Three-Dimensional Multi-UAV Placement and Resource Allocation for Energy-Efficient IoT Communication

Nima Nouri, *Student Member, IEEE*, Jamshid Abouei, *Senior Member, IEEE*, Ali Reza Sepasian, Muhammad Jaseemuddin, *Member, IEEE*, Alagan Anpalagan, *Senior Member, IEEE*, and Konstantinos N. Plataniotis, *Fellow, IEEE*

***Abstract*—This article considers the problem of an unmanned aerial vehicle (UAV)-enabled cloud network under partial computation offloading scenario, where multiple UAV-mounted aerial base stations are employed to serve a group of remote Internet of Things ground-based smart devices (ISDs). The main objective of this work is to maximize energy efficiency by minimizing the number of needed drones while minimizing the cost associated with serving the ISDs under some realistic quality of service constraints. To that end, we aim to jointly optimize the 3-D UAV placements, transmit power, and cloud resources. This represents a challenging, nonconvex, and NP-hard optimization problem. In this work, we decompose the optimization problem into three separate subproblems, namely, 2-D UAV positioning, UAV altitude optimization, and UAV-cloud resource association. These subproblems are solved using a modified global *K*-means, successive convex approximation, and successive linear programming techniques. A comprehensive simulation study and comparative evaluation against the state-of-the-art (SOTA) algorithms are conducted to demonstrate the utility of the proposed approach and its benefits in applications of interest.**

***Index Terms*—3-D placement, drone, IoT, nonorthogonal multiple access (NOMA), resource allocation, UAV-aided edge computing, unmanned aerial vehicle (UAV) communications.**

I. Introduction

With the emergence of new wireless technologies and an increase in the number of Internet of Things ground-based smart devices (ISDs), there is an explosive demand for data traffic in future ultradense wireless networks. Furthermore, the development of new applications on ISDs with compute-intensive features demands a higher data rate from the emerging 6G networks. However, the current generation of smart devices is not capable of performing delay-sensitive and compute-intensive applications. The delay and bandwidth requirements of these applications also pose several challenges for 6G wireless networks. The disparity between the computation requirements of these applications and the available compute power of smart devices drives the attention of researchers to mobile-edge computing (MEC). The MEC platform, located at the infrastructure-based edge of the network, provides cloud-computing capabilities in proximity to ISDs for improving their Quality of Experience (QoE) [1], [2]. The main challenge for conventional terrestrial MEC servers is their dependency on the ground infrastructures and environmental conditions. On the other hand, in an inaccessible area with a high traffic load, it needs an urgent deployment of a new terrestrial base station (BS) or MEC server, but the deployment may not be feasible due to the installation cost, long installation process, and environmental limitations.

We observe a paradigm shift toward unmanned aerial vehicles (UAVs)-enabled MEC systems to address the aforementioned challenges due to the cost-effectiveness, high maneuverability, flexibility, and fast deployment of UAVs to enhance the coverage and data rate [3]. Kalantari *et al.* [4] demonstrated the superiority of UAVs over conventional fixed access points in this scenario and mentions the main reason for the necessity of using UAVs in 5G and beyond wireless networks. Indeed, UAVs are able to assist conventional fixed terrestrial BSs in providing high data rate coverage wherever and whenever there is a severe need, especially when the demand occurs in difficult to predict situations. Furthermore, due to the rapid deployment of UAVs, they provide temporary coverage on *ad hoc* basis in sparsely populated or distant areas, or when the terrestrial infrastructure is damaged due to a natural disaster, inclement weather conditions, vandalism, and transmission problems. In consideration of the above benefits, the performance of UAV-assisted MEC networks is superior to terrestrial MEC to overcome the dynamic nature of traffic overloading in the network. Another key feature of UAVs is the reliable Line-of-Sight (LoS) connectivity in the direction of ISDs, which boosts the coverage area and the network's capacity when compared to deep-faded wireless networks

Manuscript received April 24, 2021; revised June 1, 2021; accepted June 15, 2021. Date of publication June 22, 2021; date of current version January 24, 2022. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) Discovery Grant. The work of Konstantinos N. Plataniotis was supported by the NSERC Discovery Grant titled "From Adaptive Signal Processing to Brain-Inspired Cognitive Systems." *(Corresponding author: Jamshid Abouei.)*

Nima Nouri and Jamshid Abouei are with the Department of Electrical Engineering, Yazd University, Yazd 123456789, Iran (e-mail: nimanouri68@gmail.com; abouei@yazd.ac.ir).

Ali Reza Sepasian is with the Department of Mathematics, Fasa University, Fasa 74616-86131, Iran.

Muhammad Jaseemuddin and Alagan Anpalagan are with the Department of Electrical, Computer and Biomedical Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada (e-mail: jaseem@ee.ryerson.ca; alagan@ee.ryerson.ca).

Konstantinos N. Plataniotis is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: kostas@ece.utoronto.ca).

Digital Object Identifier 10.1109/JIOT.2021.3091166

with static BSs. Furthermore, UAV-enabled MEC servers with the advantages of adaptive altitude and higher flexibility of drones can remarkably improve the computing performance of ISDs [5]. In spite of the aforementioned benefits, UAV-aided edge communication is facing some unprecedented challenges. In contrast to terrestrial MEC networks that operate with unlimited power supplies, drones in UAV-aided MEC networks are often powered by batteries with a limited lifetime, which constrains the execution time of the tasks processed at UAVs. Another major challenge is to optimize the 3-D placement of drones so as to have the best computing performance of ISDs. In fact, different from the terrestrial channel model, where BSs and MEC servers are located at fixed positions, and accordingly, the path loss depends on the position of ISDs, the UAV-to-ground channel model between MEC and ISDs is a function of the location of UAVs, MEC servers, and ISDs [5], [6]. Another major challenge in 5G and beyond is to maximize energy and spectrum efficiency as much as possible. To cater for the demand of spectrum resources, the nonorthogonal multiple access (NOMA) has been considered as an emerging solution for effectively utilizing limited spectrum resources while providing high performance in terms of sum rate, energy efficiency, fairness, and coverage for 5G and beyond wireless networks [7], [8]. In addition, successive interference cancelation (SIC) is utilized at the receiver side with better channel conditions to remove the messages intended for other users with weaker channel conditions before the messages of strong users are decoded.

A. Related Works

Recently, there has been a growing interest in the use of UAV-enabled MEC to address the challenges of providing high data rate coverage in a difficult environment. This framework was first introduced in [9], in which the authors indicated that by applying the UAV-aided MEC structure, the computational complexity of the network can be decreased significantly. Recent literature in UAV-aided ultradense heterogeneous networks (UDHNs) is categorized into two different research groups depending on the status of UAV's mobility when they serve terrestrial users in the network: 1) UAV placement optimization [10]–[12] and 2) UAV trajectory optimization [13]–[15]. Researches on UAV placement usually focus on optimizing the placement of UAVs as a quasistationary BS for serving users in a specific area as well as allocating communication resources [16]. On the contrary, the UAV trajectory optimization can use all the capabilities of a UAV-enabled wireless network. Due to the full control of the UAV's mobility, with the proper design of the UAV's trajectory and the communication schedule, the distance between the UAV and ground users can be reduced. Motivated by the literature review, and according to the role played by UAVs in the network, the UAV-enabled MEC system can be categorized as follows.

1) *UAV as a General ISD:* In this scenario, the UAV functions as another ISD that performs the compute-intensive tasks of the original ISD [17]. A large body of previous works examined this structure from various perspectives, including UAV placement [18], power allocation [19], and joint user association and UAV placement [20]. In [21], the simultaneous uplink and downlink transmission networks were studied. In this scenario, one UAV is connected to multiple access points working as a disseminator, while other UAVs perform as mobile BSs to collect data from different sensor nodes. The sum throughput maximization of this heterogeneous network is the main goal of the work in [21] that is achieved by the joint optimization of communication scheduling, transmission powers, and the 3-D trajectory of drones. The problem of joint power control and 3-D trajectory of multiple UAVs was studied in [22] to maximize the sum-rate aggregation of the UAV-enabled interference channel.
2) *UAV as a MEC Server:* In this scenario, the UAV performs as the MEC server to provide an execution engine to the tasks of the ISD that are offloaded to the UAV. This architecture is more suitable for the case that the UAV has ample computational capability and energy budget, or in situations such as natural disasters when the terrestrial MEC infrastructure is not available. In [23], a UAV-enabled MEC architecture was introduced that employed efficient UAV scheduling and air-to-ground cooperation to assist communication and computing of the edge network. This work was extended into a more general case where the UAV was employed as a cloudlet offering computation offloading opportunities to multiusers with limited local processing capabilities [24]. Jeong *et al.* [25] proposed a MEC-enabled UAV architecture where the energy cost of the UAV was minimized by optimizing the local computing bits along with offloading/downloading computing bits altogether. Furthermore, both NOMA and orthogonal multiple access (OMA) schemes were examined in [25]. Zhou *et al.* [5] investigated the resource allocation problem in UAV-aided MEC systems under partial and binary computation offloading modes. The main goal was to achieve the maximum weighted sum computation rate of all ISDs by optimizing the central processing unit (CPU) frequencies, user transmits powers, user offloading times, and trajectory of drones. The problem was formulated as a nonconvex problem, which was solved with a proposed multistage alternative algorithm. It was demonstrated in [5] that the computational complexity of the network is reduced as compared to its binary counterpart through partial offloading in the network.
3) *UAV as a Relay:* This scenario is adaptable for situations that UAVs are not equipped by the MEC server capability and the direct links between the MEC server and ISDs are not reliable. In this structure, UAVs work as relays to assist ISDs to offload their compute-intensive tasks to central MEC servers efficiently. Lyu *et al.* [26] proposed a hybrid wireless network by employing drones as aerial BSs working in conjunction with a terrestrial BS. The drone flies along the edge of a cell to control offloading of tasks of users located at the edge from the terrestrial BS. With the proposed framework in [26], user partitioning, spectrum allocation, and the

trajectory of UAVs are optimized to maximize the sum throughput of all ISDs subject to the max–min fairness metric. Considered as a relay, the UAV offloads the computational tasks of ISDs to the MEC server. It was demonstrated in [26] that by optimizing the UAV's trajectory and applying for the cyclical multiple access, the computation performance can be enhanced. Jiang *et al.* [27] introduced an amplify-and-forward (AF) protocol for UAVs to schedule time-division relay data among multiple pairs of users. It maximizes the minimum paired user rate by optimizing the trajectory of UAVs, transmission powers, and time slot allocation. In [28], a multihop UAV relaying system was proposed to maximize the total throughput of the network by optimizing the available resource and UAVs' trajectories subject to some limitations on the peak and average transmit powers, total spectrum bandwidth, the information-causality of multihop relaying, and collision avoidance. In [29], the throughput maximization of the UAV-aided relaying network was investigated by jointly optimizing the communication and the trajectory of UAVs, under some constraints on the UAV's mobility. To solve the derived mixed-integer nonconvex optimization problem efficiently, Zeng *et al.* [29] proposed an iterative algorithm using block coordinate descent and successive convex approximation (SCA) methods.

B. Contributions

The key contributions of this article are briefly summarized as follows.

1) This article investigates the integration of NOMA to a multi UAV-enabled MEC system, where UAVs provide MEC services to supplement ground BSs when they are damaged or overloaded. By considering the partial computation offloading mode, ISDs offload their compute-intensive tasks to UAVs. If UAVs are unable to perform all requests because of the limited computing capacity, overloaded requests are transmitted to the central cloud server.
2) The main objective in the first part of this article is to maximize the total energy efficiency by employing the minimum number of UAVs and incurring the reduced costs so that all ISDs are served subject to the constraints of CPU frequencies, the transmit power of users and UAVs, and the position of UAVs. We formulate the problem to jointly optimize the transmit power, allocation of communication and computation resources, UAVs' placement on the 3-D plane, and association of ISDs with UAVs.
3) It is demonstrated that the optimization problem is an NP-hard and nonconvex, which is intractable to solve. The underlying optimization is broken into three subproblems that are solved separately: a) optimization of the UAV's altitude; b) positioning of UAVs in a 2-D space; and c) allocation of cloud resources for UAVs. The proposed solutions are combined into a globally distributed algorithm that iteratively reaches an approximate solution with a few iterations.
4) We employ the global K-means (GKMs) clustering method with minor modifications called a modified global K-means (MGKMs) for clustering of ISDs, which effectively minimizes the clustering error. We demonstrate by simulation that the proposed MGKM clustering algorithm displays a superior performance and fast convergence when compared to the classical clustering algorithm.
5) After finding the global optimal solution for the 2-D position of drones, we apply the successive linear programming (SLP) to obtain UAVs' altitudes. Then, given the 3-D positions of UAVs, we jointly optimize the remaining parameters using the SLP method. We show that the proposed algorithm has low computational complexity compared to K-means and GKMs methods.
6) Finally, a comprehensive simulation study is conducted to evaluate the performance of the proposed multi UAV-enabled MEC offloading system. We show that a significant performance gain is achieved as compared to other benchmark schemes.

C. Article Outline

The remainder of this article is organized as follows. In Sections II and III, we describe the proposed integrated multi UAV-aided MEC systems with NOMA and mathematically formulate the problem of the optimal resource allocation, respectively. In Section IV, we introduce the proposed method of solving the optimization problem. Section VI presents the numerical results and discusses the performance of the proposed scheme. Finally, we conclude this article in Section VII.

II. System Model

This article considers a multi UAV-aided MEC network consisting of a cloud server and multiple UAV clusters (see Fig. 1). The UAVs are deployed as flying BSs, equipped with a MEC server, and provide supplementary computing resource for overlaid ground users, known as ISDs. The UAVs are able to communicate with the cloud server via a radio access network. ISDs are assumed to be uniformly distributed in a large area of size $\mathcal{A}^2$ m^2. We denote the total number of ISDs and the set of UAVs as N and $\mathcal{M} \triangleq \{1, \ldots, M\}$, respectively. Each UAV covers a cluster of ISDs so that each user[1] is covered by only one UAV. Let N_m represent the number of users covered by UAV m. We denote the set of all ISDs served by M UAVs as $\mathcal{I} \triangleq \{i_m : i = 1, \ldots, N_m, m = 1, \ldots, M\}$, so that $\sum_{m=1}^{M} N_m = N$. In addition, let us define $\mathcal{C}_m$ as the set of users overlaid by UAV m, so that $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_M\}$, $\mathcal{C}_m \cap \mathcal{C}_n = \emptyset$ *forall*$m, n \in \mathcal{M}$, and $\cup_{m \in \mathcal{M}} \mathcal{C}_m = \mathcal{I}$. Each user has a compute-intensive task that can be offloaded to UAVs. We denote the compute-intensive task related to user i_m as $\mathcal{APP}_{i_m} = \{v_{i_m}, \beta_{i_m}\}$, wherein v_{i_m} and β_{i_m} represent the computing intensity related to the ISD

[1]Throughout this article, we occasionally use the term "user" instead of ISD.

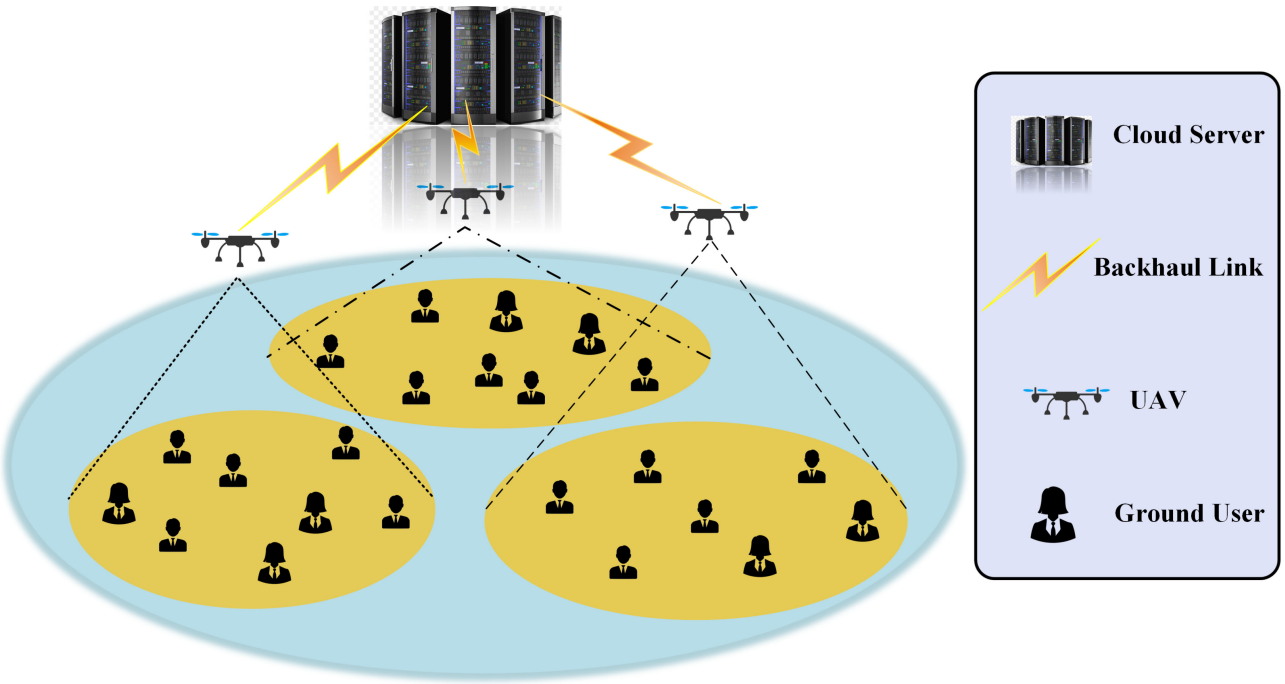

Fig. 1. Graphical illustration of an integration of drone-BSs in ultradense heterogeneous networks.

i_m in the unit of CPU cycles per bit and the size of input data in the unit of bits (containing the program code and the input parameters), respectively. A 3-D Euclidean coordinate is adopted to determine the positions of UAVs and the distance between ISDs and UAVs. The position of each ISD is assumed to be fixed and known to the corresponding UAV. The positions of ISD i_m and mth UAV are denoted by $\mathbf{p}_{i_m} = \left[x_{i_m}, y_{i_m}\right]^T \in \mathbb{R}^{2\times 1}, i_m \in \mathcal{I}$, and $\mathbf{p}_m^{\text{uav}} = \left[x_m^{\text{uav}}, y_m^{\text{uav}}, h_m^{\text{uav}}\right]^T \in \mathbb{R}^{3\times 1}, m \in \mathcal{M}$, respectively, where x_{i_m} and y_{i_m} represent the horizontal plane coordinate of i_mth ground user and h_m^{uav} denotes the flight height of the mth UAV. Therefore, the 3-D distance between ISD i_m and UAV m is determined by $d_{m,i_m} \triangleq \sqrt{\left(x_m^{\text{uav}} - x_{i_m}\right)^2 + \left(y_m^{\text{uav}} - y_{i_m}\right)^2 + \left(h_m^{\text{uav}}\right)^2}$.

In this article, the altitude of ISDs and the heights of the antenna are neglected for all ISDs and UAVs. It is assumed that the users in the network are static or move at a low velocity. Since the main focus is on the 3-D placement of UAVs, we do not take into account the velocity and flight energy consumption of UAVs in the analysis and simulation results. This assumption is widely made in the current [10], [11]. Furthermore, it is assumed that all ISDs are equipped with a single antenna, and the partial offloading technique is applied. Due to the computation and physical restrictions of ISDs, they are unable to perform their intensive tasks locally on their devices. Accordingly, they should use the UAV/cloud server and offload their computation tasks to these servers, while the UAV/cloud server can accomplish the compute-intensive offloaded task on behalf of the user. The performance improvement of multiuser computation offloading is satisfied by assuming that each ISD can access resources by deploying the NOMA scheme. Specifically, in order to make better use of the limited spectrum resources, this article considers integrating NOMA, which is widely recognized as an emerging transmission technology into UAV-aided wireless networks. In fact, NOMA increases the number of simultaneously served users, and thus, it can support massive connectivity in UDHNs similar to our system model with a large number of users. In addition, this technique can maintain the user's fairness and diverse Quality of Service (QoS) by flexible power control between the strong and weak users; particularly, as more power is allocated to a weak user, NOMA offers higher cell-edge throughput and thus, enhances the cell-edge user experience. The signals of multiple ISDs are superposed in the power domain using asymmetric channel conditions. More precisely, more power is assigned to the signal for an ISD with poor channel conditions and vice versa. Upon receiving the superposed signal, ISDs employ the SIC scheme to decode multiuser signals. In fact, the ISD with better channel conditions decodes the signals of other ISDs with poor channel conditions prior to its own message.

To capture the signal distortion based on obstacles, the extensively employed air-to-ground channel model in [30]–[32] is assumed, including nonline-of-sight (NLoS) or LoS links, with some probabilities depending on the elevation angle between the ISD and the UAV, and UAV's altitude, simultaneously. Given UAV $m \in \mathcal{M}$ with altitude h_m^{uav}, and an ISD $i_m \in \mathcal{I}$ with a distance d_{m,i_m} from the projected position of the UAV on the ground 2-D plane, the LoS connection probability is formulated as $P_{m,i_m}^{\text{LoS}} = (1/[1 + a\exp(b([180/\pi]\theta_{m,i_m} - a))])$, where a and b are constants depending on the type of environment and carrier frequency, and θ_{m,i_m} denotes the elevation angle defined as $\theta_{m,i_m} = \tan^{-1}\left(h_m^{\text{uav}}/r_{m,i_m}\right)$, where $r_{m,i_m} \triangleq \sqrt{\left(x_m^{\text{uav}} - x_{i_m}\right)^2 + \left(y_m^{\text{uav}} - y_{i_m}\right)^2}$. In addition, the average path loss expressions for LoS and NLoS connections are defined

as follows [30], [31]:

$$\ell_{m,i_m}^{\mathrm{LoS}} = 10n\log\left(\frac{4\pi f_c d_{m,i_m}}{c}\right) + \xi^{\mathrm{LoS}} \quad \forall m, i_m \tag{1}$$

$$\ell_{m,i_m}^{\mathrm{NLoS}} = 10n\log\left(\frac{4\pi f_c d_{m,i_m}}{c}\right) + \xi^{\mathrm{NLoS}} \quad \forall m, i_m \tag{2}$$

where n denotes the path-loss exponent, ξ^{LoS} and ξ^{NLoS} represent the average loss in addition to the free-space propagation loss depending on the environment, f_c is the carrier frequency, and c is the light speed. The average path loss as a function of the UAVs' altitudes and coverage's radius is given by [32]

$$\ell_{m,i_m} = P_{m,i_m}^{\mathrm{LoS}}\ell_{m,i_m}^{\mathrm{LoS}} + P_{m,i_m}^{\mathrm{NLoS}}\ell_{m,i_m}^{\mathrm{NLoS}} \tag{3}$$

where $P_{m,i_m}^{\mathrm{NLoS}} = 1 - P_{m,i_m}^{\mathrm{LoS}}$ is the NLoS probability. To this end, a closed-form expression for the average path loss is obtained by substituting (1) and (2) into (3) as follows:

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

In this work, the channel fading coefficient between ISD i_m and UAV m is considered as a complex Gaussian random variable $\tilde{h}_{m,i_m}$ with $\mathbb{E}\left[\left|\tilde{h}_{m,i_m}\right|^2\right] = 1$. The fading and path-loss effect can be shown by $(\tilde{h}_{m,i_m})/(\sqrt{\ell_{m,i_m}})$, which by taking the expected value of this equation, the average channel gain is computed as

$$h_{m,i_m} = \mathbb{E}\left[\left|\frac{\tilde{h}_{m,i_m}}{\sqrt{\ell_{m,i_m}}}\right|^2\right] = \frac{\mathbb{E}\left[\left|\tilde{h}_{m,i_m}\right|^2\right]}{\ell_{m,i_m}} = \ell_{m,i_m}^{-1}. \tag{5}$$

UAV-based communications are encountered with limited energy storage for flight control and data sensing/transmission. The finite energy storage of UAVs restricts their flight and hovering times. In this article, the total energy consumption consists of the following items: 1) energy consumption for task execution at UAVs and cloud server; 2) energy cost for offloading the tasks not only from ISDs to its corresponding UAV but also from UAV to cloud server; and 3) the energy that UAV consumes to hover in the specified place. The aforementioned energies are calculated as a function of the computational and radio resources in the uplink transmission, UAV-aided edge processing, and the energy consumption for hovering.

From the latency points of view, the tolerable delay block for each user, denoted by T_m, consists of three parts: 1) delay for offloading ISDs' tasks to UAVs, represented by T_m^{ul}; 2) time delay for executing offloaded tasks by UAVs, notated by T_m^{ser}; and 3) the spent time for transferring the results back to ISDs, denoted by T_m^{dl}. Indeed, in the first stage with a duration of T_m^{ul}, all ISDs send their tasks, simultaneously. After receiving these data, in the second step with a duration of T_m^{ser}, the UAV/cloud executes offloaded tasks remotely on behalf of these ISDs. Eventually, in the third stage with a duration of T_m^{dl}, the server sends the results back to ISDs. Because of the small sizes of computation results, the time spent for downloading these results from the UAV is negligible compared to the time of device offloading and server computing (i.e., $T_m^{\mathrm{dl}} \approx 0$). Accordingly, we can define $T_m \approx T_m^{\mathrm{ul}} + T_m^{\mathrm{ser}}$.

Uplink Transmission Model: We denote W and $p_{i_m}^{\mathrm{ul}} \in [0, p_{i_m}^{\max}]$ as the total available bandwidth in the network and the transmit power assigned to user i_m for sending messages to its corresponding UAV, respectively. All ISDs in cluster m utilize a superposition coding to transmit the associated tasks to UAV m over the same frequency band. Therefore, the transmitted signal by ISD i_m can be expressed as $x_{i_m} = \sqrt{p_{i_m}^{\mathrm{ul}}}s_{i_m}\ \forall\ i_m \in \mathcal{I}$, where s_{i_m} is the signal message satisfying $\mathbb{E}\left[\left|s_{i_m}\right|^2\right] = 1$.

In addition, the received signal at the mth UAV is obtained as

$$\begin{aligned} r_m^{\mathrm{uav}} &= \sum_{i=1}^{N_m} \frac{\tilde{h}_{m,i_m}}{\sqrt{\ell_{m,i_m}}}x_{i_m} + n_{i_m} \\ &= \sum_{i=1}^{N_m} \sqrt{\frac{p_{i_m}^{\mathrm{ul}}}{\ell_{m,i_m}}}\tilde{h}_{m,i_m}s_{i_m} + n_{i_m} \quad \forall\ m \in \mathcal{M} \end{aligned} \tag{6}$$

where n_{i_m} represents the additive white Gaussian noise at mth UAV. Let $w_m > 0$ denote the fraction of uplink channel bandwidth assigned to cluster m, so that $\sum_{m\in\mathcal{M}} w_m \le 1$. Hence, the noise power in cluster m is defined as $\mathbb{E}\left[\left|n_{i_m}\right|^2\right] \triangleq w_m W N_0$, where N_0 represents the total power spectral density of the noise. In NOMA-based networks, the SIC is performed at the receiver side to mitigate the interference effect imposed by other ISDs, which send their data on the same resource. Let $H_{m,i_m} = (h_{m,i_m})(w_m W N_0)$ denote the average channel gain normalized by the noise power (CGNNP) of ISD i_m. Without loss of generality and for ease of the analysis, suppose that the ISDs on cluster m are indexed in descending order as $H_{m,1_m} \ge H_{m,2_m} \ge \cdots \ge H_{m,N_m}\ \forall\ m \in \mathcal{M}$, meaning that ISDs N_m and 1_m have the weakest and the strongest channel conditions in cluster m, respectively. Then, the NOMA assigns higher powers to ISDs with lower CGNNPs, and thus, we come up with $p_{1_m}^{\mathrm{ul}} \le p_{2_m}^{\mathrm{ul}} \le \cdots \le p_{N_m}^{\mathrm{ul}}\ \forall\ m \in \mathcal{M}$.

At the receiver side, the SIC scheme is adopted to eliminate the co-channel interference. Specifically, for user i_m, UAV m first decodes the message of user j_m with higher CGNNP from other users with smaller CGNNP values (i.e., $i < j$), and then removes the decoded message from the received signal. Similarly, for user i_m, it treats the message of other ISDs with poorer channel conditions (i.e., $j < i$) as the interference. With the above decoding scheme, the signal-to-interference-plus-noise ratio (SINR) related to ISD i_m using the SIC technique is obtained as follows:

$$\Gamma_{i_m} \triangleq \frac{p_{i_m}^{\mathrm{ul}}H_{m,i_m}}{1 + \sum_{j_m\in\mathcal{C}_m, j>i} p_{j_m}^{\mathrm{ul}}H_{m,j_m}}. \tag{7}$$

According to the Shannon–Hartley formula, the transmission rate of ISD i_m is expressed as [33]

$$\begin{aligned} R_{i_m} &= w_m W\log_2\left(1 + \Gamma_{i_m}\right) \\ &= w_m W\log_2\left(1 + \frac{p_{i_m}^{\mathrm{ul}}H_{m,i_m}}{1 + \sum_{j_m\in\mathcal{C}_m, j>i} p_{j_m}^{\mathrm{ul}}H_{m,j_m}}\right), \text{ (b/s)}. \end{aligned} \tag{8}$$

Subsequently, the required time and the energy consumption for transmitting β_{i_m} data bits by ISD i_m to UAV m, denoted

by $\tau_{i_m}^{\mathrm{ul}}$ and $E_{i_m}^{\mathrm{ul}}$, respectively, are derived as $\tau_{i_m}^{\mathrm{ul}} = (\beta_{i_m})/(R_{i_m})$ and $E_{i_m}^{\mathrm{ul}} = p_{i_m}^{\mathrm{ul}} \tau_{i_m}^{\mathrm{ul}}$.

UAV-Aided Edge Processing: Let $F_m^{\mathrm{uav}} > 0$ denote the capacity of mth UAV in terms of the number of CPU cycles per second; and let f_{i_m} represent the fraction of processing power F_m^{uav} related to user i_m, so that $\sum_{i_m \in \mathcal{C}_m} f_{i_m} \leq 1$. The time delay to execute $v_{i_m}\beta_{i_m}$ CPU cycles for user i_m in UAV m is obtained as $\tau_{i_m}^{\mathrm{uav,exe}}(f_{i_m}) = (v_{i_m}\beta_{i_m})/(f_{i_m}F_m^{\mathrm{uav}})$. In addition, the energy consumption for the computation of the CPU for mth UAV would be as follows:

$$E_{i_m}^{\mathrm{uav,exe}} = \kappa_m \tau_{i_m}^{\mathrm{uav,exe}}(f_{i_m}) f_{i_m}^3 = \frac{\kappa_m v_{i_m}\beta_{i_m} f_{i_m}^2}{F_m^{\mathrm{uav}}}$$

where κ_m is the coefficient depending on the chip architecture of mth UAV. It is seen that computing resources of UAVs may be suitable for executing several ISDs requests simultaneously; however, these resources are insufficient for running many requests and lead to a loss of QoS. Moreover, the central cloud has rich resources for task processing; hence, it is assumed to be always attainable as long as ISDs request the service. The cloud server is used as a backup to offload the requests that cannot be offloaded to the UAVs that experience overloading on their limited computing resources. We denote the delay and the energy consumed per CPU cycle during transmission of those requests to the cloud server as e_0 and τ_0, respectively. Accordingly, the total energy consumption and delay caused by data transmission from mth UAV to the central cloud can be expressed as $\tau_{i_m}^{\mathrm{uav,tr}} = v_{i_m}\beta_{i_m}\tau_0$ and $E_{i_m}^{\mathrm{uav,tr}} = v_{i_m}\beta_{i_m}e_0$, respectively.

Energy Consumption for Hovering: We denote p_m^{hover} as the corresponding power consumed by mth UAV for hovering in the specified location. Therefore, the time delay and the energy consumption needed for hovering mth UAV can be obtained as $\tau_m^{\mathrm{hover}} = T_m$ and $E_m^{\mathrm{uav,hover}} = p_m^{\mathrm{hover}}\tau_m^{\mathrm{hover}} = p_m^{\mathrm{hover}}T_m$.

Central Cloud Processing: We denote $G^{\mathrm{cloud}} > 0$ as the capacity of the central cloud in terms of the number of CPU cycles per second; and let g_{i_m} represent the fraction of the processing power G^{cloud} assigned to ISD i_m, so that $\sum_{i_m \in \mathcal{I}} g_{i_m} \leq 1$. The runtime to remotely execute $v_{i_m}\beta_{i_m}$ CPU cycles for ISD i_m is obtained as $\tau_{i_m}^{\mathrm{cloud,exe}}(g_{i_m}) = (v_{i_m}\beta_{i_m})/(g_{i_m}G_m^{\mathrm{cloud}})$. Since the cloud server is connected to the power source without any energy constraint, the energy consumed for processing tasks in the cloud server is avoided. After processing the tasks, since the cloud server has no information about the IP address of ISDs, it will transmit the results to UAVs, and each UAV sends the results to its corresponding ISDs. It should be noted that the time delay and the consumed energy for each ISD to receive the results are negligible, which is mainly due to the fact that the size of the computation outcome, in general, is much smaller than that of input data for many applications (e.g., face recognition) [34]–[36].

The overall latency and energy consumption experienced by ISD i_m can be expressed as (9) and (10), shown at the bottom of the page, respectively, where ξ_{i_m} denotes a binary variable indicating that whether the ISD's task i_m is executed on UAV m ($\xi_{i_m} = 0$) or on the central cloud ($\xi_{i_m} = 1$). Furthermore, each ISD should pay for the used resource in UAVs or the cloud server. It is considered that the unit cost per CPU cycle for UAV m is denoted by φ_m^{uav} and the one for the cloud server is represented by φ^{cloud}. Generally, $\varphi^{\mathrm{cloud}} > \varphi_m^{\mathrm{uav}}$, since a lot of resources are necessary to maintain different powerful servers. In addition, it encourages the utilization of edge computing. It is also considered that the cost is related to the use of resources, e.g., execution rate. Taking the above considerations into account, the average cost of total ISDs is computed as follows:

$$\varphi \triangleq \frac{1}{N}\left\{ \sum_{m \in \mathcal{M}} \varphi_m^{\mathrm{uav}} \sum_{i_m \in \mathcal{C}_m} (1 - \xi_{i_m}) v_{i_m} + \varphi^{\mathrm{cloud}} \sum_{i_m \in \mathcal{C}_m} \xi_{i_m} v_{i_m} \right\}. \tag{11}$$

To investigate the performance of our proposed UAV-aided MEC system, energy efficiency has been welcomed as an interesting metric. This metric is defined as a ratio of the overall data rate, denoted by R^{tot}, over the total energy consumption, represented by E^{tot}, and is formulated as (12), shown at the bottom of the next page.

III. Problem Formulation

We introduce a joint optimization problem with the maximization of the total computation energy efficiency, defined in (12), along with minimizing the average cost of total users, defined in (11), and the number of UAVs (i.e., $|\mathcal{M}|$), while ensuring some realistic constraints. To this end, by considering the aforementioned analytic results in Section II, the

$$\begin{aligned} \tau_{i_m} &\triangleq \tau_{i_m}^{\mathrm{ul}} + (1 - \xi_{i_m})\tau_{i_m}^{\mathrm{uav,exe}}(f_{i_m}) + \xi_{i_m}\left(\tau_{i_m}^{\mathrm{uav,tr}} + \tau_{i_m}^{\mathrm{cloud,exe}}(g_{i_m})\right) \\ &= \frac{\beta_{i_m}}{w_m W \log_2\left(1 + \frac{p_{i_m}^{\mathrm{ul}} H_{m,i_m}}{1 + \sum_{j_m \in \mathcal{C}_m, j > i} p_{j_m}^{\mathrm{ul}} H_{m,j_m}}\right)} + (1 - \xi_{i_m})\frac{v_{i_m}\beta_{i_m}}{f_{i_m}F_m^{\mathrm{uav}}} + \xi_{i_m}\left(v_{i_m}\beta_{i_m}\tau_0 + \frac{v_{i_m}\beta_{i_m}}{g_{i_m}G_m^{\mathrm{cloud}}}\right) \end{aligned} \tag{9}$$

$$\begin{aligned} E_{i_m} &\triangleq E_{i_m}^{\mathrm{ul}} + (1 - \xi_{i_m})E_{i_m}^{\mathrm{uav,exe}} + \xi_{i_m}E_{i_m}^{\mathrm{uav,tr}} + E_{i_m}^{\mathrm{uav,hover}} \\ &= \frac{p_{i_m}^{\mathrm{ul}}\beta_{i_m}}{w_m W \log_2\left(1 + \frac{p_{i_m}^{\mathrm{ul}} H_{m,i_m}}{1 + \sum_{j_m \in \mathcal{C}_m, j > i} p_{j_m}^{\mathrm{ul}} H_{m,j_m}}\right)} + (1 - \xi_{i_m})\frac{\kappa_m v_{i_m}\beta_{i_m} f_{i_m}^2}{F_m^{\mathrm{uav}}} + \xi_{i_m}v_{i_m}\beta_{i_m}e_0 + p_m^{\mathrm{hover}}\tau_m^{\mathrm{hover}} \end{aligned} \tag{10}$$

multiobjective optimization problem is expressed mathematically as the following problem $\mathcal{P}\mathbf{1}$:

$$
\begin{aligned}
\mathcal{P}\mathbf{1}) \quad & \max_{\mathcal{S}} \quad \left\{ \eta_{ee}, \frac{1}{\varphi}, \frac{1}{|\mathcal{M}|} \right\} \\
\text{s.t. } \mathbf{C1}: \quad & \xi_{i_m} \in \{0,1\} \quad \forall i_m \in \mathcal{I} \\
\mathbf{C2}: \quad & 0 \le p_{i_m}^{\text{ul}} \le p_{i_m}^{\max} \quad \forall i_m \in \mathcal{I} \\
\mathbf{C3}: \quad & R_{i_m} \ge R_{i_m}^{\min} \quad \forall i_m \in \mathcal{I} \\
\mathbf{C4}: \quad & \tau_{i_m}^{\text{ul}} \le T_m^{\text{ul}} \quad \forall i_m \in \mathcal{I} \\
\mathbf{C5}: \quad & \xi_{i_m}\left(\tau_{i_m}^{\text{uav,tr}} + \tau_{i_m}^{\text{cloud,exe}}(g_{i_m})\right) \\
& +\left(1-\xi_{i_m}\right)\tau_{i_m}^{\text{uav,exe}}(f_{i_m}) \le T_m^{\text{ser}} \quad \forall i_m \in \mathcal{I} \\
\mathbf{C6}: \quad & E_{i_m} = \sum_{i_m \in \mathcal{C}_m} \Big\{ \left(1-\xi_{i_m}\right)E_{i_m}^{\text{uav,exe}} + \xi_{i_m} E_{i_m}^{\text{uav,tr}} \\
& \qquad + E_{i_m}^{\text{uav,hover}} \Big\} \le E_m^{\max} \quad \forall m \in \mathcal{M} \\
\mathbf{C7}: \quad & f_{i_m} > 0, g_{i_m} > 0 \quad \forall i_m \in \mathcal{I} \\
\mathbf{C8}: \quad & \sum_{i_m \in \mathcal{C}_m} \left(1-\xi_{i_m}\right) f_{i_m} \le 1 \quad \forall m \in \mathcal{M} \\
\mathbf{C9}: \quad & \sum_{i_m \in \mathcal{I}} \xi_{i_m} g_{i_m} \le 1 \quad \forall m \in \mathcal{M} \\
\mathbf{C10}: \quad & \sum_{m,n \in \mathcal{M}, m \ne n} ||\mathbf{p}_m^{\text{uav}} - \mathbf{p}_n^{\text{uav}}||^2 \ge \sigma^2 \\
\mathbf{C11}: \quad & w_m > 0 \quad \forall m \in \mathcal{M} \\
\mathbf{C12}: \quad & \sum_{m \in \mathcal{M}} w_m \le 1 \\
\mathbf{C13}: \quad & x_m^{\min} \le x_m^{\text{uav}} \le x_m^{\max} \quad \forall m \in \mathcal{M} \\
\mathbf{C14}: \quad & y_m^{\min} \le y_m^{\text{uav}} \le y_m^{\max} \quad \forall m \in \mathcal{M} \\
\mathbf{C15}: \quad & h_m^{\min} \le h_m^{\text{uav}} \le h_m^{\max} \quad \forall m \in \mathcal{M}.
\end{aligned}
$$

For simplicity of mathematical expressions, $\mathcal{S}$ is defined as the set of all optimization variables, i.e., $\mathcal{S} \triangleq \left(\mathbf{p}^{\text{ul}}, \mathbf{f}, \mathbf{g}, \boldsymbol{\xi}, \mathbf{w}, \mathbf{x}^{\text{uav}}, \mathbf{y}^{\text{uav}}, \mathbf{h}^{\text{uav}}\right)$, so that $\mathbf{p}^{\text{ul}} \triangleq \left(p_{i_m}^{\text{ul}}\right)_{i_m \in \mathcal{I}}$, $\mathbf{f} \triangleq \left(f_{i_m}\right)_{i_m \in \mathcal{I}}$, $\mathbf{g} \triangleq \left(g_{i_m}\right)_{i_m \in \mathcal{I}}$, $\boldsymbol{\xi} \triangleq \left(\xi_{i_m}\right)_{i_m \in \mathcal{I}}$, $\mathbf{w} \triangleq (w_m)_{m \in \mathcal{M}}$, $\mathbf{x}^{\text{uav}} \triangleq \left(x_m^{\text{uav}}\right)_{m \in \mathcal{M}}$, $\mathbf{y}^{\text{uav}} \triangleq \left(y_m^{\text{uav}}\right)_{m \in \mathcal{M}}$, and $\mathbf{h}^{\text{uav}} \triangleq \left(h_m^{\text{uav}}\right)_{m \in \mathcal{M}}$. For the optimization problem $\mathcal{P}\mathbf{1}$, constraint **C1** indicates the offloading decision between UAV and cloud server represented by a binary variable, **C2** represents the power constraint restricted to the minimum and maximum power levels in each user. Constraint **C3** ensures that the QoS requirement is met for each user where $R_{i_m}^{\min}$ denotes the target uplink transmission rate of user i_m. Constraints **C4** and **C5** represent the latency limits for the uplink direction and UAV/cloud server-side, respectively, which should be less than or equal to the maximum acceptable delay. Note that constraint **C4** comes from $\tau_{i_m}^{\text{ul}} = (\beta_{i_m})/(R_{i_m})$ and constraint **C3**. In addition, **C6** imposes the UAV energy budget constraint related to edge computing, data transmission, and hovering. In addition, **C7**–**C9** impose the mentioned restrictions on the UAV and cloud computational resources. By recalling mth UAV position definition as $\mathbf{p}_m^{\text{uav}}$ in Section II, constraint **C10** must be satisfied to avoid the collision between UAVs. **C11** and **C12** indicate the restrictions related to the network bandwidth, respectively.

It is clear that $\mathcal{P}\mathbf{1}$ is a multiobjective nonlinear optimization problem with several constraints. Some traditional optimization and intelligent optimization algorithms are widely used to solve such multiobjective optimization problems, including weighted, constrained, and linear programming methods [37]. In weighted methods, the first step is performed to obtain a dimensionless objective function. Toward this goal, since the objective function in problem $\mathcal{P}\mathbf{1}$ includes three parameters with different dimensionalities, we should divide these functions into their nominal values. Therefore, it is assumed that users in the proposed network have nominal values for total computation energy efficiency and the average cost of total users, denoted by constant parameters $(\tilde{\eta}, \tilde{\varphi})$, respectively. In addition, a scalarization or weighting method can be applied by incorporating a set of weight factors $\left\{\omega^{ee}, \omega^{\varphi}, \omega^{M}\right\}$, under condition $\omega^{ee} + \omega^{\varphi} + \omega^{M} = 1$, to reflect the tradeoff, or equivalently, relative importance of the energy efficiency, average cost of total users, and the total number of UAVs, respectively. Consequently, problem $\mathcal{P}\mathbf{1}$ is transformed to a single objective optimization problem $\mathcal{P}\mathbf{2}$ defined as

$$
\begin{aligned}
\mathcal{P}\mathbf{2}) \quad & \max_{\mathcal{S}} \ \Phi \triangleq \left\{ \frac{\omega^{ee}}{\tilde{\eta}} \eta_{ee} - \frac{\omega^{\varphi}}{\tilde{\varphi}} \varphi - \frac{\omega^{M}}{M} |\mathcal{M}| \right\} \\
& \text{s.t.} \quad \mathbf{C1} \sim \mathbf{C15}.
\end{aligned}
$$

As seen from the nonconvexity of the objective function and constraints, problem $\mathcal{P}\mathbf{2}$ is nonconvex and it belongs to a mixed-integer nonlinear problem (MINP). Besides, the coupling of some variables makes the optimization problem $\mathcal{P}\mathbf{2}$ even more complicated. Hence, finding the optimal solution for $\mathcal{P}\mathbf{2}$ is intractable. To proceed to an optimal solution, the binary variable ξ_{i_m} is relaxed to be determined in the interval [0, 1]. This relaxation leads to: 1) a lower bound on the minimum energy consumption that can be achieved by a hard selection between cloud offloading and UAV and 2) the computational task of user i_m can be divided into two distinct parts, i.e., $1-\xi_{i_m}$ and ξ_{i_m} percents, which can be executed separately on

$$
\begin{aligned}
\eta_{ee} &\triangleq \frac{R^{\text{tot}}}{E^{\text{tot}}} = \frac{\sum_{i_m \in \mathcal{I}} R_{i_m}}{\sum_{i_m \in \mathcal{I}} E_{i_m}} \\
&= \frac{\sum_{m \in \mathcal{M}} \sum_{i_m \in \mathcal{C}_m} w_m W \log_2\left(1 + \frac{p_{i_m}^{\text{ul}} H_{m,i_m}}{1+\sum_{j_m \in \mathcal{C}_m, j>i} p_{j_m}^{\text{ul}} H_{m,j_m}}\right)}{\sum_{m \in \mathcal{M}} \sum_{i_m \in \mathcal{C}_m} \frac{p_{i_m}^{\text{ul}} \beta_{i_m}}{w_m W \log_2\left(1+\frac{p_{i_m}^{\text{ul}} H_{m,i_m}}{1+\sum_{j_m \in \mathcal{C}_m, j>i} p_{j_m}^{ul} H_{m,j_m}}\right)}} + \left(1-\xi_{i_m}\right)\frac{\kappa_m v_{i_m} \beta_{i_m} f_{i_m}^2}{F_m^{\text{uav}}} + \xi_{i_m} \beta_{i_m} e_0 + p_m^{\text{hover}} \tau_m^{\text{hover}}}
\end{aligned} \tag{12}
$$

mth UAV and cloud server, respectively. In the next section, we address this issue and propose an iterative search algorithm to solve problem $\mathcal{P}\mathbf{2}$, wherein parameter ξ_{i_m} is relaxed.

IV. Proposed Solution for Problem $\mathcal{P}\mathbf{2}$

So far, we mathematically formulated our problem as a multiobjective optimization problem $\mathcal{P}\mathbf{1}$. We use the scalarization method to transform such a multiobjective optimization problem into a single objective problem $\mathcal{P}\mathbf{2}$. However, this problem is intractable to solve, due to the nonconvexity of the objective function and constraints. On the other hand, the coupling of some variables makes this problem more complicated. Hence, in order to more tractable and further reduce the time and computational complexity, in this section, we decouple the problem $\mathcal{P}\mathbf{2}$ into three optimal subproblems and solve each one through the proposed multivariable fixed iterative algorithm. In general, the proposed algorithm for solving $\mathcal{P}\mathbf{2}$ is divided into four steps as follows: 1) in the first step and with the aim of reducing the computational complexity of the algorithm, we present an initial estimation for the number of UAVs (i.e., $\hat{M}$); 2) in the second step, we propose the modified GKM algorithm to efficiently perform the users' clustering process. The output of this part determines the 2-D UAV's position [namely, $(\mathbf{x}^{\text{uav}}, \mathbf{y}^{\text{uav}})$] along with the users' clustering according to their horizontal coordination; 3) then, by any given $\boldsymbol{\mathcal{S}} \backslash \mathbf{h}^{\text{uav}}$ (i.e., the set $\boldsymbol{\mathcal{S}}$ except vector $\mathbf{h}^{\text{uav}}$), we find the optimal UAV's height by leveraging the SLP method to solve the subproblem $\mathcal{P}\mathbf{3}$; and 4) finally, given the optimal values of $\mathbf{x}^{\text{uav}}$, $\mathbf{y}^{\text{uav}}$, and $\mathbf{h}^{\text{uav}}$, we again employ the SLP method to obtain the optimal values of other parameters (i.e., $\boldsymbol{\mathcal{S}} \setminus \{\mathbf{x}^{\text{uav}}, \mathbf{y}^{\text{uav}}, \mathbf{h}^{\text{uav}}\}$) by solving the subproblem $\mathcal{P}\mathbf{4}$.

The above procedure is performed iteratively until the algorithm converges to the near-optimal solution. It is worth mentioning that the proposed algorithm eliminates the UAV that does not impact the QoS of the network during the process. Therefore, the optimum number of UAVs is determined.

Initial Estimation of M: We first provide an initial estimation of the number of UAVs that serve all users in the network, satisfying the requirements of the achievable rate and the capacity of each UAV. This stage is helpful for reducing the complexity of the proposed algorithm, in particular, when we encounter a dynamic network's topology with users. This procedure outperforms the scheme in [30], where it has been proven that a very large geographical area can be covered by only one UAV by adjusting its altitude in the optimum height for coverage constraint, which is an infeasible method in existing ultradense networks due to the high data traffic requirement of users. The initial estimation method is performed in two steps. Denoting ψ as the capacity constraint of UAV, in the first step, we find the maximum number of users that a UAV can serve them, denoted by N^{uav} calculated as $N^{\text{uav}} = \lfloor \psi/(R_{i_m}^{\min}) \rfloor$. In the second step, the number of UAVs is estimated by $\hat{M} = \min\{\tilde{M}, M\}$, where $\tilde{M} = \lceil N/(N^{\text{uav}}) \rceil$.

Users Clustering: In the second stage, we classify all users to M groups according to their horizontal coordinations by proposing a modified version of one unsupervised machine learning-based method, namely, the GKM algorithm [38]. Similar to the conventional GKM, the proposed MGKM scheme establishes a deterministic global optimization procedure that is independent of initial parameter values and utilizes the K-means algorithm as a local method. In addition, it processes in an incremental manner intending to optimally add one new cluster center at each stage. In contrast to the classical GKM algorithm, an outstanding feature of the proposed MGKM method is that it has a rather lower computational complexity, which causes it to be applicable for clustering in ultradense wireless networks. Through the offline design of the MGKM algorithm and relying on this clustering result, the corresponding UAVs' positions would be fixed.

We summarize the offline design steps of the proposed MGKM algorithm as the pseudocode in Algorithm 1. We will evaluate numerically the optimality and the convergence of this

Algorithm 1 Proposed MGKM Algorithm for Users Clustering

Input: The position of users and number of clusters $\hat{M}$.

for $m = 1, \ldots, \hat{M}$ **do**
if $m == 1$ **do**
Calculate c_1 as the centroid of the set $\mathcal{I}_0$.
else
for $k = 1, \ldots, N$ **do**
Set $c_m \leftarrow u_k$.
Employ K-means clustering algorithm for cluster centers $(c_1, \ldots, c_m)$.
$\Theta_m(c_1, \ldots, c_m) = \frac{1}{N} \sum_{i=1}^{N} \min_{j=1,\ldots,m} \left\| c_j - u_i \right\|^2$.
Set $c_m \leftarrow \arg\min_m \ \Theta_m(c_1, \ldots, c_m)$.
end for
end if
end for
for each $i \in \mathcal{I}_0$ **do**
for each $j \in \mathcal{I}_0$ && $j \neq i$ **do**
Calculate the Euclidean distance r_{ij} according to (13).
end for
end for
for each $i \in \mathcal{I}_0$ **do**
Select the point that minimize ζ_i as the first center according to (14).
end for
Set $q \leftarrow 1$.
while $q \leqslant \hat{M}$ **do**
Employ K-means clustering algorithm to obtain the best q cluster and their cluster centers $\left(c_1, c_2, \ldots, c_q\right)$.
Set $q \leftarrow q + 1$.
for each $i \in \mathcal{I}_0$ **do**
Calculate Ξ_i according to (15).
end for
Calculate the index i that has the minimum value Ξ_i as the center of the new cluster according to (16).
set $\left(c_1, \ldots, c_q, u_i\right)$ as the initial centers for next iteration.
end while

Output: $\mathcal{I}, \left(c_1, \ldots, c_{\hat{M}}\right), \left(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{\hat{M}}\right)$.

algorithm in Section VI. To get more insight into how this algorithm works, the proposed MGKM algorithm is divided into the following four steps.

1) *Step 1 (Initialization):* Let us define the initial set of the total users in the network (i.e., before the conventional indexing in Section II) as $\mathcal{I}_0 = \{u_1, \ldots, u_N\}$. In the initialization step, we need to determine the position of the network's users and the number of clusters as the inputs of the MGKM algorithm.
2) *Step 2 [Solving ($\hat{M}-1$)_Clustering Problem]:* The proposed MGKM scheme is initialized with one cluster, i.e., $\hat{M} = 1$, and obtains its optimal position corresponding to the centroid of the set of users (i.e., $\mathcal{I}_0$) followed by calculation of the two-clusters problem, i.e., $\hat{M} = 2$. This algorithm employs an objective function defined as the mean square error of the Euclidean distance metric, i.e., $\Theta_m(c_1, \ldots, c_m) = 1/N \sum_{i=1}^{N} \min_{j=1,\ldots,m} \left\|c_j - u_i\right\|^2$, where m denotes the number of clusters, and c_j, $j = 1, \ldots, m$, represents the centroid of jth cluster. The minimum value of the above objective function is chosen as the cluster center (see line 9 of Algorithm 1). For the problem with $\hat{M} = 1$, the first cluster center is always located at the optimal position, while the second one is placed at the position of each user $u_i, i = 1, \ldots, N$. The K-means algorithm should be executed for each combination of the initial points. Eventually, the combination of the initial points is selected that achieves the best clustering result as the solution for the clustering problem with $\hat{M} = 2$. We denote $\left(c_1, \ldots, c_{\hat{M}-1}\right)$ as the optimal solution of $(\hat{M}-1)$_clustering problem. The above procedure has been summarized in lines 1–12 of Algorithm 1.
3) *Step 3 (Solving $\hat{M}$_Clustering Problem):* After finding the solution for the $(\hat{M}-1)$_clustering problem, the MGKM algorithm is run to attain the solution of the $\hat{M}$_clustering problem with the initial state $\left(c_1, \ldots, c_{\hat{M}-1}\right)$. Recalling that the computational complexity reduction in the conventional GKM algorithm is one of the purposes of this work while minimizing the clustering error, the main contribution of this step is to choose the optimal initial center for the new cluster at each step. Toward this goal, let us denote the Euclidean distance between user u_i and u_j as r_{ij} given by (see lines 13–17 of Algorithm 1)

$$r_{ij} = \sqrt{\left(x_{u_i} - x_{u_j}\right)^2 + \left(y_{u_i} - y_{u_j}\right)^2}, \quad u_i, u_j \in \mathcal{I}_0. \tag{13}$$

To calculate an initial center, parameter ζ_i is defined for each user u_i as follows:

$$\zeta_i = \sum_{j=1}^{N} \frac{r_{ij}}{\sum_{l=1}^{N} r_{jl}}, \ i = 1, \ldots, N. \tag{14}$$

Clearly, the best center of a cluster is the sample with the minimum ζ_i, i.e., the one with a relatively high density around that point. Suppose that the solution of the $(\hat{M}-1)$_clustering problem is $\left(c_1, \ldots, c_{\hat{M}-1}\right)$. A new cluster center, i.e., $\hat{M}$th initial center, is added at the location of u_i so that it minimizes the following function Ξ_i:

$$\Xi_i = \frac{\zeta_i}{\sum_{j=1}^{m-1} r(u_i, c_j)}, \ i = 1, \ldots, N \tag{15}$$

where $r(u_i, c_j)$ represents the Euclidean distance between u_i and c_j. Accordingly, we can obtain the centroid of $\hat{M}$th cluster by setting $c_{\hat{M}} = u_i$ so that

$$i = \arg\min_{i} \ \Xi_i \ , \qquad i = 1, \ldots, N. \tag{16}$$

Given the cluster centers $\left(c_1, \ldots, c_{\hat{M}}\right)$, the K-means clustering scheme is implemented to classify users between clusters. It is worth mentioning that the denominator of (15) ensures that the center of the new cluster would be far away from the existing cluster centers (see lines 18–31 of Algorithm 1).
4) *Step 4 (Output):* The outputs of executing the above MGKM algorithm would be as follows: a) the total users are clustered and are new indexed by the set $\mathcal{I} \triangleq \{i_m : i = 1, \ldots, N_m, m = 1, \ldots, \hat{M}\}$ as defined in Section II; b) the center of each cluster, i.e., $(c_1, \ldots, c_{\hat{M}})$, which determines the UAV's horizontal location; and c) each cluster and its corresponding users are formed, i.e., $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{\hat{M}}$.

Flight Heights of UAVs: In the third stage, we aim to optimize the flight heights of UAVs for any given $\boldsymbol{\mathcal{S}} \backslash \mathbf{h}^{\text{uav}}$ (i.e., the set $\boldsymbol{\mathcal{S}}$ except vector $\mathbf{h}^{\text{uav}}$), based on the following optimization problem:

$$\begin{aligned}
\boldsymbol{\mathcal{P}3)} \ & \max_{\mathbf{h}^{\text{uav}}} \quad \eta_{ee} \\
\text{s.t.} \ & \mathbf{C1:} \quad h_m^{\min} \le h_m^{\text{uav}} \le h_m^{\max} \quad \forall m \in \mathcal{M} \\
& \mathbf{C2:} \quad R_{i_m} \ge R_{i_m}^{\min} \quad \forall i_m \in \mathcal{I} \\
& \mathbf{C3:} \quad \tau_{i_m}^{\text{ul}} \le T_m^{\text{ul}} \quad \forall i_m \in \mathcal{I} \\
& \mathbf{C4:} \quad \sum_{m,n \in \mathcal{M}, m \ne n} ||\mathbf{p}_m^{\text{uav}} - \mathbf{p}_n^{\text{uav}}||^2 \ge \sigma^2 \ .
\end{aligned}$$

Problem $\boldsymbol{\mathcal{P}3}$ is a nonconvex optimization problem in terms of $\mathbf{h}^{\text{uav}}$ due to the nonconvexity of the objective function and the constraints **C2**–**C4**. In the following, we overcome this problem by leveraging the SLP method.

Task Offloading Scheme Optimization: In the last step, we derive solutions to the joint optimization problem in terms of the bandwidth and power allocation, computational resources and task assignment in UAVs, and cloud server. For any given 3-D placement of UAVs (i.e., $\mathbf{x}, \mathbf{y}, \mathbf{h}^{\text{uav}}$), obtained from Algorithms 1 and 2, problem $\boldsymbol{\mathcal{P}2}$ can be transformed as follows:

$$\begin{aligned}
\boldsymbol{\mathcal{P}4)} \ & \min_{\boldsymbol{\mathcal{S}} \backslash \{\mathbf{x}^{\text{uav}}, \mathbf{y}^{\text{uav}}, \mathbf{h}^{\text{uav}}\}} \quad -\frac{\omega^{ee}}{\tilde{\eta}} \eta_{ee} + \frac{\omega^{\varphi}}{\tilde{\varphi}} \varphi \\
\text{s.t.} \ & \mathbf{C1:} \quad \xi_{i_m} \in [0, 1] \quad \forall i_m \in \mathcal{I} \\
& \mathbf{C2:} \quad R_{i_m} \ge R_{i_m}^{\min} \quad \forall i_m \in \mathcal{I} \\
& \mathbf{C3:} \quad \tau_{i_m}^{\text{ul}} \le T_m^{\text{ul}} \quad \forall i_m \in \mathcal{I} \\
& \mathbf{C4:} \quad \xi_{i_m} \left(\tau_{i_m}^{\text{uav,tr}} + \tau_{i_m}^{\text{cloud,exe}}(g_{i_m})\right) \\
& \qquad + (1 - \xi_{i_m}) \tau_{i_m}^{\text{uav,exe}}(f_{i_m}) \le T_m^{ser} \quad \forall i_m \in \mathcal{I} \\
& \mathbf{C5:} \quad 0 \le p_{i_m}^{\text{ul}} \le p_{i_m}^{\max} \quad \forall i_m \in \mathcal{I}
\end{aligned}$$

Algorithm 2 Pseudocode of the SLP Method to Solve Problems $\mathcal{P}3$ and $\mathcal{P}4$

Input: An initial point $\boldsymbol{x}_0$, $0 < \rho_{\text{bad}} < \rho_{\text{good}} < 1$, $\kappa > 1$, $\varepsilon_1 > 0$, $\varepsilon_2 > 0$, $0 < \Delta_{\min} \leq \Delta_0 \leq \Delta_{\max}$, $0 < \lambda_0 < \lambda_{\max}$, $\gamma > 1$.

Set $k \leftarrow 0$ and $stop \leftarrow 0$.
while
 Solve linear problem (21) to find a direction $\boldsymbol{d}_k^*$.
if $||\boldsymbol{d}_k^*|| < \varepsilon_1$
 $stop = 1$, the optimal solution is obtained.
else
 Compute ρ by (24).
 Check feasibility of $\boldsymbol{x}_k + \boldsymbol{d}_k^*$ by (22).
endif
 Swich case
 case: $\boldsymbol{x}_k + \boldsymbol{d}_k^*$ is feasible and $\rho > \rho_{\text{bad}}$
 $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \boldsymbol{d}_k^*$, $\lambda_{k+1} \leftarrow \lambda_k$.
if $\rho > \rho_{\text{good}}$
 $\Delta_{k+1} \leftarrow \min\{\gamma \Delta_k, \Delta_{\max}\}$
else
 $\Delta_{k+1} \leftarrow \Delta_k$
endif
 case: $\boldsymbol{x}_k + \boldsymbol{d}_k^*$ is feasible and $\rho < \rho_{\text{bad}}$
 $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k$, $\lambda_{k+1} \leftarrow \lambda_k$.
 $\Delta_{k+1} \leftarrow \max\{\frac{1}{\gamma}\Delta_k, \Delta_{\min}\}$
 case: $\boldsymbol{x}_k + \boldsymbol{d}_k^*$ is infeasible
 if $\Delta_k = \Delta_{\min}$ and $\lambda_k = \lambda_{\max}$
 $stop \leftarrow 1$.
 The problem is infeasible.
else
 $\Delta_{k+1} \leftarrow \max\{\frac{1}{\gamma}\Delta_k, \Delta_{\min}\}$
 $\lambda_{k+1} \leftarrow \min\{\kappa\lambda_k, \lambda_{\max}\}$
endif
 end Swich case
 $k \leftarrow k + 1$
end while

Output: The optimal solution $\boldsymbol{x}^*$.

$$\textbf{C6:}\quad E_{i_m} = \sum_{i_m \in \mathcal{C}_m} \Big\{ (1-\xi_{i_m}) E_{i_m}^{\text{uav,exe}} + \xi_{i_m} E_{i_m}^{\text{uav,tr}} + E_{i_m}^{\text{uav,hover}} \Big\} \leq E_m^{\max} \quad \forall m \in \mathcal{M}$$

$$\textbf{C7:}\quad f_{i_m} > 0, g_{i_m} > 0 \quad \forall i_m \in \mathcal{I}$$

$$\textbf{C8:}\quad \sum_{i_m \in \mathcal{C}_m} (1-\xi_{i_m}) f_{i_m} \leq 1 \quad \forall m \in \mathcal{M}$$

$$\textbf{C9:}\quad \sum_{i_m \in \mathcal{I}} \xi_{i_m} g_{i_m} \leq 1 \quad \forall m \in \mathcal{M}$$

$$\textbf{C10:}\quad w_m > 0 \quad \forall m \in \mathcal{M}$$

$$\textbf{C11:}\quad \sum_{m \in \mathcal{M}} w_m \leq 1\,.$$

V. Proposed SLP-Based Algorithm

In this section, we propose an efficient algorithm to solve the optimization problems $\mathcal{P}3$ and $\mathcal{P}4$. Since these problems are strongly nonlinear, we take the advantage of SLP. The SLP is one of the most successful methods for nonlinear problems, where the main goal is to find a local minimum by several consecutive linear programs obtained by the first-order Taylor series. In addition, this method is powered by the trust-region method to ensure convergence.

For the ease of our analysis in finding the optimum solution, let us start with the following nonlinear constraint problem:

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ & \boldsymbol{G}(\boldsymbol{x}) \preceq \boldsymbol{0} \\ & \boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{0} \\ & \boldsymbol{x}_{\min} \preceq \boldsymbol{x} \preceq \boldsymbol{x}_{\max} \end{aligned} \tag{17}$$

where $\boldsymbol{G}(\boldsymbol{x}) = [g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})]^{\mathrm{T}}$, $g_i(\boldsymbol{x}){:}\mathbb{R}^n \rightarrow \mathbb{R}$, and $\boldsymbol{H}(\boldsymbol{x}) = [h_1(\boldsymbol{x}), \ldots, h_p(\boldsymbol{x})]^{\mathrm{T}}$, $h_j(\boldsymbol{x}){:}\mathbb{R}^n \rightarrow \mathbb{R}$. Observe that any nonlinear problem, including $\mathcal{P}3$ and $\mathcal{P}4$, can be presented as this form.

The algorithm is started with the initial point $\boldsymbol{x}_0$ related to the set of all optimization variables except $\mathbf{x}^{\text{uav}}$, $\mathbf{y}^{\text{uav}}$, and $\mathbf{h}^{\text{uav}}$ in $\mathcal{P}4$. Then, a sequence $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ is calculated, which converges to a local optimum solution. Approximating the all functions $f(\boldsymbol{x})$, $g_i(\boldsymbol{x})$, and $h_j(\boldsymbol{x})$, by the first-order Taylor series in the kth iteration, problem (17) is changed to the following problem:

$$\begin{aligned} \min_{d} \; & f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{d} \\ & \boldsymbol{G}(\boldsymbol{x}_k) + \nabla \boldsymbol{G}(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{d} \preceq \boldsymbol{0} \\ & \boldsymbol{H}(\boldsymbol{x}_k) + \nabla \boldsymbol{H}(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{d} = \boldsymbol{0} \\ & \boldsymbol{x}_{\min} \preceq \boldsymbol{x} \preceq \boldsymbol{x}_{\max}. \end{aligned} \tag{18}$$

After solving such a problem, the new point should be calculated by $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}^*$ where d^* is the optimal solution of problem (18). However, this approximated model has some drawbacks. First, as the Taylor approximation is valid only near the point $\boldsymbol{x}_k$, the solution may be ineligible where $\boldsymbol{d}$ is improper, so the convergence may not occur. To overcome this fault, the trust region is implemented, i.e., an amount Δ_k is chosen and the inequality $||\boldsymbol{d}|| \leq \Delta_k$ is added to (18). Any norm can be used to measure $\boldsymbol{d}$. Here, we use ℓ_∞, because it has linear property. Parameter Δ_k is named the radius of trust-region and can be a constant (not suggested), or varies in each iteration. As we will see later, in the proposed algorithm, this parameter may change in each iteration.

The second fault is that the problem (18) may be infeasible for some k during the execution of the algorithm, which disrupts the algorithm process. To tackle this problem, we deploy the penalty method to converge to the optimal point without the need for feasibility of the original problems (17) or (18). In this method, the constraints are added to the objective function by some multipliers. Among various ways to create a penalty function, here, we use the following form:

$$\begin{aligned} \min_{\boldsymbol{d}} \Phi(\boldsymbol{x}) = & f(\boldsymbol{x}) + \lambda \sum_{i=1}^{m} \max(0, g_i(\boldsymbol{x})) + \lambda \sum_{j=1}^{p} |h_j(\boldsymbol{x})| \\ & \boldsymbol{x}_{\min} \preceq \boldsymbol{x} \preceq \boldsymbol{x}_{\max} \end{aligned} \tag{19}$$

where λ denotes the penalty coefficient. The main concern of this method is to determine the best value of parameter λ since the problem (19) provides a local minimum for problem (18)

only for sufficiently large λ, otherwise, the solution may be infeasible. The following theorem establishes the correlation between problems (17) and (19), and the importance of the parameter λ.

Theorem 1: Let $\boldsymbol{x}^*$ be a strict local minimum of problem (17) and all functions f, g_i, $i = 1 \dots n$, and h_j, $j = 1 \dots p$, are continuously differentiable. Then, there exists a $\tilde{\lambda} > 0$ so that $\boldsymbol{x}^*$ is a local minimum of problem (19) for every $\lambda \geq \tilde{\lambda}$.

Proof: See [39], [40]. ∎

Hence, we linearize problem (19) instead of problem (17) in $\boldsymbol{x}_k$ as follows:

$$\begin{aligned}\min_{\boldsymbol{d}} \psi(\boldsymbol{d}) = {} & f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{d} \\ & + \lambda \sum_{i=1}^{m} \max(0, g_i(\boldsymbol{x}_k) + \nabla g_i(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{d}) \\ & + \lambda \sum_{j=1}^{p} |h_j(\boldsymbol{x}_k) + \nabla h_j(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{d}| \\ & \boldsymbol{x}_{\min} \preceq \boldsymbol{x}_k + \boldsymbol{d} \preceq \boldsymbol{x}_{\max} \\ & -\Delta_k \leq d_i \leq \Delta_k. \end{aligned} \tag{20}$$

It should be noted that the added last constraint is related to the trust region. Although problem (20) seems a nonsmooth at the first glance, by defining $t_i = \max(0, g_i(\boldsymbol{x}_k) + \nabla g_i(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{d})$, and $h_j(\boldsymbol{x}_k) + \nabla h_j(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{d} = s_j^+ - s_j^-$, transforming the problem into the following linear programming is well known:

$$\begin{aligned}\min_{\boldsymbol{d},\boldsymbol{t},\boldsymbol{s}^+,\boldsymbol{s}^-} \quad & \nabla f(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{d} + \lambda \sum_{i=1}^{m} t_i + \lambda \sum_{j=1}^{p} \left(s_j^+ + s_j^-\right) \\ & \boldsymbol{G}(\boldsymbol{x}_k) + \nabla \boldsymbol{G}(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{d} \preceq \boldsymbol{t} \\ & \boldsymbol{H}(\boldsymbol{x}_k) + \nabla \boldsymbol{H}(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{d} = \boldsymbol{s}^+ - \boldsymbol{s}^- \\ & \boldsymbol{x}_{\min} - \boldsymbol{x}_k \preceq \boldsymbol{d} \preceq \boldsymbol{x}_{\max} - \boldsymbol{x}_k \\ & -\Delta_k \leq d_i \leq \Delta_k \\ & \boldsymbol{t}, \boldsymbol{s}^+, \boldsymbol{s}^- \succeq \boldsymbol{0}. \end{aligned} \tag{21}$$

As mentioned before, the used λ in an iteration may be improper because of infeasibility. Hence, λ must also be updated if it is not enough large. From the theoretical point of view, it is needed $\lambda \geq \max\{\mu_i \ \forall_i\}$, where μ_i is the Lagrange multiplier of constraints [24]. However, in practice, the algorithm may diverge if a very large value of λ is chosen. Hence, we increase the value of λ gradually at each step in case the solution is infeasible. To do this, we choose a certain tolerance ε_2. If the new solution $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d^*}_k$ is feasible, i.e.,

$$\max_i g_i(\boldsymbol{x}_{k+1}) \leq \varepsilon_2, \quad \max_j |h_j(\boldsymbol{x}_{k+1})| \leq \varepsilon_2 \tag{22}$$

then the solution is accepted, otherwise, $\boldsymbol{x}_{k+1}$ is rejected and again the problem (21) should be solved for $\boldsymbol{x}_k$ and the modified λ_k. In this situation, we increase λ_k by the following relation:

$$\lambda_k := \min\{\kappa\lambda_k, \lambda_{\max}\} \tag{23}$$

where $\kappa > 1$ is a coefficient. This guarantees that λ_k is increased and $\lambda_{\max}$ is a predetermined large value. Finally, let us comeback to the radius of trust region. The ratio

$$\rho = \frac{\Phi(\boldsymbol{x}_k + \boldsymbol{d}_k, \lambda_k) - \Phi(\boldsymbol{x}_k, \lambda_k)}{\psi(\boldsymbol{d}_k) - \psi(\boldsymbol{0})} \tag{24}$$

is a criterion for whether the radius of the trust region is sought or not [39]. More precisely, given prespecified values $0 < \rho_{\text{bad}} < \rho_{\text{good}} < 1$, if $\rho < \rho_{\text{bad}}$, then the trust region is improper. Therefore, this step must be repeated again by a new trust region $\Delta_k := 1/\gamma\Delta_k$ for some $\gamma > 1$. If $\rho_{\text{bad}} < \rho < \rho_{\text{good}}$, then this step is accepted and the trust region remains unchanged. Finally, if $\rho > \rho_{\text{good}}$, the trust region can be enlarged by $\Delta_k := \min\{\gamma\Delta_k, \Delta_{\max}\}$, for some predetermined value $\Delta_{\max}$.

Remark 1: Based on the above discussions, acceptance or rejection of a solution $\boldsymbol{x}_k$ depends on two points: 1) feasibility and 2) parameter ρ. Therefore, three cases are seen in the algorithm. In the most desirable case, the problem is feasible in $\boldsymbol{x}_k$ and $\rho > \rho_{\text{bad}}$, and only in this case, the solution is accepted. In the other two cases, the solution is rejected and the parameters λ_k and/or Δ_k should be updated. It is important to note that the infeasibility of the problem may be due to the inadequacy of one of the values λ_k or Δ_k. If the problem remains infeasible for $\lambda_k = \lambda_{\max}$ and $\Delta_k = \Delta_{\min}$ (case 3), we conclude that the problem is infeasible. In this situation, we may start with another initial solution $\boldsymbol{x}_0$. Algorithm 2 summarizes the above discussions as the pseudocode.

A. Convergence and Parameters Adjustment

The convergence of the SLP method has been extensively considered in some papers [41], [42]. Especially, in [43], the convergence of the combination of penalty methods and SLP has been investigated. In addition, there are several parameters that should be adjusted in the proposed Algorithm 2. The parameters ρ_{bad} and ρ_{good} are chosen typically in the intervals $(0, 0.25)$ and $(0.75, 1)$, respectively, and we choose $\gamma = 2$. However, the selection of κ is more difficult and depends on the scale of the problem. If κ is a small number, convergence may be slow. The numerical results show that the best value of κ is chosen in the range $\kappa \in (1.5, 2.1)$. The tolerance numbers are $\varepsilon_1 = 10^{-6}$ and $\varepsilon_2 = 10^{-2}$. It should be noted that to solve any optimization problem, it needs a starting point chosen from the feasible set. In our optimization problem, this point should satisfy the constraints specified in problems $\boldsymbol{\mathcal{P}3}$ and $\boldsymbol{\mathcal{P}4}$ and leads to a better solution. Finding the starting point is possible in different ways, e.g., genetic algorithm.

Eventually, we summarize the proposed procedure for solving problem $\boldsymbol{\mathcal{P}2}$ as pseudocode in Algorithm 3. It should be noted that Φ_υ represents the objective function of problem $\boldsymbol{\mathcal{P}2}$ in the current iteration υ.

B. Complexity Analysis

Proposition 1: The computational complexity of Algorithm 1 is of order $\mathcal{O}\left(tN^2 \sum_{k=1}^{\hat{M}} k\right)$, where t represents the number of iterations needed until convergence, N represents the number of users with d-dimensional vectors, and $\hat{M}$ denotes the number of clusters.

Proof: For Algorithm 1, there are totally four-tier loops, containing three outer "for" loops (i.e., lines 1–12, 13–17, and

18–20), and outer "while" loop (i.e., lines 22–30). It should be noted that the complexity of Algorithm 1 is mainly due to lines 7 and 23, related to determining the cluster center in the first outer "for" loop and "while" loop, respectively, where we employed the *K*-means algorithm. Also, the complexity of other lines is $\mathcal{O}(1)$. To categorize d-dimensional data vectors ϑ into k clusters, the *K*-mean algorithm has a complexity of order $\mathcal{O}(tdk\vartheta)$, wherein t is the number of iterations needed until convergence [44]. Since in this work $d = 2$, the corresponding computational complexity for the first outer "for" loop and "while" loop is $\mathcal{O}(N\sum_{k=1}^{\hat{M}} 2tkN)$ and $\mathcal{O}(\sum_{k=1}^{\hat{M}} 2tkN)$, respectively. Hence, the total running time of the MGKM algorithm is calculated by adding the aforementioned complexity terms, i.e.,

$$\mathcal{O}\left(\sum_{k=1}^{\hat{M}} 2tkN + N\sum_{k=1}^{\hat{M}} 2tkN\right) = \mathcal{O}\left(2tN(N+1)\sum_{k=1}^{\hat{M}} k\right)$$

or equivalently, of order $\mathcal{O}\left(tN^2\sum_{k=1}^{\hat{M}} k\right)$. ■

Proposition 2: The computational complexity of Algorithm 2 is of order $\mathcal{O}(S\tilde{N})$, where S indicates the number of iterations needed to converge to the desired answer and $\tilde{N} = N + M^2$ for problem $\boldsymbol{\mathcal{P}3}$ and $\tilde{N} = 3N + 4M + 1$ for problem $\boldsymbol{\mathcal{P}4}$.

Proof: There are two bottleneck lines in the complexity analysis of Algorithm 2, i.e., lines 3 and 7. Indeed, the complexity of other lines is $\mathcal{O}(1)$. Checking the feasibility of the current point in problem (17) needs to examine the point in all constraints (22). In each iteration of Algorithm 2, it is needed to solve the linear problem (21). The complexity of this task also depends on the number of constraints. Indeed, practical experience indicates that the number of iterations of the simplex algorithm is of order $\mathcal{O}(m)$ where m is the number of constraints in average case [45], [46]. Thus, each iteration of the algorithm runs in a linear time. The number of constraints of problems $\boldsymbol{\mathcal{P}3}$ and $\boldsymbol{\mathcal{P}4}$, except the constraint that bounds the variables (i.e., constraint **C1** in $\boldsymbol{\mathcal{P}3}$ and constraints **C1**, **C5**, **C7**, and **C10** in $\boldsymbol{\mathcal{P}4}$), is equal to $2N + M^2$ and $3N + 4M + 1$, respectively. Note that the number of constraints of problems in forms (17) and (21) is the same. Altogether, the complexity of each iteration of Algorithm 2 is equal to $\mathcal{O}(\tilde{N})$ where $\tilde{N} = N + M^2$ for problem $\boldsymbol{\mathcal{P}3}$ and $\tilde{N} = 3N + 4M + 1$ for problem $\boldsymbol{\mathcal{P}4}$. Assuming that the number of iterations of the while loop is equal to S, the complexity of Algorithm 2 is equal to $\mathcal{O}(S\tilde{N})$. ■

Based on Propositions 1 and 2, it is straightforward that the complexity of Algorithm 3 is mainly due to Algorithm 1 and the complexity of Algorithm 1 is dominant.

VI. Simulation Results

In this section, we present some simulation results to validate the proposed joint optimization scheme and theoretical results. Furthermore, we provide more insights on the effectiveness of the proposed optimal multi UAV 3-D placement, UAVs-users association, and resource allocation schemes. We consider a multi UAV-aided MEC network where users are distributed in a geographical urban area of size 1 km × 1 km with $a = 9.61$ and $b = 0.16$ at 2-GHz carrier frequency. Throughout the simulation, $N = 100$ is considered, unless otherwise mentioned. The remaining important simulation parameters adopted in the performance evaluation are summarized in Table I.

TABLE I
Simulation Parameters

Parameter	Value
n	2
M	15
(δ, α)	$(10^{-3}, 10^{-5})$
$p_{i_m}^{\max}$	23 dBm
f_c	2 GHz
$[\xi^{LoS}, \xi^{NLoS}]$	$[1, 20]$ dB
k_m	10^{-26}
$[h_m^{\min}, h_m^{\max}]$	$[50, 500]$ m
W	20 MHz
N_0	-174 dBm/Hz
β_{i_m}	uniform $[300 - 800]$ KB
v_{i_m}	uniform $[0.1 - 1]$ Gigacycles
T_m^{ul}	uniform $[10 - 20]$ sec
$E_m^{\max}$	uniform $[5 - 15]$ Joule
G_m^{cloud}	5×10^8 cycles per second
F_m^{uav}	uniform $[0.1-1]$ Gigacycles per second

Algorithm 3 Pseudocode of the Proposed Algorithm for Solving Problem $\boldsymbol{\mathcal{P}2}$

Input: The position of users, M and N.

1: Calculate N^{uav} according to $N^{\text{uav}} = \left\lfloor \frac{\psi}{R_{i_m}^{\min}} \right\rfloor$.
2: Set $\tilde{M} \leftarrow \left\lceil \frac{N}{N^{\text{uav}}} \right\rceil$.
3: Set $\hat{M} \leftarrow \min\{\tilde{M}, M\}$.
4: Set $\upsilon \leftarrow 1$.
5: Set $\ell_1 \leftarrow \hat{M}$.
6: Set $\Phi_0^* \leftarrow 0$.
7: **Repeat**
8: Calculate $\mathcal{I}, (c_1, \ldots, c_\ell), (\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_\ell)$ using **Algorithm** 1.
9: Solve problem $\boldsymbol{\mathcal{P}3}$ to obtain $(\mathbf{h}^{\text{uav}})_\upsilon^*$ using **Algorithm** 2.
10: Solve problem $\boldsymbol{\mathcal{P}4}$ to obtain $(\boldsymbol{\mathcal{S}}\backslash\{\mathbf{x}^{\text{uav}}, \mathbf{y}^{\text{uav}}, \mathbf{h}^{\text{uav}}\})_\upsilon^*$ using **Algorithm** 2.
11: Calculate Φ_υ^* by substituting $\boldsymbol{\mathcal{S}}_\upsilon^*$ in Φ.
12: Set $\upsilon \leftarrow \upsilon + 1$.
13: Set $\ell_\upsilon \leftarrow \hat{M} - 1$.
14: **Until** $\Phi_\upsilon^* \le \Phi_{\upsilon-1}^*$.
15: Set $\boldsymbol{\mathcal{S}}^{otp} \leftarrow \boldsymbol{\mathcal{S}}_{\upsilon-1}^*$.
16: Set $\ell^{otp} \leftarrow \ell_{\upsilon-1}^*$.

Output: $\boldsymbol{\mathcal{S}}^{otp}, \ell^{opt}$.

We evaluated the convergence of Algorithm 1 (i.e., MGKM scheme) and Algorithm 2 (i.e., SLP method), which constitute Algorithm 3. Fig. 2 shows the number of required iterations in the convergence to the optimal solution for Algorithms 1 and 2.

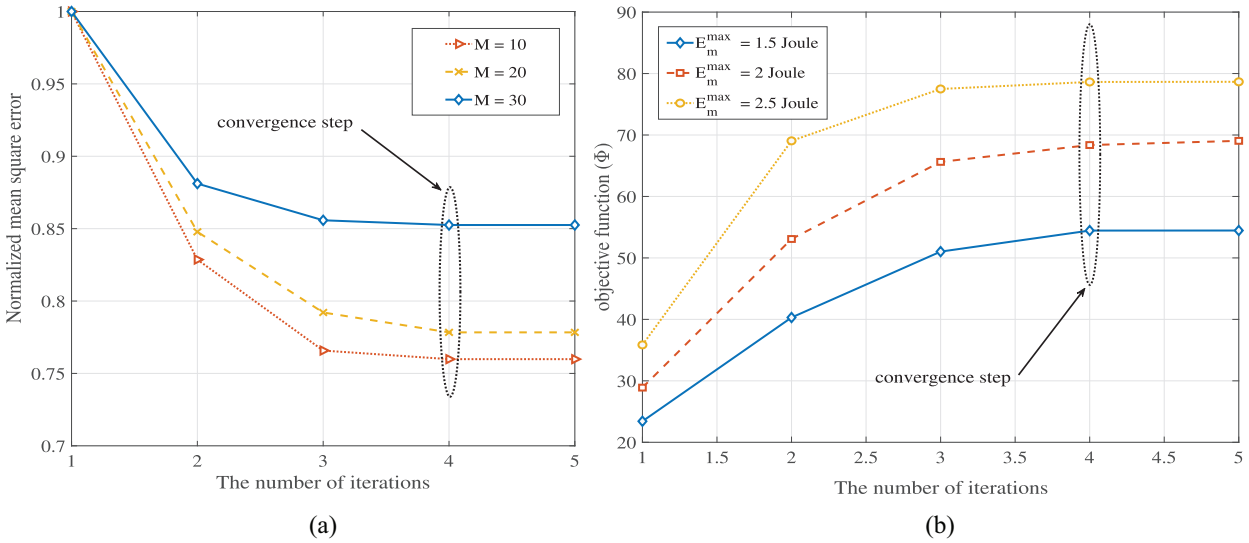

Fig. 2. Convergence behavior of the proposed algorithms for (a) normalized mean square error in Algorithm 1 and (b) objective function of $\mathcal{P}2$ in Algorithm 2.

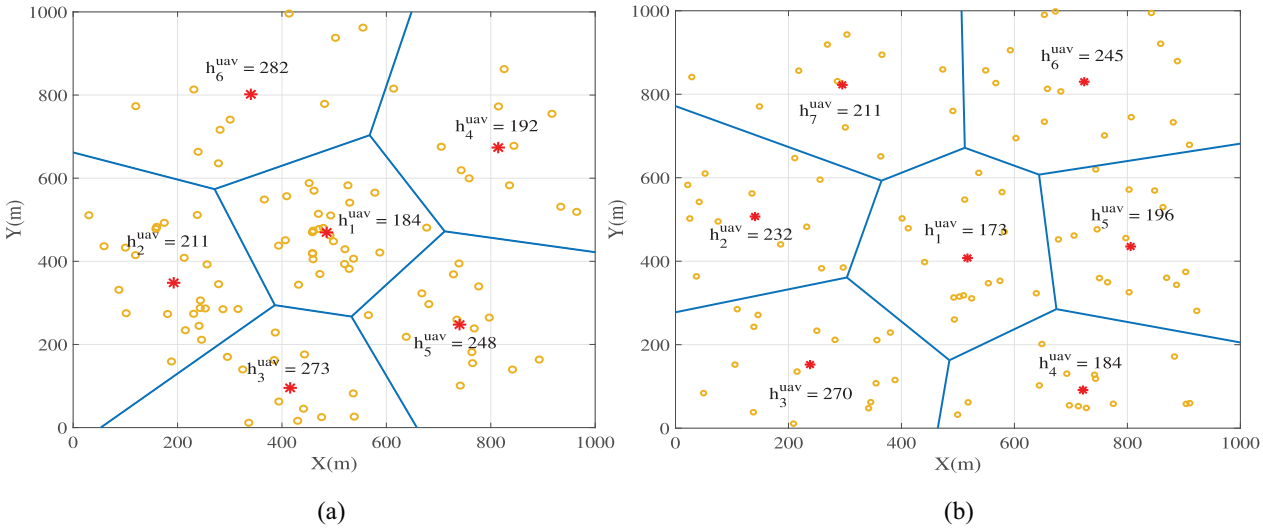

Fig. 3. User association and 2-D placement of UAVs, for (a) Gaussian distribution and (b) uniform distribution for users.

The value of normalized mean square error as a function of the number of iterations indicates the convergence of Algorithm 1, as illustrated in Fig. 2(a) for three different numbers of clusters (i.e, $M = 10, 20, 30$). The normalized mean square error in Algorithm 1 converges to a stable value after four steps for all three values of M. Fig. 2(b) shows the final values of the objective function Φ in problem $\mathcal{P}2$ versus the number of iterations under three different energy budgets $E_m^{\max} \in \{1.5, 2, 2.5\}$ joules, which indicate the convergence of Algorithm 2. We considered 8eight UAVs serving 100 users for this setup. Both Algorithms 1 and 2 attain convergence to stable values after four steps. It can be seen that the primitive solutions are relatively low. It comes from the fact that the initial solution employs random values from the feasible set, which is most likely far from the optimal solution. After three iterations, the total overhead is greatly reduced. The fast convergence of the optimization strategy takes only four iterations. For instance, the termination criterion is satisfied with $\delta = 10^{-3}$ for $\alpha = 10^{-5}$. The main benefit of the fast convergence of the proposed algorithm is fast reconfiguration if more ISDs and applications are used in the network, which makes the configuration of a dynamic environment more flexible.

Fig. 3 depicts a snapshot of the optimal 2-D view of UAVs placement and users' association resulting from our proposed algorithm. The association policy is based on the best SINR. Moreover, in order to provide a better understanding of the UAVs placement, the Voronoi tessellation of UAVs is shown. It is worth mentioning that continuous lines do not display the actual boundaries for the users' connection. For this placement, we considered an initial set of eight UAVs to support 100 users. The users are assumed to be distributed across the network following uniform and Gaussian distributions. We assume that each UAV can be allocated to at most 40 users due to the limited number of resource blocks available at UAVs. In this figure, the proposed MGKM clustering algorithm divided users into six and seven

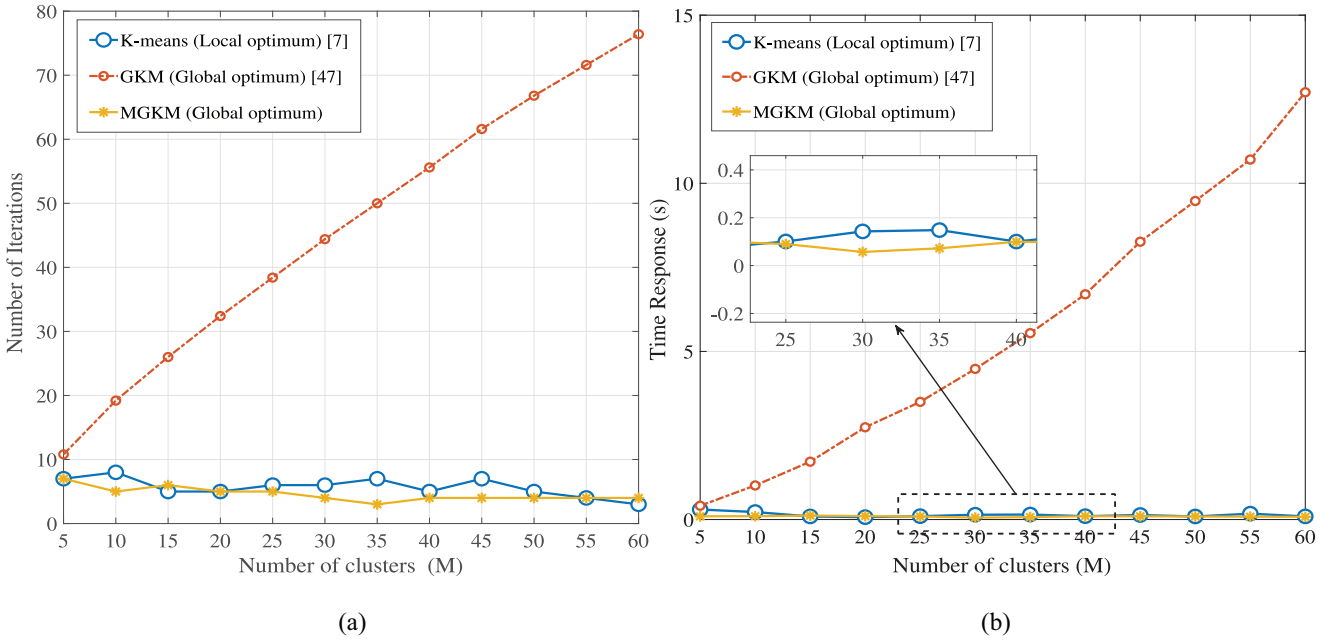

Fig. 4. Comparison of convergence behavior of K-means [7], GKM [47], and the proposed MGKM algorithms, for (a) number of iterations versus different numbers of clusters and (b) time response versus different number of clusters.

clusters when users are distributed using Gaussian and uniform distribution, respectively. The cluster centers are related to the locations of UAVs. As can be seen in Fig. 3, only one or two UAVs are found redundant, which confirms that the initial estimate is very close to the actual number of UAVs deployed. It is clear that the locations of users affect the locations of UAVs and also the number of users per cluster.

The optimal heights of UAVs for such a network topology are depicted in Fig. 3. The optimal height values show that the UAVs have covered the network by adjusting their heights to reduce the interference, which consequently improves the network performance. We can observe that the UAVs attain low altitudes in clusters where the density of users is high. This is because, during the formation of a cluster, the resources of the drone are assigned to the users that belong to its cluster; hence, the drone is not capable of providing services to users outside its cluster. In this case, the UAV tries to decrease its altitude for causing less interference to farther users that are served by UAVs of adjacent clusters. In contrast, the UAVs try to increase their heights in those clusters where the density of users is low so that the path loss can be decreased and more users can be included in their clusters.

Fig. 4 shows the performance of the proposed MGKM clustering algorithm from two different perspectives: 1) number of iterations and 2) response time. In Fig. 4(a) and (b), the number of iterations and the time required to converge to the optimal solution versus the number of users are compared, respectively. Furthermore, in this figure, the proposed MGKM algorithm is compared with the two algorithms K-means and GKM, which are used in [7] and [47], respectively. In this figure, it is assumed that the number of users varies: $N = 5, 10, 15, \ldots, 60$. It can be observed from Fig. 4(a) that the GKM algorithm requires a large number of iterations to achieve the optimal solution as compared to the other two algorithms, and it shows significantly worse performance than the other two methods. The K-means and MGKM algorithms show comparable performance. But in most cases, the MGKM algorithm works better than the K-means scheme and converges to the stable value in fewer iterations. In addition, we can see from Fig. 4(b) that both the MGKM and K-means show comparable and significantly lower response times than the GKM algorithm. However, close examination of all clusters shows that in some cases the MGKM's response times are lower than that of K-means. From Fig. 4, it can be concluded that both MGKM and K-means algorithms are much faster than the GKM both in terms of the number of iterations and response time, and the MGKM algorithm shows even marginally improved performance than the K-means method. It should be noted that the K-means scheme converges to the local solution, whereas both the GKM and MGKM algorithms converge to the global solution. In summary, the MGKM algorithm displays superior performance in comparison to other benchmark schemes.

In addition, the convergence behavior of GKM, K-means, and the proposed MGKM algorithms versus the number of clusters $M = 10, 20, 30, 40,$ and 50 is compared from three aspects in Table II: 1) the total number of iterations; 2) total time response; and 3) one iteration duration. From Fig. 4 and Table II, it can be concluded that both MGKM and K-means algorithms are much faster than the GKM both in terms of the number of iterations and response time, and the MGKM algorithm shows even marginally improved performance than the K-means method. This superiority comes from the fact that the K-means scheme converges to the local solution, whereas

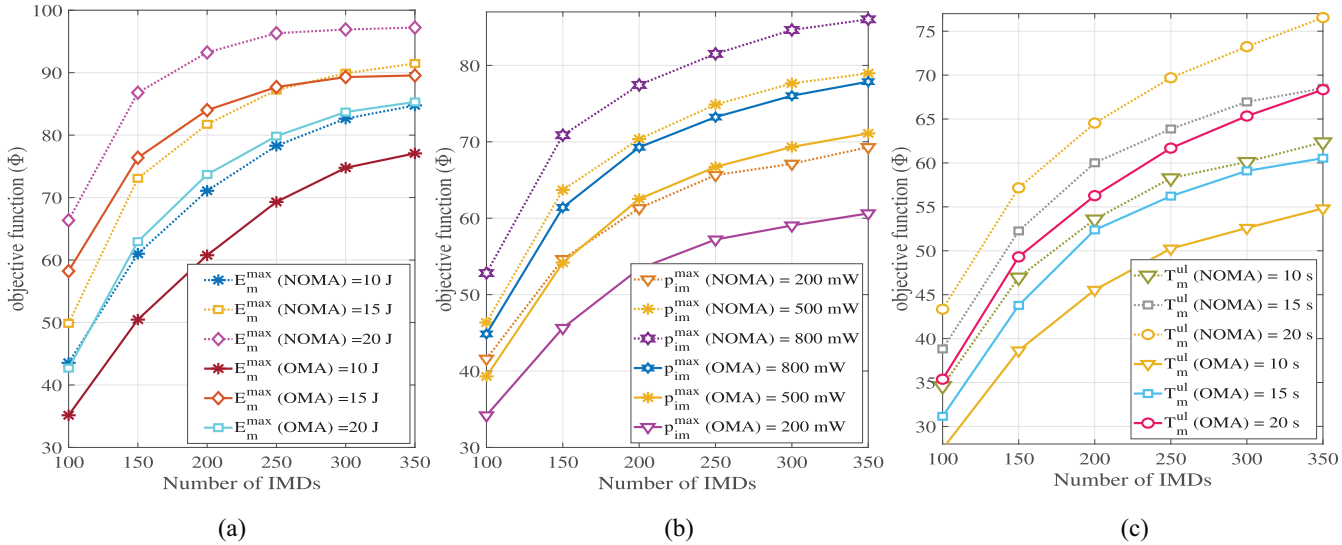

Fig. 5. Objective function of problem $\mathcal{P}2$ (i.e., Φ) versus the total number of users under different $E_m^{\max}$, $p_{i_m}^{\max}$, and T_m^{ul} for two cases of NOMA and OMA schemes.

TABLE II
COMPARISON OF THE CONVERGENCE BEHAVIOR OF GKM [47], *K*-MEANS [7], AND THE PROPOSED MGKM ALGORITHMS VERSUS DIFFERENT NUMBER OF CLUSTERS (M), FROM THREE ASPECTS: 1) TOTAL NUMBER OF ITERATIONS; 2) TOTAL TIME RESPONSE; AND 3) ONE ITERATION DURATION

	Number of clusters (M)				
Algorithm	10	20	30	40	50
GKM (Iterations)	21	33	46	58	70
GKM (Total time (s))	1.91	3.11	5.43	7.74	10.90
GKM (One iteration duration (s))	0.094	0.094	0.118	0.129	0.155
K-means (Iterations)	7	7	7	6	6
K-means (Total time (s))	1.22	1.02	0.87	0.95	1.27
K-means (One iteration duration (s))	0.174	6.862	8.04	6.31	4.72
Proposed MGKM (Iterations)	6	7	7	4	5
Proposed MGKM (Total time (s))	0.52	0.12	0.85	0.68	0.96
Proposed MGKM (One iteration duration (s))	0.086	0.017	0.12	0.17	0.192

the MGKM algorithm converges to the global point. It is worth mentioning that one of the most important factors in the proposed MGKM algorithm, which is led to the computational and time complexity improvement, is the mechanism of determining the best initial center for new clusters at each stage calculated from (14) and (15). In addition, another important factor considered in the MGKM algorithm that is led to a further reduction in the computational and time complexity is the initial estimation of the number of UAVs.

Fig. 5 shows the impact of $E_m^{\max}$, $p_{i_m}^{\max}$, and T_m^{ul} restrictions on the objective function of problem $\mathcal{P}2$ (i.e., Φ) by a varying the number of users. In addition, the performance of the NOMA scheme over the conventional OMA counterpart has been investigated in this figure, where continuous and dotted lines represent NOMA and OMA schemes, respectively. It is observed that increasing the number of users has a positive effect on the performance of the objective function Φ. In fact, as the number of users in the network increases and according to $N^{\mathrm{uav}} = \left\lfloor \psi/(R_{i_m}^{\min}) \right\rfloor$, more drones are employed, which increases the computational resources and the number of clusters. This increase in the number of clusters leads to a reduction in the size of the cluster, or equivalently, the average distance between the users and the corresponding drone is reduced, which in turn improves the channel conditions. Furthermore, it can be seen that the objective function Φ increases with larger values of the $E_m^{\max}$, $p_{i_m}^{\max}$, and T_m^{ul} limits. This is because when the feasible region of the optimization problem $\mathcal{P}1$ enlarges, higher Φ can be attained. On the other hand, it is seen that the performance of the network is improved by applying the NOMA scheme in comparison with the OMA counterpart. For example, for $N = 250$ and by setting $E_m^{\max} = 15$ J, $P_{i_m}^{\max} = 500$ mW, and $T = 15$ s, employing the NOMA scheme leads to an improvement of the network performance with the value of 10%, 11%, and 12% compared to the OMA counterpart.

In Fig. 6, we investigate the objective function of problem $\mathcal{P}2$ (i.e., Φ) by varying $E_m^{\max}$ for different values of h_m^{UAV}, $p_{i_m}^{\max}$, and T_m^{ul}. In Fig. 6(a), we examined the effect of optimal UAV altitude selection on the cost function. The figure compares the value of Φ for the optimal values of h_m^{UAV} obtained from Algorithm 2 when the altitude of UAVs is constant (for example $h_m^{\mathrm{UAV}} = 300$ and 500 m). First, it can be seen that

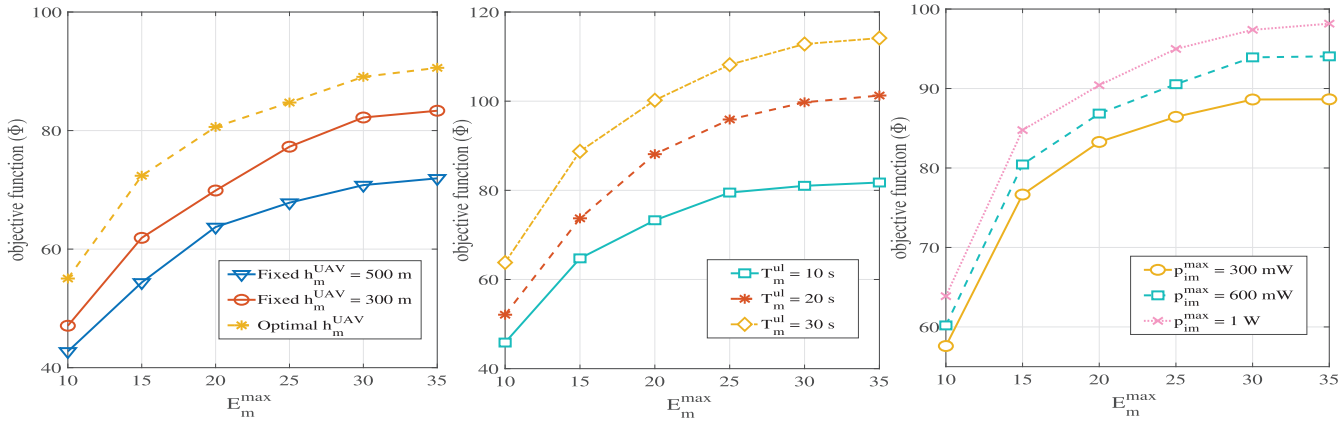

Fig. 6. Objective function of problem $\mathcal{P}2$ (i.e., Φ) versus the maximum UAV's energy budget under different h_m^{UAV}, T_m^{ul}, and $p_{i_m}^{\max}$ for $N = 100$.

increasing the value of $E_m^{\max}$ has a positive effect on the value of Φ in all cases. Second, the network performance decreases by increasing the height of the drone, because when UAVs hover at lower heights, the channel fading is less severe. The downside of the low altitude of UAVs is a reduction in the coverage area. Also, the performance gap for the optimal mode and the case in which the UAV height is fixed is quite obvious. For example, for $E_m^{\max} = 25$ joules, it can be seen that the value of Φ relative to $h_m^{\text{UAV}} = 300$ and 500 m has improved by 10% and 20%, respectively, and confirms the effectiveness of the proposed algorithm.

Due to the complexity of the mathematical proof of convergence and the optimality of the proposed Algorithm 3, we numerically examine this issue as shown in Fig. 7. For this evaluation, it is assumed that 100 users are uniformly distributed in the network area. For each specific number of clusters, three clustering algorithms were executed 100 times started from random initial points for the centers. We evaluated the average and the minimum clustering error along with its standard deviation versus the number of clusters for traditional GKM, K-means, and the proposed clustering algorithm in Fig. 7. It can be seen that Algorithms 3 and GKM display the same behavior and converge to a specific point, i.e., global optimum response, while the K-means algorithm converges to different solutions (local optimal) in each execution. More precisely, Algorithm 3 and GKM are not sensitive to starting points and both have a convergent global optimal response. However, the K-means algorithm is sensitive to the starting point and converges to a local point in each iteration. According to these results, it can be concluded that the proposed algorithm is experimentally global optimal.

In Fig. 8, we examine the system performance by evaluating the number of UAVs versus the number of users changing from 100 to 300. The figure shows both the final number of UAVs obtained from the algorithm (i.e., M^{opt}) and the initial estimate of the number of UAVs (i.e., $\hat{M}$). Due to further validation, simulation results obtained for M^{opt} are averaged over ten independent executions. It is evident from Fig. 8 that the

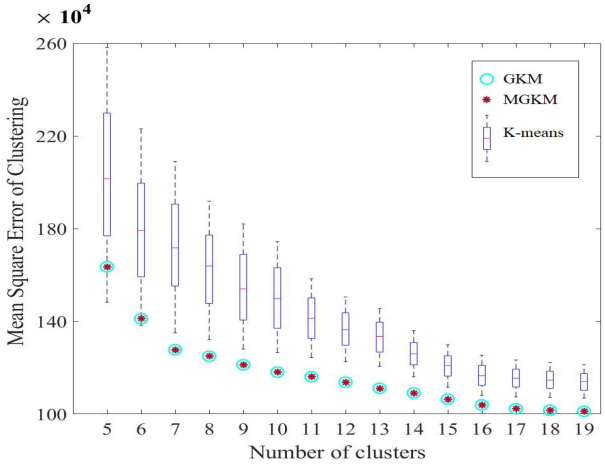

Fig. 7. Comparison of the mean square error of three different clustering algorithms versus the number of clusters.

initial estimation of the number of UAVs is very close to the actual value obtained from the algorithm. This means that the computational complexity of solving problems can significantly reduce leading to a faster convergence of the problem to the optimal solution.

VII. Conclusion

In this article, we presented the joint optimization of multi UAV 3-D placement, resource allocation, and UAVs-users association problem in the multi UAV-enabled MEC uplink network with NOMA. Maximizing the energy efficiency was the main goal of this work, which is achieved by utilizing the minimum number of UAVs and incurring the reduced costs while satisfying some existing QoS constraints. We investigated an urban area with users having different rate requirements. The underlying problem was nonconvex and MINLP, which is intractable to solve. To handle this challenging problem, we proposed an iterative scheme to solve the optimization problem in three steps within a few iterations. We divided the initial optimization problem into

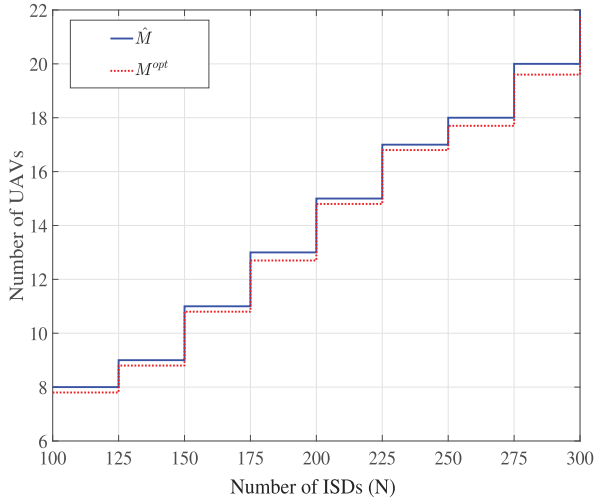

Fig. 8. Comparison of the number of UAVs obtained from the algorithm (i.e., M^{opt}) and the initial estimation of the number of UAVs (i.e., $\hat{M}$), versus the number of users.

three subproblems: 1) UAVs-users associations; 2) altitudes optimization; and 3) 2-D positioning of UAVs. We employed a low complexity algorithm solving locally each subproblem. To achieve the users clustering, we proposed the MGKM scheme-based upon the K-means clustering algorithm and machine learning theory. Moreover, an SLP method for finding the flying heights and resource allocation was proposed. The fast convergence of the proposed algorithms was verified through the simulation. In addition, it was demonstrated that the combination of UAV communication and the NOMA technique is beneficial for developing high-performance IoT systems.

So far, we assumed that users in the network are static or move at a low velocity and our main focus was on the 3-D placement of quasistationary UAVs. One possible future research work is to include the velocity and the flight energy consumption of UAVs in the trajectory optimization of UAVs in MEC architecture and solve the multiobjective optimization problem using deep reinforcement learning algorithms.

REFERENCES

[1] 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.

[2] 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.

[3] 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, pp. 1–20, Apr. 2020.

[4] E. Kalantari, M. Z. Shakir, H. Yanikomeroglu, and A. Yongacoglu, "Backhaul-aware robust 3D drone placement in 5G+ wireless networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2017, pp. 109–114.

[5] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927–1941, Sep. 2018.

[6] H. Wang, J. Wang, G. Ding, J. Chen, Y. Li, and Z. Han, "Spectrum sharing planning for full-duplex UAV relaying systems with underlaid D2D communications," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1986–1999, Sep. 2018.

[7] 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.

[8] Z. Askari, J. Abouei, M. Jaseemuddin, and A. Anpalagan, "Energy efficient and real-time NOMA scheduling in IoMT-based three-tier WBANs," *IEEE Internet Things J.*, early access, Mar. 30, 2021, doi: 10.1109/JIOT.2021.3069659.

[9] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: A crowd surveillance use case," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 128–134, Feb. 2017.

[10] S. Shakoor, Z. Kaleem, D.-T. Do, O. A. Dobre, and A. Jamalipour, "Joint optimization of UAV 3D placement and path loss factor for energy efficient maximal coverage," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9776–9786, Jun. 2021.

[11] L. Zhang and N. Ansari, "Approximate algorithms for 3-D placement of IBFD enabled drone-mounted base stations," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7715–7722, Aug. 2019.

[12] 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. (ICCT)*, Oct. 2018, pp. 967–972.

[13] Y. Sun, D. Xu, D. W. K. Ng, L. Dai, and R. Schober, "Optimal 3D-trajectory design and resource allocation for solar-powered UAV communication systems," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4281–4298, Jun. 2019.

[14] Y. Guo, S. Yin, and J. Hao, "Resource allocation and 3-D trajectory design in wireless networks assisted by rechargeable UAV," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 781–784, Jun. 2019.

[15] 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.

[16] M. Taghavi and J. Abouei, "Two-dimensional drone base station placement in cellular networks using MINLP model," *Int. J. Electron. Telecommun.*, vol. 65, no. 4, pp. 701–706, Sep. 2019.

[17] 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, 2021.

[18] A. Merwaday, A. Tuncer, A. Kumbhar, and I. Guvenc, "Improved throughput coverage in natural disasters: Unmanned aerial base stations for public-safety communications," *IEEE Veh. Technol. Mag.*, vol. 11, no. 4, pp. 53–60, Dec. 2016.

[19] Q. Wang, Z. Chen, W. Mei, and J. Fang, "Improving physical layer security using UAV-enabled mobile relaying," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 310–313, Jun. 2017.

[20] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, Mar. 2017.

[21] 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.

[22] C. Shen, T.-H. Chang, J. Gong, Y. Zeng, and R. Zhang, "Multi-UAV interference coordination via joint trajectory and power control," *IEEE Trans. Signal Process.*, vol. 68, no. 7, pp. 843–858, Jun. 2020.

[23] N. Cheng *et al.*, "Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 26–32, Aug. 2018.

[24] S. Jeong, O. Simeone, and J. Kang, "Mobile cloud computing with a UAV-mounted cloudlet: Optimal bit allocation for communication and computation," *IET Commun.*, vol. 11, no. 7, pp. 969–974, Jan. 2017.

[25] 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. 2018.

[26] J. Lyu, Y. Zeng, and R. Zhang, "UAV-aided offloading for cellular hotspot," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3988–4001, Jun. 2018.

[27] X. Jiang, Z. Wu, Z. Yin, W. Yang, and Z. Yang, "Trajectory and communication design for UAV-relayed wireless networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 6, pp. 1600–1603, Jul. 2019.

[28] J. Fan, M. Cui, G. Zhang, and Y. Chen, "Throughput improvement for multi-hop UAV relaying," *IEEE Access*, vol. 17, pp. 147732–147742, 2019.

[29] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4983–4996, Dec. 2016.
[30] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.
[31] 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 Wireless Commun. Lett.*, vol. 8, no. 3, pp. 913–916, Jun. 2019.
[32] 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, Apr. 2019.
[33] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 2012.
[34] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, "Joint offloading and trajectory design for UAV-enabled mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1879–1892, Oct. 2018.
[35] 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. (IST)*, Dec. 2018, pp. 522–527.
[36] 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 (IWCIT)*, Jul. 2018, pp. 1–6.
[37] D. Hu, Y. M. Alsmadi, and L. Xu, "High-fidelity nonlinear IPM modeling based on measured stator winding flux linkage," *IEEE Trans. Ind. Appl.*, vol. 51, no. 4, pp. 3012–3019, Jul./Aug. 2015.
[38] A. Likas, N. Vlassis, and J. J. Verbeek, "The global K-means clustering algorithm," *Pattern Recognit.*, vol. 36, no. 2, pp. 451–461, Feb. 2003.
[39] J. Nocedal and S. Wright, *Numerical Optimization*. Heidelberg, Germany: Springer, 2006.
[40] S. P. Han and O. L. Mangasarian, "Exact penalty functions in nonlinear programming," *Math. Program.*, vol. 17, no. 1, pp. 251–269, 1979.
[41] C. M. Chin and R. Fletcher, "On the global convergence of an SLP—Filter algorithm that takes EQP steps," *Math. Program.*, vol. 96, no. 1, pp. 161–177, Jul. 2003.
[42] A. Huang, C. Xu, and M. Wang, "A modified SLP algorithm and its global convergence," *J. Comput. Appl. Math.*, vol. 235, no. 14, pp. 4302–4307, May 2011.
[43] E. Riccietti, S. Bellavia, and S. Sello, "Sequential linear programming and particle swarm optimization for the optimization of energy districts," *Eng. Optim.*, vol. 51, no. 1, pp. 84–100, Jan. 2019.
[44] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A K-means clustering algorithm," *J. Roy. Stat. Soc. C Appl. Stat.*, vol. 28, no. 1, pp. 100–108, Jan. 1979.
[45] R. E. Quandt and H. W. Kuhn, "Letter to the editor—On upper bounds for the number of iterations in solving linear programs," *Oper. Res.*, vol. 12, no. 1, pp. 161–165, Feb. 1964.
[46] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*. Hoboken, NJ, USA: Wiley, 2010.
[47] M. K. Shehzad, S. A. Hassan, A. Mahmood, and M. Gidlund, "On the association of small cell base stations with UAVs using unsupervised learning," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–5.

Nima Nouri (Student Member, IEEE) received the B.Sc. degree in electrical engineering from the Shahid Bahonar University of Kerman, Kerman, Iran, in 2014, and the M.Sc. degree in electrical engineering from Yazd University, Yazd, Iran, in 2017.

Since 2017, he has been a Research Assistant with WINEL, Yazd University. His main research interests include Internet of Things, UAV communication systems, edge/fog computing, resource allocation, and nonconvex optimization.

Jamshid Abouei (Senior Member, IEEE) received the B.Sc. degree in electronics engineering and the M.Sc. degree in communication systems engineering from Isfahan University of Technology, Isfahan, Iran, in 1993 and 1996, respectively, and the Ph.D. degree in electrical engineering from University of Waterloo, Waterloo, ON, Canada, in 2009.

In 1996, he joined as a Lecturer with the Department of Electrical Engineering, Yazd University, Yazd, Iran, where he was promoted to an Assistant Professor in 2010, and an Associate Professor in 2015. From 2009 to 2010, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. During his sabbatical, he was an Associate Researcher with the Department of Electrical, Computer and Biomedical Engineering, Ryerson University, Toronto, ON, Canada. His research interests are in 5G and wireless sensor networks, with a particular emphasis on PHY/MAC layer designs, including the energy efficiency and optimal resource allocation in cognitive cell-free massive MIMO networks, multiuser information theory, mobile-edge computing, and femtocaching.

Ali Reza Sepasian received the M.S. degree from Shiraz University, Shiraz, Iran, in 2007, and the Ph.D. degree from the Ferdowsi University of Mashhad, Mashhad, Iran, in 2014.

He is currently working with the Department of Computer Science, Fasa University, Fasa, Iran. His research interests are in the fields of optimization and operations research, especially, network problems, location problems, and inverse problems.

Muhammad Jaseemuddin (Member, IEEE) received the B.E. degree from N.E.D. University, Karachi, Pakistan, in 1989, the M.S. degree from the University of Texas at Arlington, Arlington, TX, USA, in 1991, and the Ph.D. degree from the University of Toronto, Toronto, ON, Canada, in 1997.

He worked with Advanced IP Group and Wireless Technology Lab, Nortel Networks, Ottawa, ON, Canada. He is a Professor and the Program Director of Computer Networks Program with Ryerson University, Toronto. His research interests include network automation, caching in 5G and ICN networks, context-aware mobile middleware and mobile cloud, localization, power-aware MAC and routing for sensor networks, heterogeneous wireless networks, and IP routing and traffic engineering.

Alagan Anpalagan (Senior Member, IEEE) received the B.A.Sc., M.A.Sc., and Ph.D. degrees in electrical engineering from the University of Toronto, Toronto, ON, Canada, in 1995, 1997, and 2001, respectively.

In 2001, he joined with the ELCE Department, Ryerson University, Toronto, where he was promoted to a Full Professor in 2010. He served the department in administrative positions as the Associate Chair, the Program Director for Electrical Engineering, and Graduate Program Director. During his sabbatical, he was a Visiting Professor with the Asian Institute of Technology, Khlong Nueng, Thailand, and a Visiting Researcher with Kyoto University, Kyoto, Japan. His industrial experience includes working for three years with bell mobility, nortel networks, and IBM. He directs a research group working on radio resource management and radio access and networking areas within the WINCORE Laboratory.

Dr. Anpalagan was a recipient of the IEEE Canada J.M. Ham Outstanding Engineering Educator Award in 2018, the YSGS Outstanding Contribution to Graduate Education Award in 2017, the Deans Teaching Award in 2011, and the Faculty Scholastic, Research, and Creativity Award thrice from Ryerson University. He served as an Editor for the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS from 2012 to 2014, IEEE COMMUNICATIONS LETTERS from 2010 to 2013, and *EURASIP Journal of Wireless Communications and Networking* from 2004 to 2009. He also served as the Guest Editor for six special issues published in IEEE, IET, and ACM. He served as the TPC Co-Chair of IEEE VTC Fall 2017, IEEE INFOCOM'16, IEEE Globecom15, and IEEE PIMRC'11. He served as the Vice Chair of IEEE SIG on Green and Sustainable Networking and Computing with Cognition and Cooperation from 2015 to 2018, IEEE Canada Central Area Chair from 2012 to 2014, IEEE Toronto Section Chair from 2006 to 2007, ComSoc Toronto Chapter Chair 2004 to 2005, and IEEE Canada Professional Activities Committee Chair 2009 to 2011.

Konstantinos (Kostas) N. Plataniotis (Fellow, IEEE) received the B.Eng. degree in computer engineering from University of Patras, Patras, Greece, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from Florida Institute of Technology Melbourne, Melbourne, FL, USA, in 1992 and 1994, respectively.

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. He has been holds the Bell Canada Endowed Chair in Multimedia since 2014. His research interests are primarily in the areas of image/signal processing, machine learning and adaptive learning systems, visual data analysis, multimedia and knowledge media, and affective computing.

Dr. Plataniotis has served as the Editor-in-Chief for the IEEE SIGNAL PROCESSING LETTERS. He was the Technical Co-Chair of the IEEE 2013 International Conference in Acoustics, Speech and Signal Processing, and he served as the inaugural IEEE Signal Processing Society Vice President for Membership from 2014 to 2016 and the General Co-Chair for the 2017 IEEE GLOBALSIP. He served as the General Co-Chair for the 2018 IEEE International Conference on Image Processing and IEEE International Acoustics, Speech and Signal Processing in 2021. He is the General Chair for the 2027 IEEE International Conference on Acoustics, Speech and Signal Processing, Toronto, ON, Canada. He is a Fellow of the Engineering Institute of Canada and the Canadian Academy of Engineering, and a Registered Professional Engineer in Ontario.