbf{H}}_{j_m,u_m}^H \bar{\mathbf{H}}_{j_m,u_m} \mathbf{Q}_{j_m}\right)\right)}_{r_{j_m}^+(\mathbf{Q}, w_m)} \\ & \quad - \underbrace{w_m \log_2 \det\left(\mathrm{R}_m\left(\mathbf{Q}_{-j_m}, w_m\right)\right)}_{r_{j_m}^-(\mathbf{Q}_{-j_m}, w_m)},\end{aligned} \tag{45}$$

where $r_{j_m}^+(\mathbf{Q}, w_m)$ and $-r_{j_m}^-(\mathbf{Q}_{-j_m}, w_m)$ are concave and convex functions, respectively. By the view of the fact that the sum of concave and convex functions is non-convex, we construct a suitable convex surrogate, denoted by $\tilde{r}_{j_m}(\mathbf{Q}, w_m; \mathbf{Q}(\upsilon), w_m(\upsilon))$, and substitute it into (45), expressed as (46), where $\nabla_{Q_{j_m}^*} r_{j_m}^-(\mathbf{Q}_{-j_m}(\upsilon), w_m)$ and $\nabla_{w_m^*} r_{j_m}^-(\mathbf{Q}_{-j_m}, w_m(\upsilon))$ represent the conjugate gradient of $r_{j_m}^-$ with respect to $\mathbf{Q}_{j_m}$ and w_m calculated at $(\mathbf{Q}_{-j_m}(\upsilon), w_m)$ and $(\mathbf{Q}_{-j_m}, w_m(\upsilon))$, respectively, and given by (47) and (48) (see *Appendix E* of supplementary material for the proof, available online). Substituting (46) into (13), we can obtain the convex approximation for constraint **C1**, expressed as

$$\begin{aligned} & \tilde{r}_{j_m}(\mathbf{Q}, w_m; \mathbf{Q}(\upsilon), w_m(\upsilon)) \\ & = r_{j_m}^+(\mathbf{Q}, w_m) - r_{j_m}^-(\mathbf{Q}_{-j_m}(\upsilon), w_m(\upsilon)) \\ & \quad - \sum_{j=1}^{m-1}\left\langle \nabla_{\mathbf{Q}_{j_m}^*} r_{j_m}^-(\mathbf{Q}_{-j_m}(\upsilon), w_m), \mathbf{Q}_{j_m} - \mathbf{Q}_{j_m}(\upsilon)\right\rangle \\ & \quad - \left\langle \nabla_{w_m^*} r_{j_m}^-(\mathbf{Q}_{-j_m}, w_m(\upsilon)), w_m - w_m(\upsilon)\right\rangle,\end{aligned} \tag{46}$$

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

$$\nabla_{Q^*_{j_m}} r^-_{j_m}(\mathbf{Q}_{-j_m}(\upsilon), w_m) = 2w_m \mathbf{H}^H_{j_m,u_m} \mathrm{R}_m\big(\mathbf{Q}_{-j_m}(\upsilon), w_m\big)^{-1} \mathbf{H}_{j_m,u_m}, \tag{47}$$

$$\nabla_{w^*_m} r^-_{j_m}(\mathbf{Q}_{-j_m}, w_m(\upsilon)) = -2\lambda \sum_{j'=1}^{j-1} \mathrm{Re}\Big(\mathrm{tr}\Big[\mathbf{H}^H_{j'_m,u_m} \mathbf{R}^{-1}_m (\mathbf{Q}_{-j'_m}, w_m) \mathbf{H}_{j'_m,u_m} \mathbf{Q}_{j'_m}\Big]\Big) + \frac{r^-_{j_m}(\mathbf{Q}_{-j_m}, w_m(\upsilon))}{w_m(\upsilon)}. \tag{48}$$

$$\mathbf{C1}'.\ \frac{\beta_{j_m}}{\tilde{r}_{j_m}(\mathbf{Q}, w_m)} - T_{j_m} \leqslant \nu^t_{j_m},\ \forall j_m \in \boldsymbol{\chi}. \tag{49}$$

In the next step, we calculate an inner convex substitution for the constraint **C2**, while the constraints B1 ∽ B6 in [55, Section 2] are satisfied. Hence, the constraint **C2** is rewritten as

$$\mathbf{C2}'.\ \mathrm{tr}\big(\mathbf{Q}_{j_m}\big) - \frac{E^{loc}_{j_m}}{\beta_{j_m}} \tilde{r}_{j_m}(\mathbf{Q}, w_m) \leqslant \nu^e_{j_m}. \tag{50}$$

Eventually, the convex approximation of constraint **C3** is rewritten as follows:

$$\mathbf{C3}'.\ r^{min}_{j_m} - \tilde{r}_{j_m}(\mathbf{Q}, w_m) \leqslant 0. \tag{51}$$

Accordingly, for any given feasible point $\mathcal{S}_2(\upsilon)$, we define the strongly convex surrogate for subproblem $\mathscr{P}3$ as follows:

$$\begin{aligned}
\mathscr{P}4)\ \hat{\mathcal{S}}_2(\mathcal{S}_2(\upsilon)) = &\min_{\mathcal{S}_2}\ \tilde{f}(\mathcal{S}_2; \mathcal{S}_2(\upsilon)) \\
&\mathrm{s.t.} \\
&\mathbf{C1}'.\ \frac{\beta_{j_m}}{\tilde{r}_{j_m}(\mathbf{Q}, w_m)} - T_{j_m} \leqslant \nu^t_{j_m},\ \forall j_m \in \boldsymbol{\chi}, \\
&\mathbf{C2}'.\ \mathrm{tr}\big(\mathbf{Q}_{j_m}\big) - \frac{E^{loc}_{j_m}}{\beta_{j_m}} \tilde{r}_{j_m}(\mathbf{Q}, w_m) \leqslant \nu^e_{j_m},\ \forall j_m \in \boldsymbol{\chi}, \\
&\mathbf{C3}'.\ r^{min}_{j_m} - \tilde{r}_{j_m}(\mathbf{Q}, w_m) \leqslant 0,\ \forall j_m \in \boldsymbol{\chi}, \\
&\mathbf{C4}.\ \nu^t_{j_m} \geqslant 0,\ \nu^e_{j_m} \geqslant 0,\ \forall j_m \in \boldsymbol{\chi}, \\
&\mathbf{C5}.\ w_m \geqslant 0,\ \forall m \in \mathbb{M}, \\
&\mathbf{C6}.\ \sum_{m=1}^{\hat{M}} w_m \leqslant \mathcal{W}, \\
&\mathbf{C7}.\ \mathbf{Q}_{j_m} \in \mathcal{Q}_{j_m},\ \forall j_m \in \boldsymbol{\chi},
\end{aligned}$$

where $\hat{\mathcal{S}}_2$ represents the unique solution of the subproblem $\mathscr{P}4$. *Algorithm 2* is iteratively run to solve the subproblem $\mathscr{P}4$ around the current iteration $\mathcal{S}^*_2(\upsilon) \in \mathcal{Z}$, where $\mathcal{Z}$ denotes the feasible set of $\mathscr{P}4$. We summarize the steps of our proposed method as pseudocode in *Algorithm 2* by detail.

Remark 2. The goal behind employing Algorithm 2 is to maximize the number of scheduled users by gradually eliminating the users with the largest value of $\boldsymbol{\mathcal{L}} \triangleq \frac{\boldsymbol{\nu}^t}{\sum_{j_m \in \boldsymbol{\chi}} \nu^t_{j_m}} + \frac{\boldsymbol{\nu}^e}{\sum_{j_m \in \boldsymbol{\chi}} \nu^e_{j_m}}$. In fact, parameter $\boldsymbol{\mathcal{L}}$ specifies the relative amount of total violations related to delay and energy constraints by each IMD with respect to all users. It is worth mentioning that normalizing $\boldsymbol{\nu}^t$ and $\boldsymbol{\nu}^e$ by $\sum_{j_m \in \boldsymbol{\chi}} \nu^t_{j_m}$ and $\sum_{j_m \in \boldsymbol{\chi}} \nu^e_{j_m}$, respectively, guarantee that two constraints are measured on an equal scale. The proposed method is repeated until a group of IMDs, indexed by the set $\boldsymbol{\chi}^*$, is finally obtained such that each of the entries of vector $\boldsymbol{\mathcal{L}}$ returns a value close to zero, implicating the feasibility of offloading under the criteria of $\mathscr{P}4$. Eventually, *Algorithm 2* finds o^* as a possible solution so that the set of scheduled IMDs offloading their computational task to the server can be determined as $\boldsymbol{\chi}^* \triangleq \{\mathcal{L}_1, \ldots, \mathcal{L}_{o^*}\}$. In particular, upon attaining the desirable solution to problem $\mathscr{P}4$, the set of IMDs is arranged in an ascending order according to the relative values of the vector $\boldsymbol{\mathcal{L}}$. Then, the subset $\boldsymbol{\chi}^*$ of scheduled IMDs is determined via the bisection method. This method is based on finding the minimum number of IMDs in the interval $[0, N]$, in which the number of IMDs should be eliminated so that other users are able to be scheduled for offloading, while the constraints are satisfied. In particular, the set $\boldsymbol{\chi}^*$ is defined as $\boldsymbol{\chi}^* \triangleq \{\mathcal{L}_1, \ldots, \mathcal{L}_{o^*}\}$, as explained in Algorithm 2, where the value of $o^* \in [0, N]$ is obtained by sequentially searching within the interval $[\ell^{low}, \ell^{up}]$, which is initialized as $[0, N]$. In each iteration, the search interval should be halved by the midpoint o. Then, the feasibility check should be carried out to test. In order to perform the feasibility test for $\boldsymbol{\chi}^{[o]} \triangleq \{\pi_1, \ldots, \pi_o\}$, the following problem should be solved

$$\begin{aligned}
\mathscr{F}\Big(\boldsymbol{\chi}^{[o]}\Big):\ &\text{find } \{\mathbf{Q}, \mathbf{w}\} \\
&\mathrm{s.t.} \\
&\mathbf{C1}.\ \frac{\beta_{j_m}}{\tilde{r}_{j_m}(\mathbf{Q}, w_m)} - T_{j_m} \leqslant 0,\ \forall j_m \in \boldsymbol{\chi}^{[o]}, \\
&\mathbf{C2}.\ \frac{\beta_{j_m} \mathrm{tr}\big(\mathbf{Q}_{j_m}\big)}{\tilde{r}_{j_m}} \leqslant E^{loc}_{j_m}(\mathbf{Q}, w_m),\ \forall j_m \in \boldsymbol{\chi}^{[o]}, \\
&\mathbf{C3}.\ r^{min}_{j_m} \leqslant \tilde{r}_{j_m}(\mathbf{Q}, w_m),\ \forall j_m \in \boldsymbol{\chi}^{[o]}, \\
&\mathbf{C4}.\ w_m \geqslant 0,\ \forall m \in \mathbb{M}, \\
&\mathbf{C5}.\ \sum_{m=1}^{\hat{M}} w_m \leqslant \mathcal{W}, \\
&\mathbf{C6}.\ \mathbf{Q}_{j_m} \in \mathcal{Q}_{j_m},\ \forall j_m \in \boldsymbol{\chi}^{[o]},
\end{aligned}$$

4.5 Complexity and Convergence Analysis

4.5.1 Complexity Analysis

In summary, Algorithms 1 and 2 were proposed to find an optimal solution for subproblems $\mathscr{P}2$ and $\mathscr{P}4$ in an iterative manner via employing the SCA technique until a near-optimal solution is achieved for problem $\mathscr{P}1$. In this subsection, we analyze the computational complexity of Algorithms 1 and 2, and the overall algorithm, expressed as the following propositions.

Proposition 3. The computational complexity of Algorithm 1 is of order $\mathcal{O}\big(t\Omega\hat{M}\big)$, in which t represents the number of iterations needed until convergence.

Proof. Proof is available in *Appendix F* of the supplementary material, available online. □

It can be seen from Proposition 3 that any increase in the value of Ω leads to increasing the computational complexity of the algorithm. However, this increment to a certain amount leads to enhancing the exploration ability, and

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

accordingly, the search speed of the algorithm in the search space increases. Depending on the nature of the problem, it is possible to determine the upper bound of the search agents experimentally, and there is no specific rule for determining the number of search agents. In this case and to adjust the acceptable value for the number of search agents in the algorithm, a reasonable trade-off between the computational complexity and the convergence speed must be considered.

Algorithm 2. Proposed Algorithm for Solving Problem $\mathscr{P}4$.

```
Input: Parameters set in Table 1, the position of IMDs, $\boldsymbol{\mathcal{S}}_2(0) \in \mathcal{Z}$, $\lambda(\upsilon) \in (0,1]$, $p = 0.5$, $\upsilon = 0$, $\xi^t_{j_m} = \xi^e_{j_m} = 0$, $\varepsilon > 0$, $\boldsymbol{\chi} = \mathcal{I}$, $\ell^{low} = 0$, $\ell^{up} = N$, $M$ and $N$.
1:  Calculate $\hat{\boldsymbol{\mathcal{S}}}_2(\boldsymbol{\mathcal{S}}_2(\upsilon))$ by solving $\mathscr{P}4$.
2:  if $\left|\tilde{f}\left(\hat{\boldsymbol{\mathcal{S}}}_2(\boldsymbol{\mathcal{S}}_2(\upsilon))\right) - \tilde{f}(\boldsymbol{\mathcal{S}}_2(\upsilon))\right| \leqslant \delta$
3:      ${\boldsymbol{\mathcal{S}}_2}^{final} \leftarrow \hat{\boldsymbol{\mathcal{S}}}_2$.
4:      Break.
5:  else
6:      Set $\boldsymbol{\mathcal{S}}_2(\upsilon+1) \leftarrow \boldsymbol{\mathcal{S}}_2(\upsilon) + \lambda\left(\hat{\boldsymbol{\mathcal{S}}}_2(\boldsymbol{\mathcal{S}}_2(\upsilon)) - \boldsymbol{\mathcal{S}}_2(\upsilon)\right)$.
7:      Set $\xi^t_{j_m}(\upsilon+1) \leftarrow \frac{p}{2}\left[\left(\nu^t_{j_m}(\upsilon+1)\right)^2 + \varepsilon^2\right]^{\frac{p}{2}-1}$.
8:      Set $\xi^e_{j_m}(\upsilon+1) \leftarrow \frac{p}{2}\left[\left(\nu^e_{j_m}(\upsilon+1)\right)^2 + \varepsilon^2\right]^{\frac{p}{2}-1}$.
9:      Set $\upsilon \leftarrow \upsilon + 1$.
10:     return to step 1.
11: end if
12: Obtain $\boldsymbol{\mathcal{L}} \triangleq \frac{\boldsymbol{\nu}^t}{\sum_{j_m \in \boldsymbol{\chi}} \nu^t_{j_m}} + \frac{\boldsymbol{\nu}^e}{\sum_{j_m \in \boldsymbol{\chi}} \nu^e_{j_m}}$.
13: Arrange IMDs in an ascending order based on the value of $\mathcal{L}$ as $\mathcal{L}_{\pi_1} \leqslant \mathcal{L}_{\pi_2} \leqslant \cdots \leqslant \mathcal{L}_{\pi_{(\ell^{up}-\ell^{low})}}$.
14: Do
15:     Set $o \leftarrow \left\lfloor \frac{\ell^{up}+\ell^{low}}{2} \right\rfloor$.
16:     Check the feasibility test for $\boldsymbol{\chi} = \boldsymbol{\chi}^{[o]} \triangleq \{\pi_1, \ldots, \pi_o\}$.
17:     if problem $\mathscr{P}4$ was feasible for $\boldsymbol{\chi}^{[o]}$
18:         Set $\ell^{low} \leftarrow o$.
19:     else
20:         Set $\ell^{up} \leftarrow o$.
21:     endif
22: While $\ell^{up} - \ell^{low} = 1$.
23: Set $o^* \leftarrow \ell^{low}$.
24: Set $\boldsymbol{\chi}^* \leftarrow \{\pi_1, \ldots, \pi_{o^*}\}$.
Output: scheduled users $\boldsymbol{\chi}^*$.
```

Proposition 4. By employing the Interior Points Method (IPM), the computational complexity of Algorithm 2 tackled by SCA is of order $\mathcal{O}\Big(\max\Big\{\sum_{\ell=1}^{j}\big(\mathcal{F}_\ell \Theta_\ell^2 \Xi_\ell \Big(|\boldsymbol{\chi}_\ell| + 7\sum_{j_m \in \boldsymbol{\chi}_\ell} A^3_{j_m}\Big)\big)\Big\}\Big)$, where $\boldsymbol{\chi}_\ell$ denotes the number of users performing offloading in ℓ^{th} implementation of the algorithm. For instance, in the first implementation of the SCA scheme, i.e., $\ell = 1$, in which all IMDs perform offloading, $|\boldsymbol{\chi}_1| = N$ is considered. Moreover, let define the set $\mathcal{F} \overset{\Delta}{=} \{\mathcal{F}_1, \ldots, \mathcal{F}_L\}$, where $\mathcal{F}_\ell$, $\ell \in \{1, \ldots, L\}$, denotes the number of iterations needed for the convergence of the SCA algorithm in ℓ^{th} implementation.

Proof. Proof is available in *Appendix G* of the supplementary material, available online. □

TABLE 1
Simulation Parameters

Parameter	Value
n	2
ξ	95
M	15
(δ, α)	$(10^{-3}, 10^{-5})$
$p^{\max}_{j_m}$	23 dBm
f_c	2 GHz
$[\xi^{LoS}, \xi^{NLoS}]$	$[1, 20]$ dB
k_m	10^{-26}
$\mathcal{W}$	20 MHz
N_0	-174 dBm/Hz
β_{j_m}	uniform $[300-800]$ KB
υ_{j_m}	uniform $[0.1-1]$ Gigacycles
T_{j_m}	uniform $[10-20]$ sec
$\mathcal{A}$	100 km^2

It can be seen that the computational complexity is strongly impressed by the MIMO structure, which is considered as the nature of the problem. In other words, increasing the number of antennas (i.e., A) leads to an increase in the computational complexity of the problem in the order of A^7. Naturally, if IMDs and UAVs are assumed to be equipped with a single antenna, i.e. $A = 1$, the computational complexity to solve such a complex problem is of order $\mathcal{O}\Big(\sum_{l=1}^{L}\mathcal{F}_\ell|\boldsymbol{\chi}_\ell|^4\Big)$, which is quite reasonable for such complicated problem.

Based on Propositions 3 and 4, it is concluded that the computational complexity of solving problem $\mathscr{P}1$ mainly depends on Algorithm 2, which is of order $\mathcal{O}\Big(A^7\sum_{l=1}^{L}\mathcal{F}_\ell|\boldsymbol{\chi}_\ell|^4\Big)$, because the complexity of Algorithm 2 is dominant. To get more insight into this claim, we compare the computational complexity of Algorithms 1 and 2 with a practical example. By setting $\Omega = 100$, $t = 10$, and $\hat{M} = 30$, it is concluded that the computational complexity of Algorithm 1 is of order $\mathcal{O}(t\Omega\hat{M}) = \mathcal{O}(10^4)$. On the other hand, by considering a network with 1000 users, $L = 10$, $\mathcal{F}_\ell = 10$ in average, and $A = 2$, the computational complexity of Algorithm 2 is of the order $\mathcal{O}\Big(A^7\sum_{l=1}^{L}\mathcal{F}_\ell|\boldsymbol{\chi}_\ell|^4\Big) = \mathcal{O}(2^7 \times 10^{14})$. It can be seen that the order of complexity of Algorithm 1 is not comparable to that of Algorithm 2 (i.e., $\mathcal{O}(10^3) \ll \mathcal{O}(2^7 \times 10^{14})$). Hence, the computational complexity of Algorithm 2 is dominant.

4.5.2 Convergence Analysis

For the proof of convergence analysis refer to the *Appendix H* of the supplementary material, available online.

5 Simulation Results

In this section, we present some simulation results to verify the validity of the proposed joint user scheduling and 3D multi UAV placement. In our simulations, we consider a multi UAV-aided MEC system, where IMDs are distributed in a geographical area of size 10km × 10km. Considering an urban environment, we set $a = 9.61$ and $b = 0.16$ at $f_c = 2$ GHz. Throughout the simulation, $N = 1000$ is considered, unless we mention other values for the number of IMDs. The remaining important simulation parameters

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

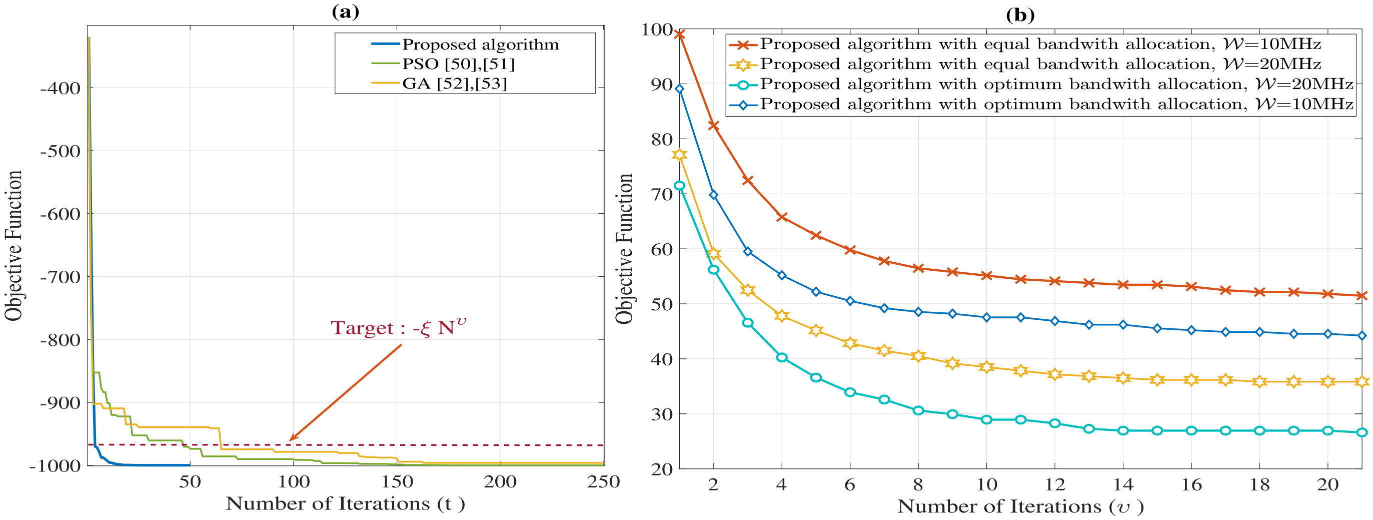


Fig. 2. a) Comparison of the convergence speed of the proposed Algorithm 1 with the PSO [56], [57] and GA [58], [59] schemes, b) convergence speed of Algorithm 2 versus the number of IMDs for equal bandwidth and optimal bandwidth allocations by considering $\mathcal{W} = 10$ and 20 MHz.

TABLE 2
Studied Scenarios and Performance Comparison of the Proposed Algorithm, Denoted by PA, and the PSO [56], [57] and GA [58], [59] Algorithms

Scenarios	N	N_A	Scenario Description	No. of UAVs				No. of served IMDs		
				$\hat{M}$	PSO	GA	PA	PSO	GA	PA
Scenario A	1000	2	Subarea 1 is normally distributed, Subarea 2 is uniformly distributed $(\mathcal{A}_1, \mathcal{D}_1)$ = (35 km^2, 60) $(\mathcal{A}_2, \mathcal{D}_2)$ = (65 km^2, 40)	30	29	29	28	976	971	989
Scenario B	1000	2	Subarea 1 is uniformly distributed, Subarea 2 is uniformly distributed $(\mathcal{A}_1, \mathcal{D}_1)$ = (35 km^2, 60) $(\mathcal{A}_2, \mathcal{D}_2)$ = (65 km^2, 40)	30	29	28	28	981	978	987
Scenario C	1000	4	All subareas are uniformly distributed $(\mathcal{A}_1, \mathcal{D}_1)$ = (33.3 km^2, 35) $(\mathcal{A}_2, \mathcal{D}_2)$ = (16.6 km^2, 40) $(\mathcal{A}_3, \mathcal{D}_3)$ = (16.6 km^2, 5) $(\mathcal{A}_4, \mathcal{D}_4)$ = (33.3 km^2, 20)	30	28	28	27	987	980	989

adopted in performance evaluation are summarized in Table 1. Note that all tests were performed on a PC featuring an Intel Xeon CPU and running Windows 10 Professional. The clock of the machine is set to 2.66 GHz with a 48 GB memory. Further, we employ Matlab software as a simulation platform.

In Fig. 2a, we evaluate the objective function, or equivalently, the convergence speed in subproblem $\mathscr{P}2$ using Algorithm 1 versus the number of iterations $t \in [0, 250]$. To make a fair comparison and considering similar assumptions for all schemes under simulation, we compare the number of iterations needed until convergence for the proposed Algorithm 1 with the PSO [56], [57] and GA [58], [59] schemes. As seen from this figure, the stopping criterion of the algorithm, with the target $-\xi N^{\nu}$, is depicted in Fig. 2a. It is observed that the proposed Algorithm 1 converges after 15 iterations, while, the number of iterations for the convergence of the PSO [56], [57] and GA [58], [59] schemes are 130 and 160, respectively. Based on the above results, it is concluded that the proposed Algorithm 1 outperforms the PSO [56], [57] and GA [58], [59] schemes considerably. In addition, the objective function in subproblem $\mathscr{P}4$ using Algorithm 2 versus the number of iterations, or equivalently, the number of IMDs varying in the range $\nu \in [0, 22]$ is investigated in Fig. 2b. Considering $\mathcal{W} = 10$ and 20, we employ two strategies for bandwidth allocation: Equal bandwidth and optimum bandwidth allocations. In the first method, $\mathcal{W}$ is divided into equal parts employed for each UAV's transmission, while in the second method, the bandwidth allocation is optimized using Algorithm 2. The results show that employing the optimum bandwidth allocation leads to at least 18% improvement in the objective function value in comparison with the equal bandwidth allocation.

Then, we investigate the performance of the proposed method to confirm the theoretical analysis mentioned in the previous sections in three scenarios as described in Table 2 in detail. We determine the number of subareas and the user densities related to the same subarea in each scenario. More precisely, we assume that 60% of the users are normally distributed in subarea 1 with a concentration in the area's center, while other users are uniformly distributed in the second subarea. Besides, in Scenario B, the interest area is divided into two equal subareas, and 40% and 60% of IMDs are uniformly distributed in the left and right subareas, respectively. In addition, the area of study in Scenario

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

TABLE 3
Measured CPU Running Time in Seconds and the Number of Iterations (i.e., n_{I^*})

Parameter	Ω	CPU time (sec)	n_{I^*}
Proposed Algorithm 1	25	334	21
Proposed Algorithm 1	50	509	16
PSO [56], [57]	25	568	63
PSO [56], [57]	50	813	49
GA [58], [59]	25	483	104
GA [58], [59]	50	716	81

C is divided into 4 subareas, and users are uniformly distributed with different user densities as specified in Table 2.

The advantage of using the initial estimation for the number of UAVs is well illustrated in Table 2. It can be seen that the initial estimation of the number of UAVs for scenario A (i.e., $\hat{M} = 30$) is very close to its optimal value (i.e., 28), and finally, just two UAVs are diagnosed as redundant and eliminated. This means that the initial estimation has reached the minimum number of UAVs with two iterations that are acceptable for such an optimization subproblem.

Next, we investigate the number of UAV servers and IMDs served by UAVs. Based on the results in Table 1, we can perceive that the total number of UAVs deployed in each subarea is proportional to the users' density in the respective region. Considering $\hat{M} = 30$ as the estimated number of UAVs in scenario A, the number of UAVs employed in the proposed Algorithm 1, PSO [56], [57], and GA [58], [59] schemes are 28, 29, and 29, respectively. Therefore, we can conclude that the initial estimation of the number of UAVs is very close to the optimum value and this leads to a significant reduction in the computational complexity.

Comparing the initial estimated (i.e., $\hat{M}$) and the optimal values for the number of UAVs applied in the network, we can see that the proposed Algorithm 1 outperforms its counterparts, i.e., the PSO [56], [57] and the GA [58], [59] schemes. In other words, using the proposed Algorithm 1, a fewer number of UAVs have been used to serve users than the other two algorithms, so that the QoS is preserved. Another superiority of the proposed algorithm is related to the number of IMDs served by UAVs compared to PSO [56], [57] and GA [58], [59] algorithms. In each scenario, it is shown that more IMDs are served via the proposed algorithm compared to the PSO scheme in [56], [57] and the GA in [58], [59]. For instance, 98.9%, 97.6%, and 97.1% users in scenario A are served by the above algorithms, respectively, leading to at least 1.3% and 1.8% improvement in the percentage of served IMDs. The information related to other scenarios has been specified in detail in Table 2.

Moreover, in Table 3, we compare the proposed Algorithm 1 with PSO [56], [57] and GA [58], [59] in terms of CPU time (in seconds) and the number of required iterations, denoted by n_{I^*}, until the proposed Algorithm 1 and other schemes converge. For this purpose, we independently simulated these algorithms 1000 times and illustrated the average number of iterations to obtain the desired solutions, i.e., satisfying the termination criteria. The results in Table 3 indicate that the proposed Algorithm 1 converges faster than the PSO [56], [57] and GA [58], [59] algorithms for $\Omega = 25, 50$ in scenario C. Note that increasing the number of members Ω leads to decreasing the number of iterations needed until convergence; however, the CPU time for the convergence of these algorithms increases. In other words, they can achieve their desired solutions in lower iterations; however, they need more CPU time since they should perform more operations in each iteration. These results are also confirmed by the computational complexity analysis presented in Section 4.5. Based on Proposition 3, increasing Ω leads to increasing the computational complexity of Algorithm 1. In this regard, the time complexity, followed by the CPU time, increases as well. Furthermore, increasing Ω leads to enhancing the exploration ability. In this case, it is a well-known fact for artificial intelligence-based algorithms such as PSO and GA, using more agents (by increasing Ω), the algorithm can converge to the desired solution with fewer iterations.

Figs. 3a, 4a, and 5a depict a snapshot of the optimal 2D view related to UAV placement and the user association

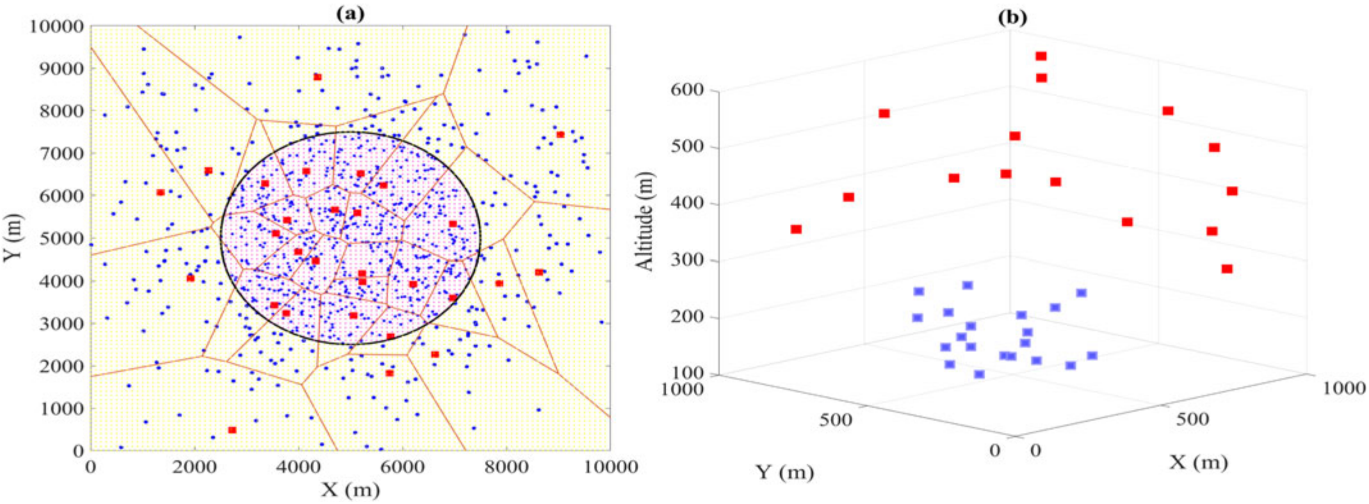


Fig. 3. Scenario A: $a)$ User association and 2D placement of UAVs, where IMDs and UAVs are shown by blue dots and red squares, respectively. $b)$ UAVs' altitude, where the UAVs are depicted with blue and red squares in subarea 1 and 2, respectively.

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

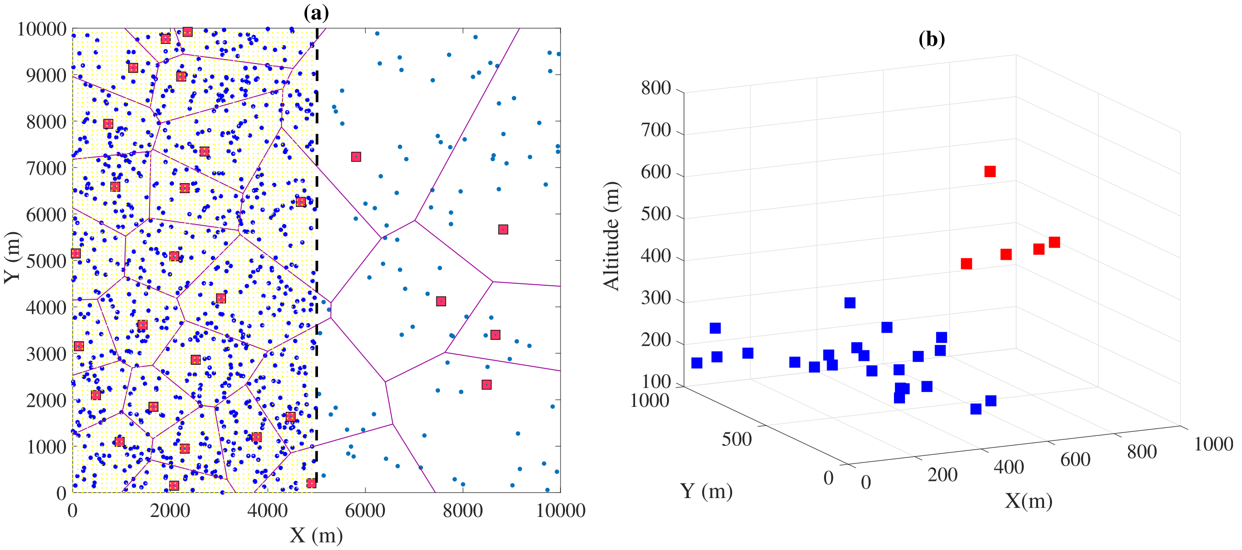


Fig. 4. Scenario B: $a)$ User association and 2D placement of UAVs, where IMDs and UAVs are shown by blue dots and red squares, respectively. $b)$ UAVs' altitude, where the UAVs are depicted with blue and red squares in subarea 1 and 2, respectively.

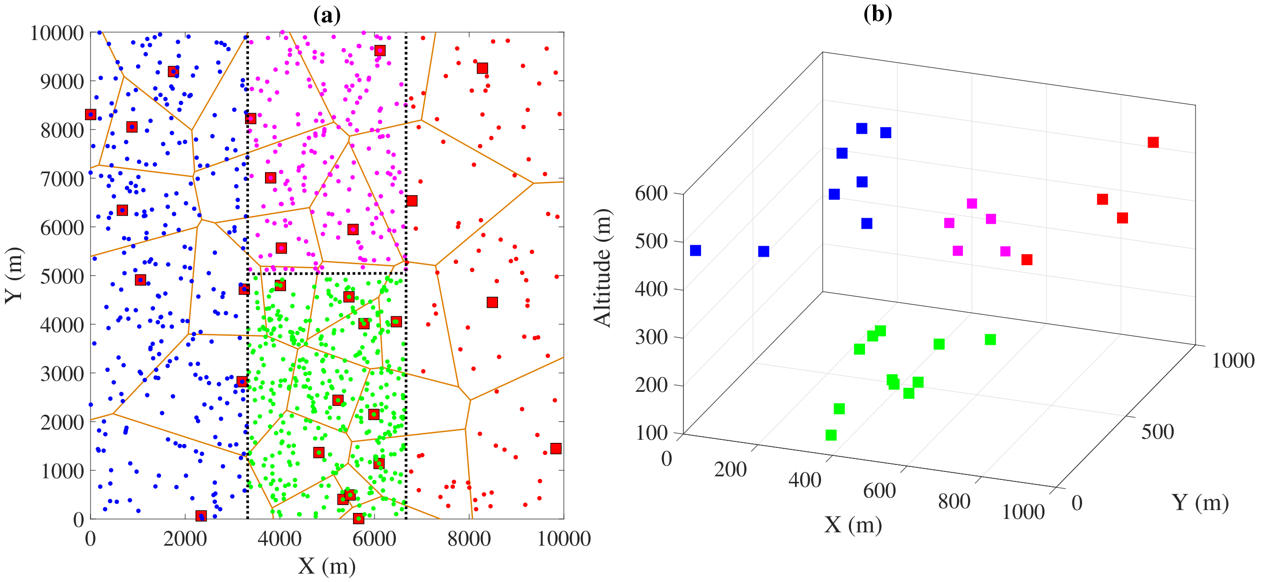


Fig. 5. Scenario C: $a)$ User association and 2D placement of UAVs, where IMDs are shown by blue, pink, green, and red dots in subareas 1, 2, 3, and 4, respectively. UAV are depicted by red squares. $b)$ UAVs' altitude, where the UAVs are depicted with blue, pink, green, and red squares in subareas 1, 2, 3, and 4, respectively.

based on the proposed algorithm in Scenarios A, B, and C. Besides, the Voronoi tessellation of UAVs provides a better insight into the UAVs' placement. Note that the continuous lines do not display the actual boundaries for the IMDs connection. Further, the optimal values for UAVs' height are shown in Figs. 3b, 4b, and 5b. Accordingly, the 3D placement of UAVs are shown in the aforementioned figures, obtained by Algorithm 1. It is shown that more UAVs have been deployed to serve users in the area with a higher user density compared to the less crowded area in each scenario. In fact, when an area has a higher density, all the UAV resources are utilized by close users. Accordingly, the UAVs usually are not able to be further away from the users, and they decrease their altitude to reduce the imposed interference by other IMDs in nearby clusters. On the other hand, in the area with a lower user density, UAVs raise their altitude to increase their coverage on the ground.

In Fig. 6, we evaluate the percentage of users served by UAVs versus the number of IMDs, varying in the range [250,2500] for two cases: $i)$ employing the proposed algorithm, $ii)$ not applying Algorithm 2. As seen, any increase in the number of IMDs causes a decrease in the percentage of users served by UAVs. In fact, by raising the number of IMDs, the number of requests sent to the server increases, leading to an increase in the interference imposed by other users. Accordingly, the traffic requirements, such as power and delay constraints, are not satisfied, and the number of users served by UAVs declines. Moreover, compared to the case without applying Algorithm 2, employing the proposed Algorithm leads to an 18% improvement in the

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

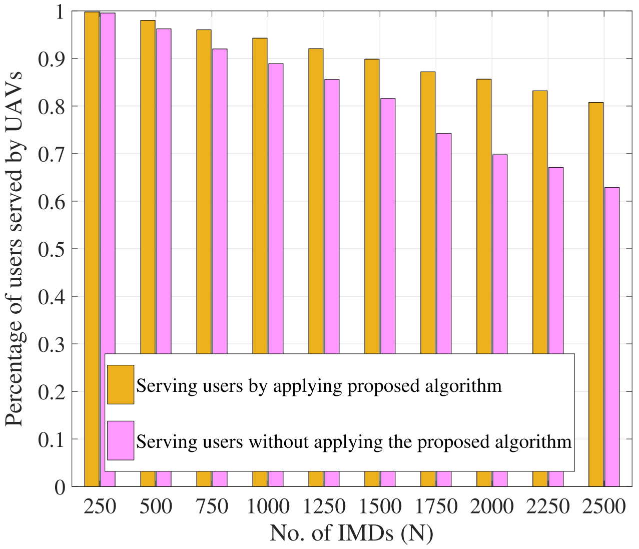


Fig. 6. Comparison of the percentage of serving users by applying Algorithm 2 with the case of not employing the algorithm versus the number of IMDs.

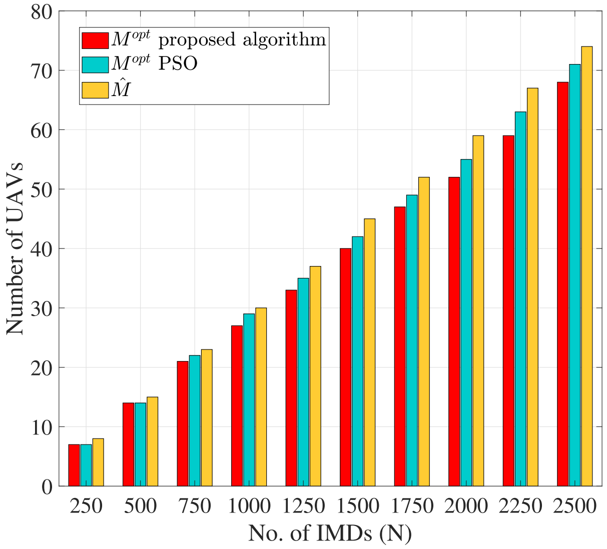


Fig. 7. Comparison of the number of required UAVs by adopting Algorithm 1, PSO scheme, and the initial approximation (i.e., $\hat{M}$) versus the number of IMDs.

percentage of users who UAVs serve by considering $N = 2500$ as the total number of IMDs in the network.

In Fig. 7, we investigate the optimum number of UAVs required for serving IMDs versus the number of IMDs for our proposed Algorithm 1 and the PSO scheme. In addition, the estimated number of UAVs for each number of IMDs is depicted in this figure. As shown in Fig. 7, increasing the number of IMDs leads to raising the number of required UAVs to which IMDs offload their tasks. Compared to the PSO algorithm, fewer UAVs are deployed as servers by applying the proposed algorithm, decreasing the network cost related to operating UAVs. For instance, the number of estimated UAVs required to serve $N = 2500$ IMDs is 74, while 2 and 5 UAVs are redundant by applying the PSO and the proposed Algorithm 1, indicating the superiority of our scheme.

Remark 3. The proposed algorithm is employed at the UAV server when a considerable amount of traffic data is generated in UDHNs under temporary adventures when the terrestrial BSs cannot serve all the IMDs or when ground BSs are broken due to vandalism, bad weather conditions, and transmission problems. In this case, the optimal number of UAVs and their 3D positions are determined to maximize the number of IMDs served by UAVs. Unlike previous researches, there is no limitation for the user distribution in the UAV-enabled network, and the proposed algorithm obtains an optimal solution for the number of UAVs and their locations without considering the user densities in the network. Moreover, the delay and energy consumption of IMDs are regarded as practical constraints in the optimization algorithm to maximize the number of served IMDs.

6 Conclusion

In this paper, we considered a UDHN consisting of IMDs and UAV servers with multiple antennas. We aimed to maximize the number of IMDs offloading their tasks to the UAVs by optimizing the user association and the number of UAV servers deployed in optimal 3D placements. To this end, we formulated a mixed-integer non-convex optimization problem subject to some QoS constraints. To make the problem more tractable, we divided the problem into two subproblems: 3D UAV placements along with the user association and the user scheduling. We proposed an ML-based algorithm for the first subproblem and calculated the computational complexity of the proposed algorithm, indicating its superiority compared to existing algorithms in the literature review. Considering the second subproblem, i.e., user scheduling, we utilized the ℓ_p -norm concept to find a smooth upper bound. Then, we computed the first-order Taylor series to use a convex surrogate for the non-convex constraints, and the SCA method was applied to solve the second subproblem. Finally, the process was run iteratively until the overall algorithm converges. In this way, the near-optimal solution was obtained for the optimization problem.

Acknowledgments

The work of Jamshid Abouei and Konstantinos N. Plataniotis was supported in part by the Natural Sciences and Engineering Research Council of Canada NSERC. In addition, the work of Jamshid Abouei was performed when he was a visiting professor in the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada.

References

[1] N. Nouri, P. Rafiee, and A. Tadaion, “NOMA-based energy-delay trade-off for mobile edge computation offloading in 5G networks,” in *Proc. IEEE 9th Int. Symp. Telecommun.*, Dec. 2018, pp. 522–527.

[2] Z. Hajiakhondi-Meybodi , A. Mohammadi, and J. Abouei, “Deep reinforcement learning for trustworthy and time-varying connection scheduling in a coupled UAV-based femtocaching architecture,” *IEEE Access*, vol. 9, pp. 32263–32281, Feb. 2021.

[3] X. Lin, J. Li, J. Wu, H. Liang, and W. Yang, “Making knowledge tradable in edge-AI enabled IoT: A consortium blockchain-based efficient and incentive approach,” *IEEE Trans. Ind. Inform.*, vol. 15, no. 12, pp. 6367–6378, Dec. 2019.

[4] Z. Yu, Y. Gong, S. Gong, and Y. Guo, “Joint task offloading and resource allocation in UAV-enabled mobile edge computing,” *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147–3159, Apr. 2020.

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

[5] Y. Zhang, D. Niyato, and P. Wang, “Offloading in mobile cloudlet systems with intermittent connectivity,” *IEEE Trans. Mobile Comput.*, vol. 14, no. 12, pp. 2516–2529, Dec. 2015.

[6] B. Li, K. Wang, D. Xue, and Y. Pei, “K-means based edge server deployment algorithm for edge computing environments,” in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov.*, 2018, pp. 1169–1174.

[7] X. Liu, A. Liu, T. Qiu, B. Dai, T. Wang, and L. Yang, “Restoring connectivity of damaged sensor networks for long-term survival in hostile environments,” *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1205–1215, 2019.

[8] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, “Edge intelligence: Paving the last mile of artificial intelligence with edge computing,” *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[9] M. McClellan, C. Cervelló-Pastor, and S. Sallent, “Deep learning at the mobile edge: Opportunities for 5G networks,” *Appl. Sci.*, vol. 10, no. 14, 2020, Art. no. 4735.

[10] N. H. Motlagh, T. Taleb, and O. Arouk, “Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives,” *IEEE Internet Things J.*, vol. 3, no. 6, pp. 899–922, Dec. 2016.

[11] V. Kouhdaragh, F. Verde, G. Gelli, and J. Abouei, “On the application of machine learning to the design of UAV-based 5G radio access networks,” *Electronics*, vol. 9, no. 4, 2020, Art. no. 689. [Online]. Available: https://www.mdpi.com/2079–9292/9/4/689

[12] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications,” *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Sep. 2017.

[13] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, “Placement optimization of UAV-mounted mobile base stations,” *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 604–607, 2016.

[14] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage,” *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1647–1650, Aug. 2016.

[15] Q. Zhang, M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Machine learning for predictive on-demand deployment of UAVs for wireless communications,” in *Proc. IEEE Glob. Commun. Conf.*, 2018, pp. 1–6.

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

[17] Q. Wu, Y. Zeng, and R. Zhang, “Joint trajectory and communication design for multi-UAV enabled wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

[18] M. Hua, L. Yang, Q. Wu, and A. L. Swindlehurst, “3D UAV trajectory and communication design for simultaneous uplink and downlink transmission,” *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5908–5923, Sep. 2020.

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

[20] S. Jeong, O. Simeone, and J. Kang, “Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2017.

[21] N. Zhao *et al.*, “Joint trajectory and precoding optimization for UAV-assisted NOMA networks,” *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3723–3735, May 2019.

[22] G. Jiangchun, D. Guoru, X. Yitao, W. Haichao, and W. Qihui, “Proactive optimization of transmission power and 3D trajectory in UAV-assisted relay systems with mobile ground users,” *Chinese J. Aeronaut.*, vol. 34, no. 3, pp. 129–144, Mar. 2021.

[23] X. Zhou, Q. Wu, S. Yan, F. Shu, and J. Li, “UAV-enabled secure communications: Joint trajectory and transmit power optimization,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4069–4073, Apr. 2019.

[24] N. Nouri, J. Abouei, A. R. Sepasian, M. Jaseemuddin, A. Anpalagan, and K. N. Plataniotis, “3D multi-UAV placement and resource allocation for energy-efficient IoT communication,” *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1322–1333, Feb. 2021.

[25] X. Pang, J. Tang, N. Zhao, X. Zhang, and Y. Qian, “Energy-efficient design for mmWave-enabled NOMA-UAV networks,” *Sci. China Inf. Sci.*, vol. 64, pp. 1–14, Apr. 2021.

[26] X. Duan, S. Ao, W. Feng, J. Tang, and J. Hu, “Joint 3D placement and multi-beam design for UAV-assisted wireless power transfer networks,” *Phys. Commun.*, vol. 44, pp. 1–8, Feb. 2021.

[27] W. Feng *et al.*“Joint 3D trajectory design and time allocation for UAV-enabled wireless power transfer networks,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9265–9278, Sep. 2020.

[28] R. Zhang, X. Pang, J. Tang, Y. Chen, N. Zhao, and X. Wang, “Joint location and transmit power optimization for NOMA-UAV networks via updating decoding order,” *IEEE Wireless Commun. Lett.*, vol. 10, no. 1, pp. 136–140, Jan. 2021.

[29] Z. Xiong *et al.*, “UAV-assisted wireless energy and data transfer with deep reinforcement learning,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 1, pp. 85–99, Mar. 2021.

[30] K. Li, W. Ni, E. Tovar, and M. Guizani, “Joint flight cruise control and data collection in UAV-aided internet of things: An onboard deep reinforcement learning approach,” *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9787–9799, Jun. 2021.

[31] K. Li, W. Ni, E. Tovar, and A. Jamalipour, “Deep Q-learning based resource management in UAV-assisted wireless powered IoT networks,” in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–6.

[32] A. Al-Hourani , S. Kandeepan, and S. Lardner, “Optimal LAP altitude for maximum coverage,” *IEEE Wirel. Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Jul. 2014.

[33] C.-C. Lai, C.-T. Chen, and L.-C. Wang, “On-demand density-aware UAV base station 3D placement for arbitrarily distributed users with guaranteed data rates,” *IEEE Wirel. Commun. Lett.*, vol. 8, no. 3, pp. 913–916, Feb. 2019.

[34] A. Ahmed, M. Awais, T. Akram, S. Kulac, M. Alhussein, and K. Aurangzeb, “Joint placement and device association of UAV base stations in IoT networks,” *Sensors*, vol. 19, no. 9, pp. 21–57, Jan. 2019.

[35] L. Sboui, H. Ghazzai, Z. Rezki, and M.-S. Alouini, “Achievable rates of UAV-relayed cooperative cognitive radio MIMO systems,” *IEEE Access*, vol. 5, pp. 5190–5204, 2017.

[36] N. Nouri, A. Entezari, J. Abouei, M. Jaseemuddin, and A. Anpalagan, “Dynamic power–latency tradeoff for mobile edge computation offloading in NOMA-based networks,” *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2763–2776, Apr. 2020.

[37] N. Nouri, J. Abouei, M. Jaseemuddin, and A. Anpalagan, “Joint access and resource allocation in ultradense mmWave NOMA networks with mobile edge computing,” *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1531–1547, Feb. 2020.

[38] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, “Energy-optimal mobile cloud computing under stochastic wireless channel,” *IEEE Trans. Wirel. Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[39] T. D. Burd and R. W. Brodersen, “Processor design for portable systems,” *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 13, no. 2/3, pp. 203–221, 1996.

[40] H. Ghazzai, E. Yaacoub, M.-S. Alouini, Z. Dawy, and A. Abu-Dayya , “Optimized LTE cell planning with varying spatial and temporal user densities,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1575–1589, 2015.

[41] Z. Wang, Z. Lin, T. Lv, and W. Ni, “Energy-efficient resource allocation in massive MIMO-NOMA networks with wireless power transfer: A distributed ADMM approach,” *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2021.3068721.

[42] A. Azizi, N. Mokari, and M. R. Javan, “Joint radio resource allocation, 3D placement and user association of aerial base stations in IoT networks,” 2017, *arXiv: 1710.05315*.

[43] L. Sboui, H. Ghazzai, Z. Rezki, and M.-S. Alouini, “Achievable rates of UAV-relayed cooperative cognitive radio MIMO systems,” *IEEE Access*, vol. 5, pp. 5190–5204, 2017.

[44] F. Fang, H. Zhang, J. Cheng, S. Roy, and V. C. M. Leung, “Joint user scheduling and power allocation optimization for energy-efficient NOMA systems with imperfect CSI,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 12, pp. 2874–2885, Dec. 2017.

[45] J. H. Holland, “Genetic algorithms,” *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, July 1992.

[46] R. V. Rao, *Teaching Learning Based Optimization Algorithm*. Cham, Switzerland: Springer, 2016.

[47] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.

[48] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, 1995, pp. 1942–1948.

[49] E. Rashedi, H. Nezamabadi-Pour , and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.

[50] D. Ge, X. Jiang, and Y. Ye, "A note on the complexity of Lp minimization," *Math. Program.*, vol. 129, no. 2, pp. 285–299, 2011.

[51] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*. Philadelphia, PA, USA: SIAM, 2014.

[52] K. Lange and J. S. Sinsheimer, "Normal/independent distributions and their applications in robust regression," *J. Comput. Graphical Statist.*, vol. 2, no. 2, pp. 175–198, 1993.

[53] D. Ba, B. Babadi, P. L. Purdon, and E. N. Brown, "Convergence and stability of iteratively re-weighted least squares algorithms," *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 183–195, Jan. 2014.

[54] N. Nouri and A. Tadaion, "Energy optimal resource allocation for mobile edge computation offloading in presence of computing access point," in *Proc. IEEE Iran Workshop Commun. Inf. Theory*, Jul. 2018, pp. 1–6.

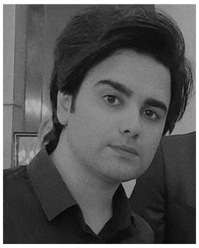
[55] G. Scutari, F. Facchinei, and L. Lampariello, "Parallel and distributed methods for constrained nonconvex optimization: Part I: Theory," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929–1944, Apr. 2017.

[56] M. A. Ali and A. Jamalipour, "UAV placement and power allocation in uplink and downlink operations of cellular network," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4383–4393, Jul. 2020.

[57] X. Li, H. Yao, J. Wang, X. Xu, C. Jiang, and L. Hanzo, "A near-optimal UAV-aided radio coverage strategy for dense urban areas," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 9098–9109, Sep. 2019.

[58] N. Ceccarelli, P. A. Regis, S. Sengupta, and D. Feil-Seifer , "Optimal UAV positioning for a temporary network using an iterative genetic algorithm," in *Proc. IEEE 29th Wireless Optical Commun. Conf.*, May. 2020, pp. 1–6.

[59] Y. Chen, N. Li, C. Wang, W. Xie, and J. Xv, "A 3D placement of unmanned aerial vehicle base station based on multi-population genetic algorithm for maximizing users with different QoS requirements," in *Proc. IEEE 18th Int. Conf. Commun. Technol.*, Oct. 2018, pp. 967–972.



Nima Nouri (Student Member, IEEE) received the BSc degree in communication engineering from the Shahid Bahonar University of Kerman, Kerman, Iran, in 2014 and the MSc degree in communication systems engineering from Yazd University, Yazd, Iran, in 2017. Since 2017, he has been a research assistant with the WINEL Lab, Yazd University. His main research interests include Internet of Things, UAV communication systems, edge/fog computing, resource allocation, and nonconvex optimization.

Fahimeh Fazel (Student Member, IEEE) received the BSc degree in electrical engineering from the Amirkabir University of Technology, Tehran, Iran, in 2016 and the MSc degree in communication systems engineering from the University of Tehran, Tehran, Iran, in 2019. Her research interests include wireless communication, cache-enabled UAV networks, visible light communication, and coding theory and their applications in visible light communications.

Jamshid Abouei (Senior Member, IEEE) received the BSc degree in electronics engineering and the MSc degree (with the highest honor) in communication systems engineering from the Isfahan University of Technology, Iran, in 1993 and 1996, respectively, and the PhD degree in electrical engineering from the University of Waterloo, Canada, in 2009. In 1996, he joined the Department of Electrical Engineering, Yazd University, Iran, as a lecturer and was promoted to an assistant professor in 2010, and an associate professor in 2015. From 1998 to 2004, he was a technical advisor and design engineer with the R&D Center and Cable Design Department, SGCC, Iran. From 2009 to 2010, he was a postdoctoral fellow with the Multimedia Lab, Department of Electrical & Computer Engineering, University of Toronto, Canada, and was a research fellow with the Self-Powered Sensor Networks (ORF-SPSN) Consortium. During his sabbatical, he was an associate researcher with the Department of Electrical, Computer and Biomedical Engineering, Ryerson University, Toronto, Canada. His research interests include next generation of wireless networks (5G) and wireless sensor networks (WSNs), with a particular emphasis on PHY/MAC layer designs, including the energy efficiency and optimal resource allocation in cognitive cell-free massive MIMO networks, multi-user information theory, mobile edge computing, and femtocaching. He was the international relations chair in 27th ICEE2019 Conference, Iran, in 2019. He currently directs the Research Group, Wireless Networking Laboratory (WINEL), Yazd University. He is a member of the IEEE Information Theory and was the recipient of several awards and scholarships, including the FOE and IGSA awards for Excellence in Research from the University of Waterloo, Canada, MSRT PhD Scholarship from the Ministry of Science, Research and Technology, Iran, in 2004, Distinguished Researcher Award in the Province of Yazd, Iran, 2011, Distinguished Researcher Award from the Electrical Engineering Department, Yazd University, Iran, 2013, and Best Paper Award for the IEEE Iranian Conference on Electrical Engineering (ICEE 2018).

Konstantinos N. Plataniotis (Fellow, IEEE) received the BEng degree in computer engineering from the University of Patras, Greece, and the MS and PhD degrees in electrical engineering from the Florida Institute of Technology Melbourne, Florida. He is currently a professor with The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, where he directs the Multimedia Laboratory. His research interests include image/signal processing, machine learning and adaptive learning systems, visual data analysis, multimedia and knowledge media, and affective computing. He is a fellow of the Engineering Institute of Canada, fellow of the Canadian Academy of Engineering, and a registered professional engineer in Ontario. He was the editor-in-chief of *IEEE Signal Processing Letters*, technical co-chair of the IEEE 2013 International Conference in Acoustics, Speech and Signal Processing, inaugural IEEE Signal Processing Society Vice President for Membership (2014–2016), and the general co-chair for the 2017 IEEE GLOBALSIP. He was the general co-chair for 2018 IEEE International Conference on Image Processing (ICIP 2018) and IEEE International Acoustics, Speech and Signal Processing (ICASSP 2021) and is currently the general chair for 2027 IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP2027), Toronto, ON, Canada. He also holds the Bell Canada endowed chair in multimedia since 2014.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Authorized licensed use limited to: ASU Library. Downloaded on August 18,2023 at 20:12:01 UTC from IEEE Xplore. Restrictions apply.