Novel Robust Dynamic Distributed Drone-Deployment Strategy for Channel-Capacity Optimization for 3-D UAV-Aided Ad Hoc Networks

Xiao Yan, *Member, IEEE*, Yehan Lin, Hsiao-Chun Wu, *Fellow, IEEE*, Qian Wang, and Shenglong Zhu

***Abstract*—Unmanned aerial vehicles (UAVs) are considered to be excellent candidates of airborne relays or base stations for the 3-D ad hoc networks. They can be promptly deployed to serve large bursts of communication traffic in cellular networks or provide timely network support in wireless sensor networks (WSNs). It is desirable but challenging to dynamically find the optimal deployment strategy of UAVs in the air to provide a better Quality of Service (QoS) for a UAV-assisted wireless network. In this article, we propose a novel Gibbs-sampling distributed algorithm (GSDA) to dynamically optimize the UAVs' locations when they serve as airborne base stations for ground users. In our proposed GSDA, channel capacity is adopted as the objective function and a distributed approach is employed such that each UAV is able to optimize its location independently and asynchronously. Furthermore, we propose a polynomial-regression-based predictor to make use of users' moving trajectories and take advantage of the predicted users' future locations to expedite the convergence of the GSDA. Meanwhile, we also compare our proposed GSDA with the existing distributed genetic algorithm. The asynchronization of UAV location updates and the location errors are also investigated to evaluate the robustness of the GSDA. Simulation results demonstrate that our proposed novel GSDA is quite robust and superior to the existing distributed genetic algorithm.**

***Index Terms*—Channel-capacity, deployment optimization, distributed algorithm, unmanned aerial vehicle (UAV) aided communications.**

I. Introduction

IN RECENT decades, unmanned aerial vehicle mounted base stations (UAV-BSs) have been applied for rapid network deployment due to their advantages in mobility and agility [1]. UAV-BSs can assist terrestrial cellular networks to improve the Quality of Service (QoS) in a wide variety of scenarios since UAV-BSs have a higher probability of experiencing a favorable Line-of-Sight (LoS) propagation path [2], [3], [4]. For example, when the terrestrial base stations are destroyed due to a natural disaster, the UAV-BSs can be deployed swiftly to provide a backup communication network. In addition, unmanned aerial vehicles (UAVs) also play a vital role in satellite communications, where a number of UAVs carrying communication payloads form a swarm to facilitate network routing between ground users and low Earth orbit (LEO) satellites [5]. However, due to the physical limit of a UAV, the battery it can carry in a single flight is restricted. Thus, a single UAV-BS indeed cannot provide continuous communication service like a terrestrial base station due to the battery limitation of its platform [2], but the sustained communication service can still be provisioned by UAV-BS replacement whenever a UAV-BS is exhausted in a UAV-assisted wireless network.

Due to the aforementioned UAVs' advantages, UAV-aided networks have been emerging for applications to the 5th generation (5G) and 6th generation (6G) wireless communications. An important problem in the UAV-assisted wireless systems is the location optimization of UAV-BSs for establishing a reliable dynamic communication network [6]. Different objectives have been proposed for optimization of UAVs' locations in the literature. A method to find the optimal altitude of low-altitude platforms (LAPs) and UAVs for coverage maximization was proposed in [7]. The optimal altitude of an LAP can be determined by searching for the critical height that satisfies the pathloss function. However, this method was based on the static LAPs and UAVs, and, thus, their mobility was not considered. Meanwhile, in [8], the optimal 3-D location of a UAV was found to maximize the coverage by decoupling such an optimization problem into the vertical and horizontal dimensions. The vertical-optimization method was proposed in [7] where the horizontal optimization was restricted to a circle placement. This method could ensure that every user to be within the coverage of a UAV-BS but the communication qualities of users were not considered. The trajectories of multiple UAVs could be planned to improve the capacity of the communication system by a constrained

Manuscript received 4 April 2022; revised 12 September 2022; accepted 7 January 2023. Date of publication 17 January 2023; date of current version 7 June 2023. This work was supported in part by the Natural Science Foundation of Sichuan Province under Grant 2022NSFSC0545, and in part by the Louisiana Board of Regents Research Competitiveness Subprogram under Grant LEQSF(2021-22)-RD-A-34. *(Corresponding author: Qian Wang.)*

Xiao Yan is with the School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the Aircraft Swarm Intelligent Sensing and Cooperative Control Key Laboratory of Sichuan Province, Chengdu 611731, China (e-mail: yanxiao@uestc.edu.cn).

Yehan Lin, Qian Wang, and Shenglong Zhu are with the School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: yehan@std.uestc.edu.cn; job_wangqian@uestc.edu.cn; shenglongzhu@std.uestc.edu.cn).

Hsiao-Chun Wu is with the School of Electrical Engineering and Computer Science, Louisiana State University, Baton Rouge, LA 70803 USA (e-mail: wu@ece.lsu.edu).

Digital Object Identifier 10.1109/JIOT.2023.3237656

deep Q-network (cDQN) [9]. The strategy in [9] was based on deep reinforcement learning, and such an approach considered both the real-time downlink capacity and a coverage constraint. This method can take advantage of UAVs' mobility to facilitate a more reliable network, but the deep reinforcement learning scheme requires enormous computation power along with sufficiently large training data and the undesirable centralized processing scheme. To maximize the secure cache throughput of a UAV-assisted network, the number of UAVs, their 3-D placements, and the cache placement probability of contents were jointly optimized in [10]. The optimization problem in [10] was decomposed into three optimization subproblems involving the horizontal placement, the altitude, and the secure cache throughput, respectively, due to the associated nonconcave objective function which inevitably leads to huge computational complexity. Moreover, the objectives of the optimization of UAVs' locations for the Internet of Things (IoT) are quite different from those for the UAV-assisted communication networks. In [11], the main objective of the 3-D deployment optimization of multi-UAVs is jointly maximizing the energy efficiency and minimizing the number of involved UAVs serving the ground-based devices. For wireless charging sensor networks (WCSNs), the first-rank UAV route can be determined by jointly optimizing the UAV's hovering location and duration to maximize the minimal energy of sensors after energy harvesting [12].

Different from the aforementioned studies in the existing literature, we would like to dedicate this work to dynamically adjusting multiple UAVs' locations for overall channel-capacity maximization in an arbitrary UAV-aided 3-D ad hoc network. It is intriguing to study how to deploy multiple UAVs in the air so that the overall network performance can be improved. A UAV-aided cellular wireless network was constructed to enhance the system capacity and coverage for ground users [13]. First, UAVs' locations were optimized to maximize the network coverage by different schemes in [8], [14], [15], and [16]. Second, UAVs' locations were optimized to maximize the network capacity by different *centralized* methods in [17], [18], [19], and [20]. Recently, we have conducted signal processing research for UAV controller detection in [21] and studied the problem of the optimal base-station (road-side unit) deployment in the 1-D space in [22] and [23]. Here, we would like to investigate a different and much more challenging problem involving the deployment of UAV-BSs in the 3-D space. In this work, we attempt to improve the QoS for the ground users by optimizing UAVs' locations. To achieve this goal, we assume that all ground users are located within the coverage area of a UAV-aided wireless network and adopt the channel capacity as our objective function. In the centralized approach to deploying UAVs, every UAV needs to broadcast to the control center while the control center should also broadcast to each UAV. Thus, it is quite difficult to implement such a centralized scheme as sophisticated synchronization and scheduling mechanisms are required. Even if we try to deploy the centralized approach, deciding where to locate the controller on the ground is quite challenging [21]. In order to avoid massive communications and stringent requirements of synchronization and scheduling incurred in the centralized approach, here, we focus on the distributed approach instead. Different from the approaches in [17], [18], [24], and [25], we design a novel distributed algorithm, by which each UAV can update its own location independently and asynchronously.

In this work, we propose a novel Gibbs-sampling distributed algorithm (GSDA) to dynamically optimize UAVs' locations. However, the underlying objective function, capacity, is a highly nonlinear complicated function. We adopt the pathloss model proposed in [7] to simplify the formula of channel capacity, which can simplify our optimization problem favorably. In our proposed distributed approach, the update-asynchronization situation is often encountered and needs to be investigated to evaluate the robustness of the GSDA. As the capacity function is nonconvex, the popular alternative to the GSDA is the genetic algorithm (GA). We modify the conventional GA to the distributed GA for comparison. The GSDA will be proven to converge under certain practical assumptions. To consider ground users' mobility, we also propose a polynomial-regression-based predictor to facilitate a more robust *look-ahead* version of GSDA which can make use of the predicted users' future locations. Crucial factors, including *update asynchronization* and *location* (*localization*) *errors* are also studied through simulations. Our proposed GSDA can help to establish robust UAV-aided 3-D wireless ad hoc networks in the future. Note that our proposed GSDA is established on the network topology under two assumptions as follows. Each UAV-BS has a fixed coverage area, i.e., all partitioned ground zones are invariant when UAV-BSs alter their locations, and only ground users within the same zone can be served by its corresponding UAV-BS. Meanwhile, each UAV-BS can only adjust its position within the boundaries of its coverage area to maintain the same connection topology of UAV-BSs. Furthermore, all UAV-BSs in the network must be capable of communicating with each other to know the locations of their surrounding UAV-BSs before each optimization iteration of our proposed GSDA. The main contributions of this work are summarized as follows.

1) In our work, we propose a novel robust GSDA for the UAV-aided communication network to achieve the maximum channel capacity based on the pathloss model.
2) A novel *weighted average strategy* is theoretically derived to merge the local UAV-location updates contributed by all UAV-BSs using a distributed algorithm to reach the ultimate optimal UAV-locations at every iteration.
3) An efficient *polynomial-regression-based predictor* is designed to predict future users' locations by utilizing users' historical locations. The predicted results are then employed as the presumptive input of the GSDA to adjust the UAV-BSs' locations well ahead of time for expediting the optimization process.

The remainder of this article is organized as follows. Section II presents the basic system model and configuration of a UAV-aided wireless ad hoc network consisting of multiple UAV-BSs and ground users. Our proposed novel robust dynamic distributed UAV-deployment strategy is introduced in Section III. Performance evaluation and comparison

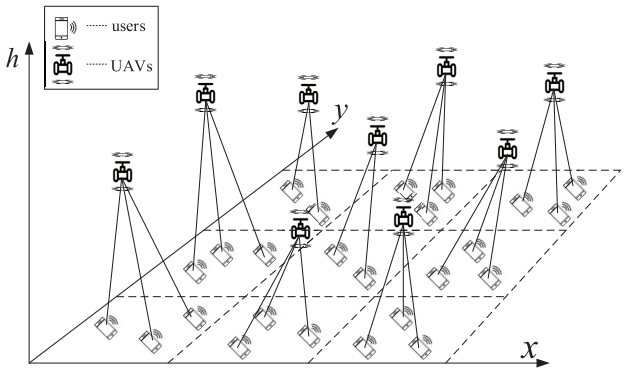

Fig. 1. Illustration of a UAV-aided wireless ad hoc network.

between the GSDA and the distributed GA will be presented in Section IV. Conclusion will be finally drawn in Section V.

Nomenclature: Scalars are denoted by Italianized alphabets such as a; vectors are denoted by alphabets with overhead arrow notions such as $\vec{A}$; matrices are denoted by alphabets with overhead tilde notions such as $\tilde{A}$; sets are denoted by blackboard bold alphabets such as $\mathbb{A}$. The cardinality of a set $\mathbb{A}$ is denoted by $|\mathbb{A}|$. $\vec{A}^{\mathrm{T}}$ and $\tilde{A}^{\mathrm{T}}$ represent the transposes of a vector $\vec{A}$ and a matrix $\tilde{A}$, respectively, and $\|\vec{A}\|$ denotes the Euclidean norm of $\vec{A}$. The sets of all real numbers and all natural numbers are denoted by $\mathbb{R}$ and $\mathbb{N}^+$, respectively. Note that α mod β will return the remainder of α divided by β.

II. System Model

A 3-D ad hoc communication network formed by mobile UAV-BSs is depicted by Fig. 1, which has been considered a potential solution to the future 5G/6G networks. Suppose that each UAV-BS has a fixed coverage area (we call it a "zone") such that it can adjust its location within the boundaries of this area according to ground users' (user equipments') mobility. Users can move within a zone or across different zones to suit the realistic scenario. Meanwhile, UAVs should optimize their locations (in terms of 3-D coordinates) to provide a better overall network service for all users. Without loss of generality, the underlying wireless network in a certain time instant is illustrated by Fig. 1. The corresponding channel model and network configuration will be discussed later on. Suppose that the aforementioned UAV-aided wireless network consists of J UAV-BSs in the air and I users on the ground. In practice, each user has only one connection with a UAV-BS. To improve the quality of such a UAV-aided wireless network, UAV-BSs should communicate with each other to exchange the information about their own locations. Each UAV-BS only has the communication links with its neighboring UAV-BSs; once a communication link between UAV j_1 and UAV j_2 is set up, these two drones can exchange information with each other.

A. Channel Model

In this section, we would like to establish the channel model for the UAV-aided wireless ad hoc network illustrated by Fig. 1. First, we need to characterize the air-to-ground (ATG) pathloss for different environments. The ATG pathloss depends on the UAV's height and the elevation angle between the UAV and its served user. There involve two propagation classes, namely, LoS connections and Nonline-of-Sight (NLoS) connections [26]. Therefore, the mean ATG pathloss $\mathrm{PL}_\zeta(t)$ at time t (in dB) is given by

$$\mathrm{PL}_\zeta(t) = \mathrm{FSPL}(t) + \eta_\zeta \tag{1}$$

where $\mathrm{FSPL}(t)$ represents the *free-space pathloss* between a UAV and its served ground user, $\zeta \in \{\mathrm{LoS}, \mathrm{NLoS}\}$, and η_ζ, $\zeta =$ "LoS" or "NLoS" denotes the *excessive pathloss* due to an LoS or NLoS channel between a UAV (say the UAV j) and its served user (say the user i). Thus, $\mathrm{FSPL}_{ij}(t)$ is given by

$$\mathrm{FSPL}_{ij}(t) = 20\log\left[\frac{4\pi d_{ij}(t) f}{c}\right] \tag{2}$$

where f specifies the carrier frequency of the transmitted signal, $d_{ij}(t)$ denotes the distance between the transmitter (UAV j) and the receiver (user i) at time t, and c represents the speed of light through air. According to [7], the probability of having an LoS connection between UAV j and user i can be expressed by

$$P\big(\theta_{ij}(t)|\mathrm{LoS}\big) = \frac{1}{1 + a\exp\big[-b\big(\theta_{ij}(t) - a\big)\big]} \tag{3}$$

where both a and b are the environment parameters such that

$$\theta_{ij}(t) \overset{\mathrm{def}}{=} \frac{180}{\pi} \times \tan^{-1}\left[\frac{\rho_j(t)}{\sigma_{ij}(t)}\right] \tag{4}$$

where $\rho_j(t)$ specifies the instantaneous altitude of the UAV j and $\sigma_{ij}(t)$ denotes the projection of the distance between the UAV j and the user i onto the ground at time t. Therefore, the spatial expectation of the pathloss given by (3) (measured in dB) between User i and UAV-BS j at time t is given by

$$\Lambda_{ij}(t) = \sum_{\zeta \in \{\mathrm{LoS}, \mathrm{NLoS}\}} \mathrm{PL}_\zeta \times P\big(\theta_{ij}(t)|\zeta\big) \tag{5}$$

where $P(\theta_{ij}(t)|\zeta)$ represents the probability of occurrence of a certain propagation dependent on the elevation angle between the UAV-BS j and the user i (which complies with (3) for $\zeta =$ LoS), and the probability of having an NLoS connection (for $\zeta =$ NLoS) is given by

$$P\big(\theta_{ij}(t)|\mathrm{NLoS}\big) = 1 - P\big(\theta_{ij}(t)|\mathrm{LoS}\big). \tag{6}$$

B. Network Configuration

Generally speaking, a seamless network coverage area served by drones can be partitioned into polygons. For analytic simplicity, we assume that each UAV will serve a square coverage area on the ground in this work. The configuration of our proposed UAV-aided wireless communication network is shown in Fig. 2, where there exist J UAV-BSs, each labeled by a circled number, and I users, each represented by a red dot. In the network illustrated by Fig. 2, each UAV can know its own 3-D coordinates from GPS (global positioning system). The center coordinates (x_j^C, y_j^C) of the jth UAV's coverage zone are represented by $\vec{v}_j \overset{\mathrm{def}}{=} (x_j^C, y_j^C)^{\mathrm{T}} \in \mathbb{R}^{2\times 1}$. Suppose that each UAV-BS cannot fly out of its coverage zone so that UAV-BS j

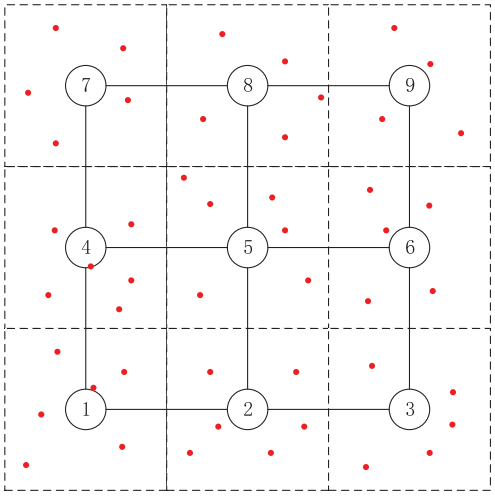

Fig. 2. Illustration of our proposed UAV-aided network configuration.

coverage area remains the same all the time. In addition, UAV j's 3-D coordinates are $\vec{L}_j(t) \stackrel{\text{def}}{=} (x_j^D(t), y_j^D(t), h_j^D(t))^{\mathrm{T}} \in \mathbb{R}^{3\times 1}$. Moreover, the UAV j should communicate with its neighbors by broadcast. In Fig. 2, solid lines between UAVs indicate established communications for exchanging information about their own locations with their neighbors. We define UAV $\bar{j}$ is a neighbor of UAV j only if

$$\left|x_j^C - x_{\bar{j}}^C\right| \le \left|x_j^C - x_{\bar{j}}^D(t)\right| \text{ or } \left|y_j^C - y_{\bar{j}}^C\right| \le \left|y_j^C - y_{\bar{j}}^D(t)\right|$$
$$\left|x_j^C - x_{\bar{j}}^C\right| = C \text{ or } \left|y_j^C - y_{\bar{j}}^C\right| = C, \quad j = 1, 2, \ldots, J \quad (7)$$

where C denotes the side length of a square coverage area. According to (7), the neighborhood relationship between UAV-BSs remains invariant even under UAVs' movements.

For any $j = 1, 2, \ldots, J$, the set of its neighboring zone indices $\bar{j}$ can be written as follows:

$$\mathbb{N}_j \stackrel{\text{def}}{=} \{\bar{j} | \bar{j} \text{ is a neighbor of } j \text{ according to (7)}$$
$$\bar{j} = 1, 2, \ldots, J\}. \quad (8)$$

The ground user i, which is located at $\vec{\ell}_i(t) \stackrel{\text{def}}{=} (x_i^U(t), y_i^U(t))^{\mathrm{T}} \in \mathbb{R}^{2\times 1}$, should connect to the UAV that covers the zone the user is located within. Thus, we have

$$\kappa_i(t) \stackrel{\text{def}}{=} \underset{j=1,2,\ldots,J}{\operatorname{argmin}} \left\|\vec{\ell}_i(t) - \vec{v}_j\right\| \quad (9)$$

and the set of users covered by UAV j at time t is given by

$$\mathbb{U}_j(t) \stackrel{\text{def}}{=} \{i | \kappa_i(t) = j\}. \quad (10)$$

Equation (10) implies that the UAV-BS connecting to the user i will change over time. In practice, each UAV-BS will know the users' locations to determine whether it should communicate with them or transfer them to the neighboring UAV-BSs.

According to (5), the ATG pathloss between the user i and its connected UAV $j = \kappa_i(t)$ is given by $\Lambda_{ij}(t)$. If a uniform transmitting power for all UAV-BSs is fixed to be P_T (measured in dBm), then the user i's received signal power $\Psi_i(t)$ is given by

$$\Psi_i(t) = P_T - \Lambda_{ij}(t) \quad (11)$$

where $j \stackrel{\text{def}}{=} \kappa_i(t)$. On the other hand, the interference $\Phi_i(t)$ coming from the connected UAV $\kappa_i(t)$'s neighbors $\bar{j} \in \mathbb{N}_{\kappa_i(t)}$ to the user i can be written as follows:

$$\Phi_i(t) = \sum_{\bar{j}\in\mathbb{N}_{\kappa_i(t)}} \delta\left(P_T - \Lambda_{i\bar{j}}(t)\right) \quad (12)$$

where $\delta()$ indicates the function converting the dBm value to watt. Furthermore, according to [17], [25], and [27], the *total channel capacity* for all users in the coverage area where the UAV j with the coordinates $\tilde{L}(t)$ serves at time t is given by

$$R_j\left(\tilde{L}(t)\right) = \sum_{i\in\mathbb{U}_j(t)} \log_2[1 + \mathrm{SNR}_i(t)] \quad (13)$$

where $\tilde{L}(t) \stackrel{\text{def}}{=} [\vec{L}_1(t), \vec{L}_2(t), \ldots, \vec{L}_J(t)] \in \mathbb{R}^{3\times J}$, $\mathrm{SNR}_i(t) = \Psi_i(t) - N_i(t)$, $N_i(t) \stackrel{\text{def}}{=} \gamma(\Phi_i(t) + \delta(\mathcal{N}))$ [25], $\mathcal{N}$ is the environmental noise power in dBm, and $\gamma()$ indicates the function converting the watt value to the dBm value.

C. Optimization Model

Sections II-A and II-B have presented the underlying channel model and network configuration of the proposed UAV-aided wireless ad hoc network. As previously stated, the coverage area of each UAV-BS remains the same all the time, while the location adjustment of each UAV-BS is restricted to be within its coverage area. According to (7), the neighborhood relationships between UAV-BSs remain the same all the time even under UAVs' movements.

In general, the higher channel capacity, the better performance a network can have. In this section, we will, thus, study the optimization problem to maximize the overall channel capacity $(1/I)\sum_{j=1}^{J} R_j(\tilde{L}(t))$ for all UAV-user links as both UAVs and users are mobile. So the target of the optimization is to maximize the mean channel capacity of all users. In order to guarantee that the user can receive the signal correctly, we should set a limitation: $\Psi_i(t) \ge P_{\min}$, where $P_{\min}$ is the minimum required received signal power. When the received signal strength reaches above this minimum, the information can be delivered correctly. Meanwhile, the height $h_j^D(t)$ of each UAV must satisfy $h_{\max} \ge h_j^D(t) \ge h_{\min}$ $\forall j$ such that all UAVs have to stay within such an altitude range. In summary, the optimization problem for our proposed UAV-aided ad hoc network is stated by

$$\max_{\tilde{L}(t)} \sum_{j=1}^{J} R_j\left(\tilde{L}(t)\right)$$
$$\text{s.t. } \Psi_i(t) \ge P_{\min} \quad \forall i \text{ and } h_{\max} \ge h_j^D(t) \ge h_{\min} \quad \forall j. \quad (14)$$

Note that the optimization problem formulated by (14) is not a convex optimization problem. Besides, we would like to seek a *distributed algorithm*, by which each UAV can adjust its own location independently at the same time without any need of a centralized control mechanism. We will discuss how to solve this optimization problem in the next section.

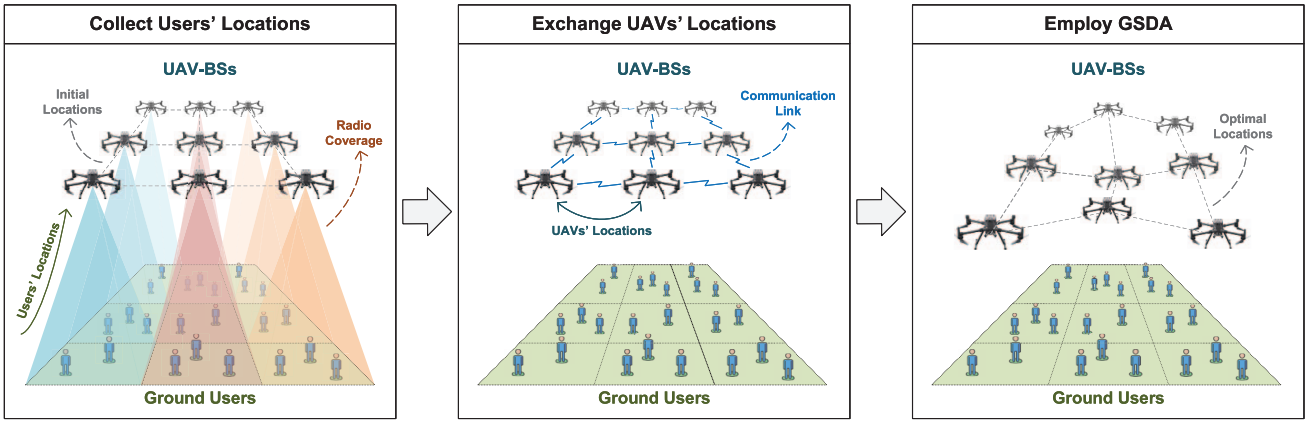

Fig. 3. Illustration of consecutive snapshots (critical steps) of our proposed UAV-assisted wireless communication network in an optimization interval.

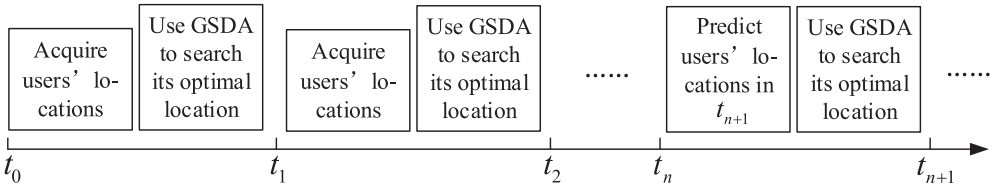

Fig. 4. Flow chart of a UAV-BS's actions in the proposed UAV-aided ad hoc network.

III. Proposed Novel Robust Dynamic Distributed UAV-Deployment Strategy

In this section, we propose a GSDA to optimize the UAV-BSs' locations as illustrated by Fig. 3. In an optimization interval, each UAV-BS collects its served users' locations and exchanges the location information with the neighboring UAV-BSs. The aforementioned location information of users and UAV-BSs is then utilized by the GSDA to find the optimal location for the UAV-BS to move to. The optimization problem is formulated by (14). Besides, a theorem is presented to show that our proposed algorithm will converge under some assumptions. Moreover, a user-movement predictor is also proposed in our algorithm to further boost the performance. Finally, we will discuss the computational complexity and the transmission overhead required by this algorithm.

A. Gibbs-Sampling Distributed Algorithm

Let us consider discrete-time instants, say $t = t_0, t_1, t_2, \ldots$; at each discrete-time instant, each UAV-BS adjusts its location and calculates its new location for the next time instant. Suppose that each UAV-BS will learn the current locations of all users within its coverage area and a few neighbors through communications at each time instant. Such a UAV-BS will execute the algorithm to search its optimal location independently of all other nonneighboring UAV-BSs. After several time instants, UAV-BSs have acquired enough previous locations of users to predict users' new locations. Hence, UAV-BSs may apply the predicted users' locations as the presumptive input to the algorithm to adjust the UAV-BSs' locations well ahead of time for expediting the learning. The actions of a UAV-BS can be demonstrated by the flow chart depicted in Fig. 4.

Without loss of generality, at a certain time instant, say t_n, we suppose that the UAV-BSs will learn all of their served ground users' locations as a collective matrix $\tilde{\ell}(t = t_n) \stackrel{\text{def}}{=} [\vec{\ell}_1(t), \vec{\ell}_2(t), \ldots, \vec{\ell}_I(t)]|_{t=t_n} \in \mathbb{R}^{2\times I}$. The UAV-BSs' positions can be collected as another matrix $\tilde{L}(t) = [\vec{L}_1(t), \vec{L}_2(t), \ldots, \vec{L}_J(t)]$. Let

$$\tilde{\mathcal{L}}_k(t_n) \stackrel{\text{def}}{=} \left[\vec{\mathcal{L}}_{1,k}(t_n), \vec{\mathcal{L}}_{2,k}(t_n), \ldots, \vec{\mathcal{L}}_{J,k}(t_n)\right] \in \mathbb{R}^{3\times J} \quad (15)$$

denote the collection of UAV-BSs' locations in kth iteration during the time interval $[t_n, t_{n+1})$ where the first iteration starts at $t = t_n$, $\tilde{\mathcal{L}}_0(t_n) \stackrel{\text{def}}{=} \tilde{L}(t_n)$, $\vec{\mathcal{L}}_{j,0}(t_n) \stackrel{\text{def}}{=} \vec{L}_j(t_n)$, and $\vec{\mathcal{L}}_{j,k}(t_n) \stackrel{\text{def}}{=} [x^D_{j,k}(t_n), y^D_{j,k}(t_n), h^D_{j,k}(t_n)]^{\mathrm{T}}$ specifies the UAV-BS j's x-axis, y-axis, and z-axis coordinates, respectively. The problem formulated by (14) is equivalent to

$$\min_{\tilde{L}(t)} \sum_{j=1}^{J} F_j\big(\tilde{L}(t)\big)$$
$$\text{s.t.} \quad \Psi_i(t) \geq P_{\min} \quad \forall i \text{ and } \quad h_{\max} \geq h_j{}^D(t) \geq h_{\min} \quad \forall j \quad (16)$$

where $F_j(\tilde{L}(t)) \stackrel{\text{def}}{=} -R_j(\tilde{L}(t))$. Then, we can use the initial UAVs' positions and ground users' positions (all at a time instant t_n) and apply our proposed new GSDA to solve the optimization problem given by (16). The procedure of our proposed new GSDA is demonstrated by Fig. 5.

After several time instants, UAV-BSs can collect sufficient information about the tracks of users within their coverage areas. Then UAV-BSs may utilize these information to predict users' new locations in the next time instant so that the GSDA

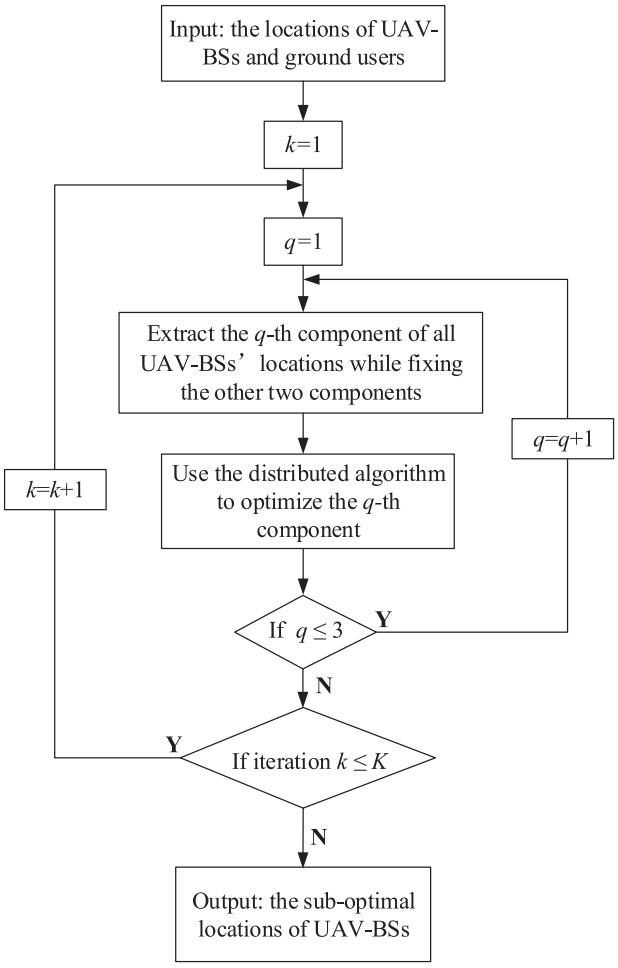

Fig. 5. Flow chart of our proposed GSDA.

can take advantage of these predicted users' locations to calculate the "prospected" optimal results under users' mobility. Meanwhile, the prediction mechanism can also predetermine the connections between UAV-BSs and users. If a user is predicted to move from its original serving UAV-BS's coverage area to a neighbor's, then the UAV-BS will hand such a user over to the appropriate neighbor. In our proposed approach, the *polynomial-fitting* method is adopted to predict a user's next location, where two independent polynomials will be fitted along x- and y-axes.

B. Distributed Algorithm

In our GSDA, we propose to adopt the distributed algorithm in [28] to solve the optimization problem given by (16). Although the algorithm proposed in [28] is primarily for the convex optimization problem, we will justify that it can also converge for the underlying nonconvex optimization formulated by (16) later on. Define three indicator vectors $\vec{\mathbf{1}}_1 \overset{\text{def}}{=} [1,0,0]^{\mathrm{T}} \in \mathbb{R}^{3\times 1}$, $\vec{\mathbf{1}}_2 \overset{\text{def}}{=} [0,1,0]^{\mathrm{T}} \in \mathbb{R}^{3\times 1}$, and $\vec{\mathbf{1}}_3 \overset{\text{def}}{=} [0,0,1]^{\mathrm{T}} \in \mathbb{R}^{3\times 1}$. Then, let

$$\vec{W}_k^q(t) \overset{\text{def}}{=} \vec{\mathbf{1}}_q^{\mathrm{T}} \tilde{L}(t) \in \mathbb{R}^{1\times J}, \quad q=1,2,3, \quad t \in \left[t_n, t_{n+1}\right). \tag{17}$$

Equation (16) can be optimized using an *iterative Gibbs sampler* such that

$$\min_{\vec{\mathbf{1}}_q^{\mathrm{T}}\tilde{L}(t)} \sum_{j=1}^{J} F_j\big(\tilde{L}(t)\big)$$
$$\text{s.t.} \begin{cases} \vec{W}_k^{q'}(t) = \begin{cases} \vec{\mathbf{1}}_{q'}^{\mathrm{T}} \tilde{\mathcal{L}}_k(t_n), & \text{if } q' < q \\ \vec{\mathbf{1}}_{q'}^{\mathrm{T}} \tilde{\mathcal{L}}_{k-1}(t_n), & \text{if } q' > q \end{cases} \\ \vec{W}_k^{q''}(t) = \begin{cases} \vec{\mathbf{1}}_{q''}^{\mathrm{T}} \tilde{\mathcal{L}}_k(t_n), & \text{if } q'' < q \\ \vec{\mathbf{1}}_{q''}^{\mathrm{T}} \tilde{\mathcal{L}}_{k-1}(t_n), & \text{if } q'' > q \end{cases} \\ \Psi_i(t) \geq P_{\min} \quad \forall i \\ h_{\max} \geq h_j^D(t) \geq h_{\min} \quad \forall j \end{cases} \tag{18}$$

where $q' \overset{\text{def}}{=} [q \bmod 3]+1$, $q'' \overset{\text{def}}{=} [(q+1) \bmod 3]+1$, $\vec{W}_k^{q'}(t)$ and $\vec{W}_k^{q''}(t)$ are defined by (17), the sequential order of q can be $q = 1 \to 2 \to 3 \to 1 \to 2 \to 3 \to \cdots$, and $t \in [t_n, t_{n+1})$.

Now we will design a distributed algorithm (as a centralized coordination mechanism for dispatching all UAVs at every iteration is not feasible in practice), which is modified from [28], to solve the optimization formulated by (18) above. Because our proposed algorithm is distributive, we can simply *decouple* the overall optimization objective formulated by (18) into J separate subproblems as given by

$$\min_{\vec{\mathbf{1}}_q^{\mathrm{T}}\tilde{L}(t)} F_j\big(\tilde{L}(t)\big)$$
$$\text{s.t.} \begin{cases} \vec{W}_{k,j}^{q'}(t) = \begin{cases} \vec{\mathbf{1}}_{q'}^{\mathrm{T}} \tilde{\mathcal{L}}_k(t_n), & \text{if } q' < q \\ \vec{\mathbf{1}}_{q'}^{\mathrm{T}} \tilde{\mathcal{L}}_{k-1}(t_n), & \text{if } q' > q \end{cases} \\ \vec{W}_{k,j}^{q''}(t) = \begin{cases} \vec{\mathbf{1}}_{q''}^{\mathrm{T}} \tilde{\mathcal{L}}_k(t_n), & \text{if } q'' < q \\ \vec{\mathbf{1}}_{q''}^{\mathrm{T}} \tilde{\mathcal{L}}_{k-1}(t_n), & \text{if } q'' > q \end{cases} \\ \Psi_i(t) \geq P_{\min} \quad \forall i \\ h_{\max} \geq h_j^D(t) \geq h_{\min} \quad \forall j \end{cases} \tag{19}$$

where q' and q'' are defined below (18), $\vec{W}_{k,j}^{q'}(t)$ and $\vec{W}_{k,j}^{q''}(t)$ are identical to $\vec{W}_k^{q'}(t)$ and $\vec{W}_k^{q''}(t)$ defined by (17), $\vec{W}_{k,j}^{q}(t)$ is the update of $\vec{W}_k^q(t)$ contributed by UAV j at the iteration k, $j = 1,2,\ldots,J$, and $t \in [t_n, t_{n+1})$. Besides, we propose to invoke a *hierarchical procedure* to update $\vec{W}_{k,j}^{q}(t)$, $q = 1,2,3$ as follows. Given UAV j at the iteration k, we will execute a subloop to update $\vec{W}_{k,j}^{q}(t)$ multiple times at UAV j with a subloop (or subiteration) index $\tau = 1,2,\ldots,\mathcal{K}$ ($\mathcal{K}$ subiterations totally) for each $q = 1,2,3$. That is, we keep updating $\vec{W}_{k,j}^{q}(t)$ for $\mathcal{K}$ times at the iteration k before starting to update $\vec{W}_{k,j}^{q'}(t)$ where $q' \overset{\text{def}}{=} [q \bmod 3]+1$ and so on. Thus, we can denote the τth intermediate update of $\vec{W}_{k,j}^{q}(t)$ calculated locally by UAV j for the iteration k and $t \in [t_n, t_{n+1})$ by $\vec{w}_{k,\tau,j}^{q}(t_n)$, where the initial condition is set to be $\vec{w}_{k,0,j}^{q}(t_n) \overset{\text{def}}{=} \vec{\mathbf{1}}_q^{\mathrm{T}} \tilde{\mathcal{L}}_k(t_n) \quad \forall j$ and $q = 1,2,3$.

However, the aforementioned distributed optimization problem formulated by (19) is different from the global optimization problem given by (18). Obviously, a major problem arises as one needs to *merge* the local updates $\vec{W}_{k,j}^{q}(t)$ contributed by all UAVs $j = 1,2,\ldots,J$ to the ultimate updates of $\vec{W}_k^q(t)$, for $q = 1,2,3$, which are the actual solutions J UAVs seek at every iteration. Motivated by [28], we design

a sophisticated merging procedure in this regard. Denote $\vec{v}^{q}_{k,\tau,j}(t_n) \in \mathbb{R}^{1\times J}$ by the weighted average of $\vec{w}^{q}_{k,\tau,j}(t_n)$'s over UAV j and its neighbors, which UAV j needs to calculate at the beginning of the subiteration τ. It is given by

$$\vec{v}^{q}_{k,\tau,j}(t_n) \overset{\text{def}}{=} \sum_{i\in\{j\}\cup\mathbb{N}_j} \xi_{ji} \times \vec{w}^{q}_{k,\tau,i}(t_n) \tag{20}$$

where $\mathbb{N}_j$ is defined by (8) which includes all neighbors being capable of exchanging update information with UAV j. The *default coefficients* ξ_{ji}'s are nonnegative constants ($\xi_{ji} \geq 0$, for all i, j) and should be determined heuristically. If we write

$$\tilde{L}(t) \overset{\text{def}}{=} \tilde{\Upsilon} \overset{\text{def}}{=} \left[\vec{\Omega}_1^{\mathrm{T}}, \vec{\Omega}_2^{\mathrm{T}}, \vec{\Omega}_3^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{3\times J} \tag{21}$$

where $\vec{\Omega}_m \in \mathbb{R}^{1\times J}$, $m = 1, 2, 3$, we may have $F_j(\tilde{L}(t)) = F_j(\tilde{\Upsilon})$. The gradient of $F_j(\tilde{\Upsilon})$ with respect to $\vec{\Omega}_q$, $q = 1, 2, 3$ can be expressed by $\nabla_q F_j(\tilde{\Upsilon})$. Define

$$\tilde{\Xi}^{q}_{k,\tau,j} \overset{\text{def}}{=} \left[\vec{\varsigma}_1^{\mathrm{T}}, \vec{\varsigma}_2^{\mathrm{T}}, \vec{\varsigma}_3^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{3\times J} \tag{22}$$

where

$$\vec{\varsigma}_q \overset{\text{def}}{=} \vec{v}^{q}_{k,\tau,j}(t_n) \tag{23}$$

$$\vec{\varsigma}_{q'} \overset{\text{def}}{=} \begin{cases} \vec{w}^{q'}_{k,\mathcal{K},j}(t_n), & \text{if } q' < q \\ \vec{w}^{q'}_{k-1,\mathcal{K},j}(t_n), & \text{if } q' > q \end{cases} \tag{24}$$

$$\vec{\varsigma}_{q''} \overset{\text{def}}{=} \begin{cases} \vec{w}^{q''}_{k,\mathcal{K},j}(t_n), & \text{if } q'' < q \\ \vec{w}^{q''}_{k-1,\mathcal{K},j}(t_n), & \text{if } q'' > q \end{cases} \tag{25}$$

where q' and q'' are defined below (18). Now each UAV j will use the weighted average $\vec{v}^{q}_{k,\tau,j}(t_n)$ given by (20) to update $\vec{w}^{q}_{k,\tau+1,j}(t_n)$ as follows:

$$\vec{w}^{q}_{k,\tau+1,j}(t_n) \overset{\text{def}}{=} \mathcal{P}_{\mathbb{W}_j}\left[\vec{v}^{q}_{k,\tau,j}(t_n) - \alpha^{q}_{\tau,j}\nabla_q F_j\left(\tilde{\Xi}^{q}_{k,\tau,j}\right)\right] \tag{26}$$

where $\alpha^{q}_{\tau,j} > 0$ specifies the step size, $\nabla_q F_j(\tilde{\Xi}^{q}_{k,\tau,j})$ is defined above, and $\mathcal{P}_{\mathbb{W}_j}[]$ denotes the Euclidean projection of the enclosed argument onto the set $\mathbb{W}_j$ such that

$$\mathbb{W}_j \overset{\text{def}}{=} \Big\{\vec{w}^{q}_{k,\tau,j}(t_n) \Big| \vec{W}^{q'}_{k,j}(t_n) \text{ and } \vec{W}^{q''}_{k,j}(t_n) \text{ are given} \quad \Psi_i(t_n) \geq P_{\min} \quad \forall i \in \mathbb{U}_j(t_n) \quad h_{\max} \geq h^{D}_{j}(t_n) \geq h_{\min}\Big\}. \tag{27}$$

When UAVs apply our proposed distributed algorithm to produce the solution to the problem formulated by (19), each UAV j should apply (20) and (26) alternatively for each subiteration τ during the iteration k. To prove the *asymptotic convergence* such that $\lim_{\tau\to\infty} \vec{w}^{q}_{k,\tau,j}(t_n) = \vec{w}^{q}_{k,\tau}(t_n) \quad \forall j$, $q = 1, 2, 3$ (all local updates will converge to the same vector asymptotically), we need to have the following assumptions.

Assumption 1: The set $\mathbb{W}_j$ is closed and convex.

Define a graph $(\mathbb{J}, \mathbb{E})$, where $\mathbb{J}$ and $\mathbb{E}$ are the vertex and edge sets such that

$$\mathbb{E} \overset{\text{def}}{=} \left\{(i, j) : i \in \mathbb{N}_j, j \in \mathbb{J}\right\} \tag{28}$$

and $\mathbb{J}$ is the set of UAV indices.

Algorithm 1 Our Proposed GSDA

Input: discrete time instant t_n, initial UAV-BSs' location matrix $\tilde{\mathcal{L}}_0(t_n)$, and ground users' location matrix $\tilde{\ell}(t_n)$
Output: UAV-BSs' sub-optimal location matrix $\tilde{\mathcal{L}}_K(t_n)$

for $k \leftarrow 1$ to K **do**
  for $q \leftarrow 1$ to 3 **do**
    Assign values to $\vec{W}^{q'}_{k}(t_n)$ and $\vec{W}^{q''}_{k}(t_n)$ according to Eq. (19).
    /* Invoke the distributed algorithm to obtain the sub-optimal $\vec{W}^{q}_{k}(t_n)$ */
    $\forall j \in \mathbb{J}$, initialize $\vec{w}^{q}_{k,0,j}(t_n) \overset{\text{def}}{=} \vec{\mathbf{1}}^{T}_{q}\,\tilde{\mathcal{L}}_k(t_n)$
    for $\tau \leftarrow 1$ to $\mathcal{K}$ **do**
      for $j \leftarrow 1$ to J **do**
        $\vec{v}^{q}_{k,\tau,j}(t_n) = \sum_{i\in\{j\}\cup\mathbb{N}_j} \xi_{ji} \times \vec{w}^{q}_{k,\tau,j}(t_n)$;
        $\vec{w}^{q}_{k,\tau+1,j}(t_n) = \mathcal{P}_{\mathbb{W}_j}\Big[\vec{v}^{q}_{k,\tau,j}(t_n) - \alpha^{q}_{\tau,j} \times \nabla_q F_j\Big(\tilde{\Xi}^{q}_{k,\tau,j}\Big)\Big]$.
    Update the q-th row of $\tilde{\mathcal{L}}_k(t_n)$

Assumption 2: The graph $(\mathbb{J}, \mathbb{E})$ is strongly connected at any iteration.

Assumption 2 ensures that the update of every UAV influences the updates contributed by all other UAVs directly or indirectly.

Assumption 3: For $j \in \mathbb{J}$:

1) $\xi_{ji} \geq 0$, and $\xi_{ji} = 0$ if $i \notin \mathbb{N}_j$ and $i \neq j$;
2) $\sum_{i=1}^{J} \xi_{ji} = 1$;
3) there exists a scalar $\varepsilon \in (0, 1)$ such that $\xi_{ji} \geq \varepsilon$ if $i \in \mathbb{N}_j$;
4) $\sum_{j=1}^{J} \xi_{ji} = 1$.

It can be proven that all the assumptions above are satisfied by our proposed approach (the distributed algorithm) in this work.

Theorem 1: Assume that Assumptions 1–3 are valid. If the step size $\{\alpha^{q}_{\tau,j}\}$ satisfies $\lim_{\tau\to\infty} \alpha^{q}_{\tau,j} = 0$. Then $\lim_{\tau\to\infty} \mathrm{E}[\|\vec{w}^{q}_{k,\tau,j}(t_n) - \vec{w}^{q}_{k,\tau,\text{ave}}(t_n)\|] = 0$ for all $j \in \mathbb{J}$, where $\vec{w}^{q}_{k,\tau,\text{ave}}(t_n) \overset{\text{def}}{=} (1/J)\sum_{j=1}^{J} \vec{w}^{q}_{k,\tau,j}(t_n)$.

Theorem 1 has been proved in [29]. This theory reveals that even if $F_j(\tilde{\Upsilon})$ is not convex, $\vec{w}^{q}_{k,\tau,j}(t_n)$ can also converge as $\tau \to \infty \quad \forall j \in \mathbb{J}$. Our proposed GSDA is introduced as Algorithm 1.

C. Computational-Complexity Analysis

The majority of the computational burden in the GSDA is on the calculation of the gradient vectors $\nabla_q F_j(\tilde{\Upsilon})$, $q = 1, 2, 3$. Thus, we can approximate the computational complexity of the GSDA by the complexity required to create the above-mentioned gradient vectors. Once the UAV-aided wireless network is configured like Fig. 2, one can set up the explicit Formulas for $\nabla_q F_j(\tilde{\Upsilon})$, $q = 1, 2, 3$. Then, UAV j will plug $\tilde{\Upsilon} = \tilde{\Xi}^{q}_{k,\tau,j}$ in $\nabla_q F_j(\tilde{\Upsilon})$ at each intermediate subiteration τ of a certain iteration k. For each UAV j, the formulas of the gradient vectors $\nabla_q F_j(\tilde{\Upsilon})$ for $q = 1$, $q = 2$, and $q = 3$ are quite similar to each other. Consequently, without loss of generality, we just discuss the complexity for $q = 1$, which can

TABLE I
NUMBERS OF OPERATIONS FOR CALCULATING $\nabla_1 F_j(\tilde{\Xi}^1_{k,\tau,j})$

Operation type	Number of operations
Sum	$(69\,\lvert\mathbb{N}_j\rvert^2 + 184\,\lvert\mathbb{N}_j\rvert + 115)\,\lvert\mathbb{U}_j(t_n)\rvert$
Product	$(102\,\lvert\mathbb{N}_j\rvert^2 + 275\,\lvert\mathbb{N}_j\rvert + 170)\,\lvert\mathbb{U}_j(t_n)\rvert$
Exponential	$(51\,\lvert\mathbb{N}_j\rvert^2 + 137\,\lvert\mathbb{N}_j\rvert + 86)\,\lvert\mathbb{U}_j(t_n)\rvert$
Logarithm	$(12\,\lvert\mathbb{N}_j\rvert^2 + 32\,\lvert\mathbb{N}_j\rvert + 20)\,\lvert\mathbb{U}_j(t_n)\rvert$
Inverse trigonometric	$(6\,\lvert\mathbb{N}_j\rvert^2 + 16\,\lvert\mathbb{N}_j\rvert + 10)\,\lvert\mathbb{U}_j(t_n)\rvert$

be extended to both $q = 2$ and $q = 3$ later on. For $q = 1$, according to (21), we can write $\vec{\Omega}_1 \overset{\text{def}}{=} [\varpi_1, \varpi_2, \ldots, \varpi_J]$. Thus

$$\nabla_1 F_j\left(\tilde{\Upsilon}\right) = \left[\frac{\partial F_j\left(\tilde{\Upsilon}\right)}{\partial \varpi_1}, \frac{\partial F_j\left(\tilde{\Upsilon}\right)}{\partial \varpi_2}, \ldots, \frac{\partial F_j\left(\tilde{\Upsilon}\right)}{\partial \varpi_J}\right]^{\mathrm{T}}. \quad (29)$$

As UAV j has $|\mathbb{N}_j|$ neighbors and $|\mathbb{U}_j(t_n)|$ served users at the time instant t_n, the numbers of different kinds of operations involved in the calculation of $([\partial F_j(\tilde{\Xi}^1_{k,\tau,j})]/[\partial \Omega_\iota])$, $1 \le \iota \le J$ are listed in Table I. As previously discussed, one can apply the listed numbers in Table I to closely approximate the computational complexities for $q = 2$ and $q = 3$ as well. The practical arithmetic-execution benchmark criterion can be found in [30], which lists the required numbers of CPU-clock cycles to execute those operations addressed in Table I according to the recent and current generations of computer architectures. For the worst-case study, it requires ten cycles to execute each of "sum" and "product" operations, 50 cycles to execute "exponential" function, and 1000 cycles to execute each of "logarithm" and inverse "trigonometric" functions. Take the Intel Xeon processor as an example, whose CPU speed is 2.8 GHz. According to the configured UAV-aided wireless network in Fig. 2, each UAV will have up to four neighbors and 10 served users on average. Consequently, when a UAV undertakes our proposed GSDA using an Intel Xeon processor, it needs to take 0.0023 s per subiteration τ.

D. Transmission Overhead Analysis

In our proposed network configuration, assume that there are $J = u^2$ UAVs ($u > 1$) serving I ground users in both centralized and distributed optimization schemes, where $u \in \mathbb{N}^+$ indicates the numbers of UAVs in a single row or column of the entire coverage area, and both optimization schemes optimize UAVs' locations for maximizing the channel capacity according to the location information of these I served users in every iteration. On average, each UAV collects locations of I/J ground users. For the centralized optimization scheme, the hub UAV-BS connects all other UAV-BSs within the network. At each iteration, all other UAV-BSs transmit their own positions and the locations of their served ground users to the hub; it requires $(J-1)(I/J+1)$ transmissions in total. The UAV hub then employs the received location information to generate all optimal UAV-BSs' locations and distributes them to the corresponding UAV-BSs, which requires $J-1$ transmissions. Overall, the total number of transmissions is $\rho_C \overset{\text{def}}{=} (J-1)(I/J+1) + (J-1) = I + 2J - I/J - 2$.

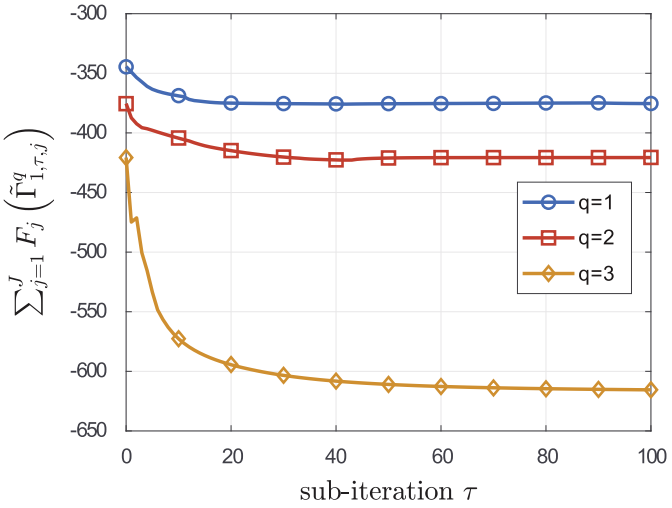

Fig. 6. Learning curves of $\sum_{j=1}^{J} F_j(\tilde{\Gamma}^q_{1,\tau,j})$ versus the number of subiterations τ for different spatial dimensions q.

On the other hand, for our proposed distributed optimization algorithm, at every iteration, each UAV-BS is only required to connect with its neighbor UAV-BSs for acquiring their positions, then the location optimization is independently and asynchronously implemented in local. Specifically, there are $(u-1)^2$ UAVs which have four neighbors, $4(u-1)$ UAVs which have three neighbors, and four UAVs which have two neighbors. The information transmission only occurs before and after each iteration. Consequently, the total number of transmissions is $\rho_D \overset{\text{def}}{=} (u-1)^2 \times 4 + 4(u-1) \times 3 + 4 \times 2 = 4u^2 + 4u$. The transmission overhead (signaling overhead) ratio between the distributed and centralized optimization scheme is, thus, given by

$$\frac{\rho_D}{\rho_C} = \frac{4u^2 + 4u}{I + 2J - I/J - 2} = \frac{4J^2 + 4J\sqrt{J}}{(I-2)J + 2J^2 - I}. \quad (30)$$

When $I \gg J$, $(\rho_D/\rho_C) < 1$ or $\rho_D < \rho_C$. For instance, in a UAV-assisted network consisting of $J = 9$ UAV-BSs serving $I = 90$ users. Thus, each UAV serves ten users on average. The transmission (signaling) overhead required by the centralized optimization scheme is $\rho_C = 96$, while that required by the distributed optimization scheme is $\rho_D = 48$. In this scenario, our proposed GSDA can significantly reduce the required transmission (signaling) overhead in comparison with the centralized optimization scheme.

IV. SIMULATION

In this section, our proposed GSDA is evaluated by simulations and the learning curves of the proposed GSDA are delineated by Fig. 6. Our proposed novel GSDA is also compared with the existing distributed genetic algorithm in Fig. 8. Moreover, the users' mobility, update asynchronization, and inaccurate positioning are also taken into account to justify the robustness of our proposed GSDA. The corresponding results are shown in Figs. 11, 12, and 14, respectively.

TABLE II
LIST OF NUMERICAL VALUES OF ENVIRONMENT PARAMETERS

Environment Parameters	a	b	η_{LoS}	η_{NLoS}
Suburban	4.88	0.43	0.1	21
Urban	9.61	0.16	1	20
Dense Urban	12.08	0.11	1.6	23
High-rise Urban	27.23	0.08	2.3	34

A. Learning Curves

In our simulations, we set the size of a zone on the ground covered by each of nine UAVs is 600 m ×600 m while 100 users are randomly distributed over the entire area (consisting of nine zones thereby) at time t_n. Prior to executing our proposed GSDA, every UAV is located right above the center of its served zone with a height of 100 m. Note that every UAV cannot exceed its zone boundaries and its height $h_j^D(t_n)$ has to be within 30 m$= h_{\min} \leq h_j^D(t_n) \leq h_{\max} =$ 100 m. Besides, each UAV's transmitting power is set to be $P_T = 30$ dBm, the minimum required received power is $P_{\min} = -70$ dBm, the carrier frequency of the transmitted signal is set to be $f = 3.5$ GHz, and the environmental noise power is $\mathcal{N} = -100$ dBm. Moreover, to obtain the environment parameters a and b in (3), we employ the values of ηLoS and ηNLoS [characterized by the ATG pathloss model given by (1)] presented by [7] and [26] for different environments. The numerical values of the relevant parameters are also listed in Table II. Here, we choose the suburban environment for simulation.

Now we determine the numerical values of the crucial parameters in the GSDA. Assume that ξ_{ji} in (20) is fixed for all subiterations τ. In order to manifest the neighborhood relationship between UAVs shown by Fig. 2 and comply with Assumption 3, we set the values of ξ_{ji} as listed by Table III.

Next, we would like to show how to adjust the step sizes $\alpha_{\tau,j}^q$ in Algorithm 1. Define

$$\tilde{\Gamma}_{k,\tau,j}^q \stackrel{\text{def}}{=} \left[\vec{\chi}_1^{\mathrm{T}}, \vec{\chi}_2^{\mathrm{T}}, \vec{\chi}_3^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{3\times J} \tag{31}$$

where

$$\vec{\chi}_q \stackrel{\text{def}}{=} \vec{w}_{k,\tau,j}^q(t_n) \tag{32}$$

$$\vec{\chi}_{q'} \stackrel{\text{def}}{=} \begin{cases} \vec{w}_{k,\mathcal{K},j}^{q'}(t_n), & \text{if } q' < q \\ \vec{w}_{k-1,\mathcal{K},j}^{q'}(t_n), & \text{if } q' > q \end{cases} \tag{33}$$

$$\vec{\chi}_{q''} \stackrel{\text{def}}{=} \begin{cases} \vec{w}_{k,\mathcal{K},j}^{q''}(t_n), & \text{if } q'' < q \\ \vec{w}_{k-1,\mathcal{K},j}^{q''}(t_n), & \text{if } q'' > q \end{cases} \tag{34}$$

where q' and q'' are defined below (18) and $\vec{w}_{k,\tau,j}^q(t_n)$ is defined above (20). We propose to set the step sizes according to (35)–(37), shown at the bottom of the next page. In our simulation, we set $s_1 = 1000$, $s_2 = 1000$, and $s_3 = 300$. Without loss of generality, we observe the performance of our proposed GSDA for an arbitrary iteration, say $k = 1$. The evolutions (learning curves) of the objective function $\sum_{j=1}^J F_j(\tilde{\Gamma}_{1,\tau,j}^q)$ for $q = 1, 2, 3$ are delineated in Fig. 6 at the time instant t_n. In Fig. 6, one can find that the objective function decreases over subiteration τ as we desire for each dimension q. Moreover, all three learning curves converge. According to Theorem 1, $\vec{w}_{1,\tau,j}^q(t_n)$, $j \in \mathbb{J}$ will converge to $\vec{w}_{1,\tau,\text{ave}}^q(t_n)$, which means that the μth component of $\vec{w}_{1,\tau,j}^q(t_n)$, $j \in \mathbb{J}$ for $1 \leq \mu \leq J$ will converge to a certain number. As $\vec{w}_{1,\tau,j}^q(t_n) \in \mathbb{R}^{1\times J}$, its μth component can be written as $u_{1,\tau,j}^{q,\mu}(t_n) \stackrel{\text{def}}{=} \vec{w}_{1,\tau,j}^q(t_n) \times \vec{\mathbf{e}}_\mu$; the $J \times 1$ vector $\vec{\mathbf{e}}_\mu$ is an indicator vector which has a "1" at its μth entry and its all other entries are zero. Define $\vec{U}_{1,\tau,\mu}^q(t_n) \stackrel{\text{def}}{=} [u_{1,\tau,1}^{q,\mu}(t_n), u_{1,\tau,2}^{q,\mu}(t_n), \ldots, u_{1,\tau,J}^{q,\mu}(t_n)] \in \mathbb{R}^{1\times J}$. We will depict the mean bars with a confidence interval for the vectors $\vec{U}_{1,\tau,\mu}^q(t_n)$ versus τ for $q = 1$, $q = 2$, and $q = 3$. In Fig. 7, we plot such bar-curves for $\mu = 2$, $\mu = 5$, and $\mu = 9$ to illustrate the optimal-location convergence for UAVs 2, 5, and 9. According to Fig. 7, the standard deviations (red-bar lengths) of $\vec{U}_{1,\tau,\mu}^q(t_n)$ decrease as the subiteration number τ increases for $q = 1, 2, 3$ and $\mu = 2, 5, 9$, while the corresponding means also converge. In other words, the new locations of UAVs resulting from our proposed GSDA will converge eventually to maximize the overall channel capacity. On the other hand, we have also tested the performance of our proposed GSDA for hexagonal zones. In such network configuration, each UAV-BS tends to have more neighbors than the configuration of square zones. Therefore, the corresponding objective function to hexagonal zones is much more complicated and causes very slow convergence of our proposed GSDA for finding the optimal UAV locations. Then, we compare the learning curves of the objective function $\sum_{j=1}^J F_j(\tilde{L}_k(t_n))$ over iterations k resulting from our proposed GSDA and the distributed genetic algorithm (refer to Section IV-B for details) in Fig. 8 for the time instant t_n. According to Fig. 8, the distributed genetic algorithm (GA) obviously requires much more iterations than our proposed GSDA. Besides, the ultimate channel capacity achieved by the GSDA is higher than that achieved by the distributed GA. As a result, our proposed GSDA greatly outperforms the distributed GA in terms of convergence speed and optimal capacity.

The initial and terminal (optimal) locations along with the trajectories of all nine UAVs resulting from our proposed GSDA are depicted by Figs. 9 and 10, where it has been run for $K = 10$ iterations, each of which includes $\mathcal{K} =$ 100, 100, 500 subiterations for $q = 1, 2, 3$, respectively. Note that all users are assumed to be static and their locations are also marked in Figs. 9 and 10.

The stopping criterion of the GSDA is introduced as follows. For UAV j, we consider that the GSDA converges if and only if there exists $\mathcal{H}_j^q \in \mathbb{N}^+$, where $|F_j(\tilde{\Gamma}_{k,\tau,j}^q) - F_j(\tilde{\Gamma}_{k,\tau-1,j}^q)| < 0.01$ and $\|\vec{w}_{k,\tau,j}^q(t_n) - \vec{w}_{k,\tau-1,j}^q(t_n)\| < 0.01$ as $\tau > \mathcal{H}_j^q$. Hence, $\mathcal{H}_j^q$ represents the number of iterations required by the GSDA to converge for UAV j and q. As each UAV invokes the GSDA to search its next optimal location within two consecutive time instants, say between t_n and t_{n+1}, the computation time of the GSDA should be short enough to accommodate such a time duration. We calculate the average numbers of iterations according to the aforementioned stopping criterion for all UAVs to optimize with different selections of step

TABLE III
NUMERICAL VALUES OF ξ_{ji}

j \ i	1	2	3	4	5	6	7	8	9
1	0.5	0.25	0	0.25	0	0	0	0	0
2	0.25	0.5	0.125	0	0.125	0	0	0	0
3	0	0.125	0.6875	0	0	0.1875	0	0	0
4	0.25	0	0	0.5	0.125	0	0.125	0	0
5	0	0.125	0	0.125	0.5	0.125	0	0.125	0
6	0	0	0.1875	0	0.125	0.5	0	0	0.1875
7	0	0	0	0.125	0	0	0.6875	0.1875	0
8	0	0	0	0	0.125	0	0.1875	0.5	0.1875
9	0	0	0	0	0	0.1875	0	0.1875	0.625

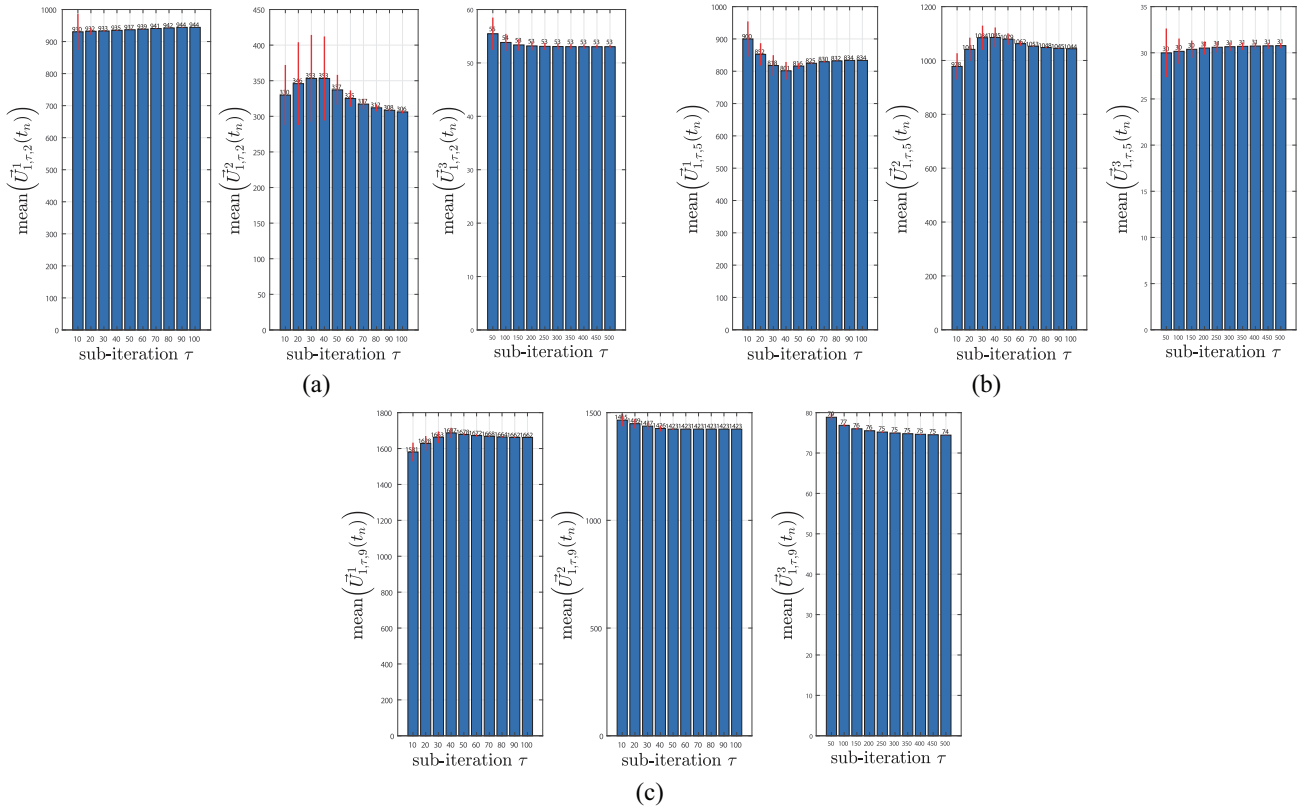

Fig. 7. Mean-bars with confidence interval of the updated coordinates in three dimensions by the GSDA (from left to right: $q = 1, 2, 3$) for (a) UAV 2, (b) UAV 5, and (c) UAV 9.

sizes subject to different choices of (s_1, s_2, s_3) in (35)–(37). From simulation, the average numbers of iterations according to the aforementioned stopping criterion for all UAVs in different sets of step sizes are listed in Table IV, where $\mathcal{H}^q_{\text{ave}} \overset{\text{def}}{=} 1/J \sum_{j=1}^{J} \mathcal{H}^q_j$. According to Table IV, the total number of iterations, namely, $\mathcal{H}^1_{\text{ave}} + \mathcal{H}^2_{\text{ave}} + \mathcal{H}^3_{\text{ave}}$ ranges from 263 to 749 in each iteration k. As illustrated by Fig. 8, the learning curve of the GSDA converges at $k = 2$, which means that each UAV only needs to take 1.21 to 3.45 s (according to the computational-complexity analysis stated in Section III-C) to run the GSDA at the time instant t_n. This computation time is short enough in practice.

$$\alpha^1_{\tau,j} \overset{\text{def}}{=} \begin{cases} s_1/\tau^2, & \text{if } \tau > 1 \text{ and} \left| F_j\left(\tilde{\Gamma}^1_{k,\tau,j}\right) - F_j\left(\tilde{\Gamma}^1_{k,\tau-1,j}\right) \right| < 0.01 \\ s_1, & \text{otherwise} \end{cases} \tag{35}$$

$$\alpha^2_{\tau,j} \overset{\text{def}}{=} \begin{cases} s_2/\tau^2, & \text{if } \tau > 1 \text{ and} \left| F_j\left(\tilde{\Gamma}^2_{k,\tau,j}\right) - F_j\left(\tilde{\Gamma}^2_{k,\tau-1,j}\right) \right| < 0.01 \\ s_2, & \text{otherwise} \end{cases} \tag{36}$$

$$\alpha^3_{\tau,j} \overset{\text{def}}{=} s_3/\tau \tag{37}$$

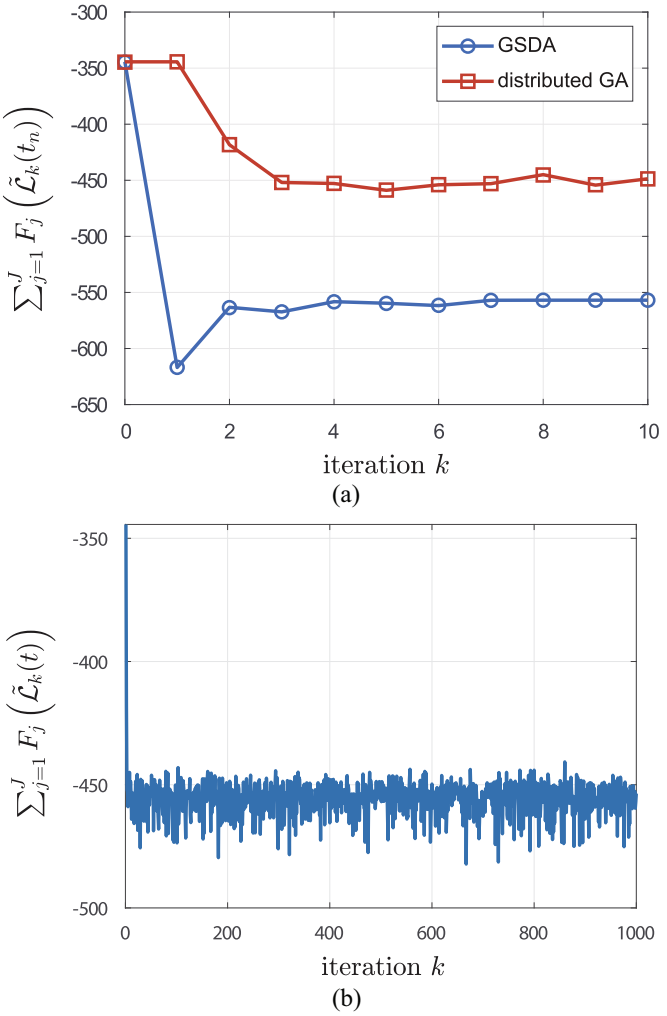

Fig. 8. Performance comparison between the GSDA and the distributed genetic algorithm (distributed GA): (a) learning curves for GSDA and distributed GA over ten iterations and (b) learning curve for distributed GA over a 1000 iterations.

TABLE IV
REQUIRED NUMBERS OF ITERATIONS FOR DIFFERENT STEP SIZES

s_1	$\mathcal{H}^1_{\text{ave}}$	s_2	$\mathcal{H}^2_{\text{ave}}$	s_3	$\mathcal{H}^3_{\text{ave}}$
1500	47	1500	61	1000	211
1000	87	1000	53	500	163
500	84	500	105	300	183
100	183	100	196	100	370

B. Distributed Genetic Algorithm

Here, we modify the conventional genetic algorithm in [31] to design a *distributed genetic algorithm* here to solve the optimization problem formulated by (16). Different from the GSDA, the distributed genetic algorithm has to fix the locations of all other UAVs while optimizing the jth UAV's location. Algorithm 2 manifests our proposed distributed genetic algorithm (distributed GA), where there are K_{G} iterations totally (indexed by k), each of which has K'_{G} subiterations (indexed by k').

The learning curves for the objective function $\sum_{j=1}^{J} F_j(\tilde{L}_k(t))$ versus iteration k for both GSDA and distributed GA are exhibited in Fig. 8, where $K_{\text{G}} \overset{\text{def}}{=} 1000$, $K'_{\text{G}} \overset{\text{def}}{=} 5$, and the initial UAV-aided network configurations are identical for both algorithms. Further, we keep the other parameters' value of the network configuration as the same. According to Fig. 8(b), the learning curve for the distributed GA does not tend to converge. It is quite usual that the genetic algorithm cannot converge or converge very slowly on practical circumstances [32], [33], [34]. Besides, the minimum of the learning curve for the distributed GA is around $\sum_{j=1}^{J} F_j(\tilde{L}_k(t)) = -480$, which is much larger (worse) than that achieved by the GSDA.

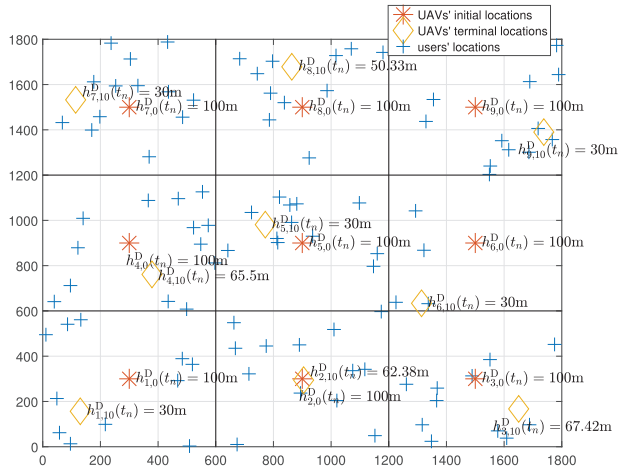

Fig. 9. Initial and terminal locations of UAV-BSs (for the iteration $k = 10$) and the locations of users.

Algorithm 2 Distributed Genetic Algorithm

Input: discrete time instant t_n, initial UAV-BSs' location matrix $\tilde{L}_0(t_n)$, and ground users' location matrix $\tilde{\ell}(t_n)$
Output: UAV-BSs' sub-optimal location matrix $\tilde{L}^{\text{G}}_{K_{\text{G}}}(t_n)$

1: **for** $k \leftarrow 1$ to K_{G} **do**
2: **for** $j \leftarrow 1$ to J **do**
3: Fix all UAV-BSs' locations except the j-th UAV-BS's location.
4: **for** $k' \leftarrow 1$ to K'_{G} **do**
5: Use the conventional genetic algorithm to find the UAV j's optimal location, say $\vec{L}^{\text{G}}_{k,k',j}(t_n)$ in the sub-iteration k'.
6: Choose the optimal location among $\left\{\vec{L}^{\text{G}}_{k,k',j}(t_n), 1 \le k' \le K'_{\text{G}}\right\}$ to minimize the objective equation $\sum_{j=1}^{J} F_j\big(\tilde{L}(t)\big)$ as the UAV-BS j's location update.

C. GSDA With Mobility Prediction

1) Location Predictor for Mobile Users: To deal with users' mobility, which will lead to a time-variant network topology and vary the channel capacity from time to time, we propose to apply a *polynomial-regression-based predictor* to track users' moving trajectories at the time instant t_n as follows. For a user i served by UAV j, the latter receives and records φ locations (at time instants $t \in \mathbb{T}$, where $\mathbb{T} \overset{\text{def}}{=} \{t_{n-\varphi+1}, t_{n-\varphi+2}, \ldots, t_n\}$) of user i prior to

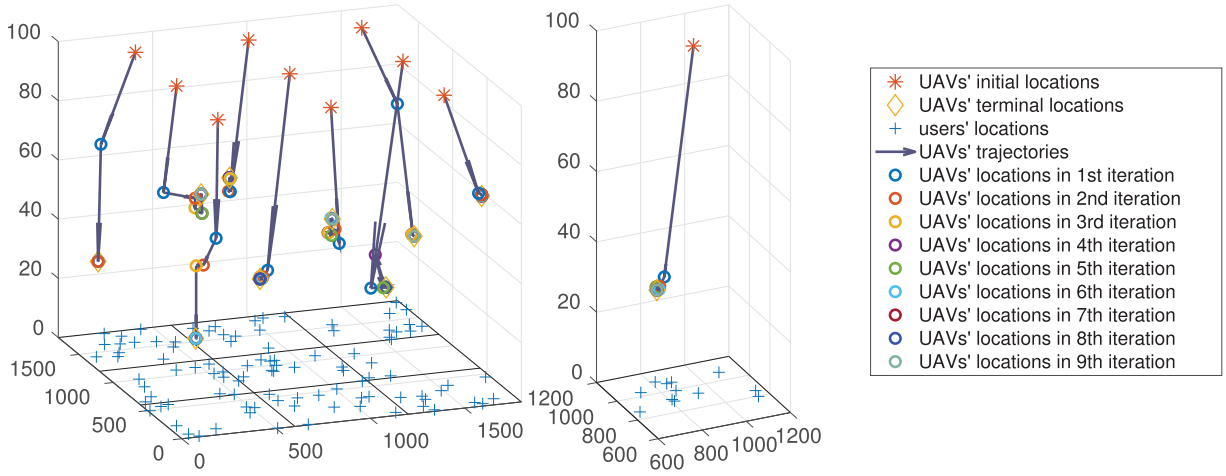

Fig. 10. Trajectories of all UAV-BSs (left) and UAV 5 (right) during an optimization interval (for the iteration $k = 10$). All users' locations are also shown here.

time t_n. For each spatial dimension q, UAV j can collect a set of user i's qth coordinates over time: $\mathbb{V}_q \overset{\text{def}}{=} \{\vec{\varkappa}_q \vec{\ell}_i(t_{n-\varphi+1}), \vec{\varkappa}_q \vec{\ell}_i(t_{n-\varphi+2}), \ldots, \vec{\varkappa}_q \vec{\ell}_i(t_n)\}$ for $q = 1, 2$, where $\vec{\varkappa}_1 \overset{\text{def}}{=} [1, 0]$ and $\vec{\varkappa}_2 \overset{\text{def}}{=} [0, 1]$. Thus, for each dimension q, UAV j can predict user i's future location as $\hat{\vec{\ell}}_i(t_{n+1})$ in the next time instant t_{n+1} simply by applying two separate polynomial-regressors over $\mathbb{V}_1$ and $\mathbb{V}_2$, respectively. Consequently, the predicted future coordinates $\hat{\vec{\ell}}_i(t_{n+1})$ of each user i can replace the current coordinates [columns of $\tilde{\ell}(t_n)$] in Algorithm 1 such that a *predicted users' future location matrix* $\hat{\tilde{\ell}}(t_{n+1}) \overset{\text{def}}{=} [\hat{\vec{\ell}}_1(t_{n+1}), \hat{\vec{\ell}}_2(t_{n+1}), \ldots, \hat{\vec{\ell}}_I(t_{n+1})]$ instead of the current users' location matrix $\tilde{\ell}(t_n)$ is employed by a "look-ahead" GSDA.

2) Numerical Evaluation: In our simulation, suppose that each user i's moving trajectory follows a parabolic trajectory subject to some deviation caused by noise. As mentioned in Section IV-C1, UAV j collects φ previous locations in terms of $\mathbb{V}_1$ and $\mathbb{V}_2$ for each of its served user i then applies two respective polynomial-regressors for generating $\hat{\vec{\ell}}_i(t_{n+1})$. In order to quantify the aforementioned deviation, we first assume that for any user i, all of its location coordinates $\vec{\ell}_i(t_{n+1})$ fall exactly along a parabolic curve. Then we artificially deviate these coordinates by adding random noise such that the noise-corrupted coordinates are $\vec{\ell}'_i(t) \overset{\text{def}}{=} \vec{\ell}_i(t) + \vec{r}_i(t) \in \mathbb{R}^{2\times 1}$, where the corresponding location-to-deviation ratio (LDR) can be characterized as follows:

$$\text{LDR}_i \overset{\text{def}}{=} \frac{1}{\varphi} \sum_{\iota=0}^{\varphi-1} 20 \log_{10} \left[\frac{\left\| \vec{\ell}_i(t_{n-\iota}) \right\|}{\|\vec{r}_i(t_{n-\iota})\|} \right] \text{dB}. \tag{38}$$

Two different LDRs are chosen in our simulation, i.e., $\text{LDR}_i = 0$ dB $\forall i$ and $\text{LDR}_i = -20$ dB $\forall i$. All other simulation parameters stay the same as those stated in Section IV-A. According to (11)–(16), the objective function $\sum_{j=1}^{J} F_j(\tilde{L}(t))$ is an implicit function of $\Lambda_{ij}(t)$, while according to (1)–(5), $\Lambda_{ij}(t)$ is also an implicit function of $\rho_j(t)$ and $\sigma_{ij}(t)$. Consequently, the objective function (the reciprocal of the overall channel capacity) is an implicit function of $\rho_j(t)$ and $\sigma_{ij}(t)$. Here, we use two different approaches to produce $\rho_j(t)$ and $\sigma_{ij}(t)$. First, we have $\rho_j^{\text{cur}}(t)$ and $\sigma_{ij}^{\text{cur}}(t)$, which result from the users' location matrix $\tilde{\ell}(t_{n+1})$ at time t_{n+1} and the optimal UAVs' location matrix $\tilde{L}(t_n)$ based on the current users' location matrix $\tilde{\ell}(t_n)$ at time t_n. Second, we have $\rho_j^{\text{pre}}(t)$ and $\sigma_{ij}^{\text{pre}}(t)$, which result from user's location matrix $\tilde{\ell}(t_{n+1})$ at time t_{n+1} and the optimal UAVs' location matrix based on the predicted users' location matrix $\hat{\tilde{\ell}}(t_{n+1})$ at time t_{n+1}. As users' mobility is considered, there tends to obvious mismatch between the optimal UAVs' locations (computed at time t_n but adjusted at time t_{n+1}) and the actual users' locations at t_{n+1} by the first approach. This mismatch drawback can be alleviated by the second approach since the users' future locations are predicted prior to optimization of UAVs' locations.

We plot the learning curves for the above-stated two approaches for different LDR conditions, namely, $\text{LDR}_i = \infty$ (noise free), 0, and -20 dB $\forall i$ in Fig. 11, where the first and second approaches are denoted by "cur." and "pre.," respectively. One can find that the objective function decreases and converges in the long run on all circumstances as illustrated by Fig. 11. That is, the GSDA can surely optimize the UAVs' locations as expected for different moving trajectories of users. According to Fig. 11, the performance resulting from the GSDA with our proposed polynomial-regression-based predictor for users' future locations is better than that resulting from the standard GSDA without any predictive mechanism for users' future locations. Meanwhile, the larger the LDR, the smaller (better) the optimal value resulting from the GSDA with the aforementioned predictive mechanism for users' future locations.

D. Asynchronous Updates in GSDA

As stated in Section III, we assume that each UAV can finish the τth subiteration at the same time, which is unrealistic in practice. Thus, we need to investigate such *asynchronous update effect* to see how robust our proposed GSDA would be under the update asynchronization. When the UAV j invokes the distributed algorithm to solve the optimization problem

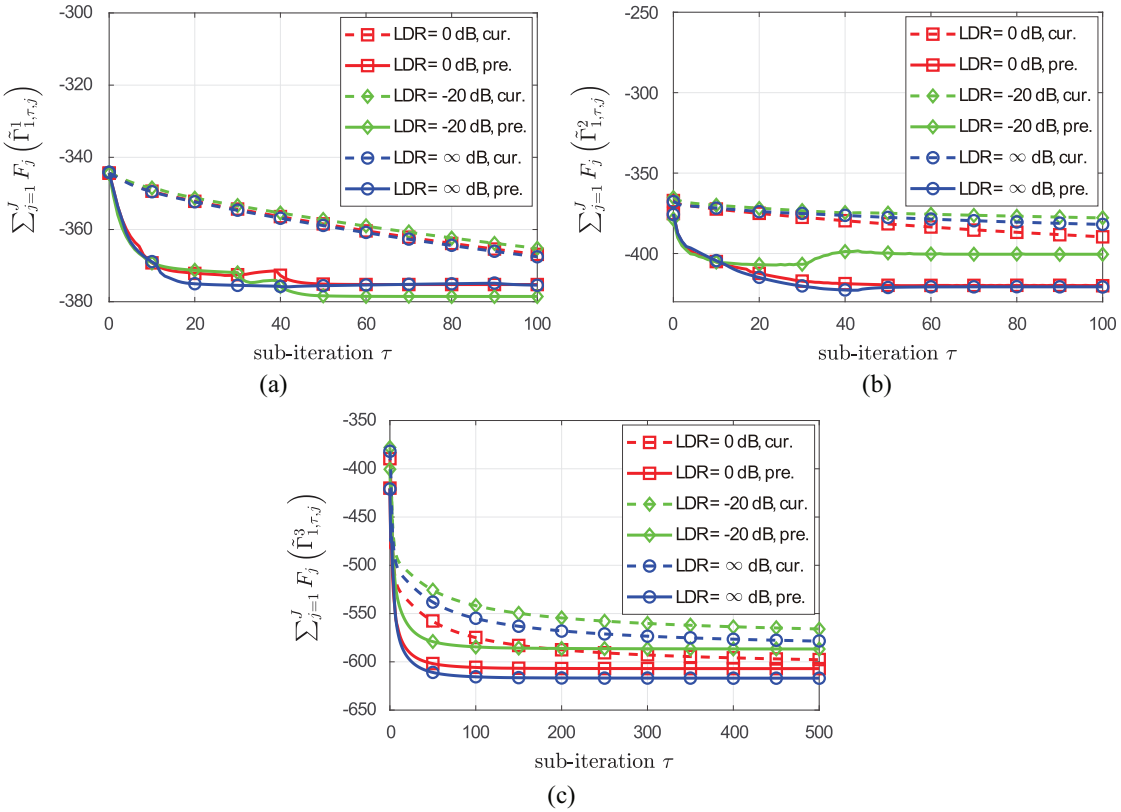

Fig. 11. Learning curves for the GSDA with (denoted by "pre.") and without (denoted by "cur.") prediction of users' future locations for (a) $q = 1$, (b) $q = 2$, and (c) $q = 3$.

formulated by (19), it needs to acquire the *time-stamped* information about its neighboring UAVs in $\bar{j} \in \mathbb{N}_j$. Without loss of generality, in a subiteration τ, as the UAV j inspects the time stamps associated with such received messages from its neighbors, it can halt the optimization task to adjust its own location if at least one time-stamp is out of date (the corresponding message has been received quite long ago from a neighbor).

Here, we perform the simulation to evaluate the aforementioned update asynchronization effect on the GSDA performance. In each subiteration τ, a UAV j does not carry out optimization occasionally, say 10% of subiterations. These occasional subiterations correspond to the situation that UAV j has not received the message from one of its neighbors for a while, and, thus, it halts the optimization task. Other simulation parameters stay the same as in Section IV-A. The comparison between the GSDA performances subject to update synchronization (UAVs update their locations at every subiteration) and update asynchronization (UAVs occasionally miss the updates of their locations) is demonstrated by Fig. 12 for $k = 1$ and $q = 1, 2, 3$. Obviously, the GSDA performance subject to the synchronous-update mode is superior to that subject to the asynchronous-update mode. The converging mean bars with confidence intervals for UAVs 2, 5, 9 and $q = 1, 2, 3$ can also be found for the asynchronous-update mode according to Fig. 13. Besides, as one can compare Fig. 7 with Fig. 13, it can be observed that these three UAVs' locations obtained by the synchronous and asynchronous modes are very similar to each other. In conclusion, our proposed GSDA can lead to satisfactory performance under occasional update asynchronization.

E. GSDA With Inaccurate Localization

At last, an important factor inherent in our proposed GSDA is also investigated here. In reality, the precise UAVs' 3-D locations are very difficult (rare) to obtain due to the limited precision of GPS and the wind effect. Therefore, localization (location) errors will often occur. In reality, when UAVs update the qth row of $\tilde{\mathcal{L}}_k(t_n)$ in Algorithm 1, the location matrix $\tilde{\mathcal{L}}_k(t_n)$ will be subjected to an additive error matrix, say $\tilde{E}_k^{\lambda}(t_n) \in \mathbb{R}^{3\times J}$, whose entries are independently and identically distributed uniform processes over the dynamic range of $[-\lambda, \lambda]$. In other words, $\tilde{\mathcal{L}}_k^{\lambda}(t_n) \overset{\text{def}}{=} \tilde{\mathcal{L}}_k(t_n) + \tilde{E}_k^{\lambda}(t_n)$, where $\tilde{\mathcal{L}}_k^{\lambda}(t_n)$ specifies the UAVs' "perturbed" location matrix from the desired location matrix $\tilde{\mathcal{L}}_k(t_n)$ by such location errors. In this section, the simulation parameters remain the same as in Section IV-A. Now let's modify the definition of LDR given by (38) as follows:

$$\text{LDR}_k(t_n) \overset{\text{def}}{=} 20\log_{10}\left[\frac{\|\tilde{\mathcal{L}}_k(t_n)\|_F}{\|\tilde{E}_k^{\lambda}(t_n)\|_F}\right] \text{dB} \tag{39}$$

where $\|\|_F$ denotes of the *Frobenius norm* [35]. We take the iteration $k = 1$ and an arbitrary time instant t_n for illustration here. Three LDRs, namely, $\text{LDR}_1(t_n) = \infty$ dB (error free),

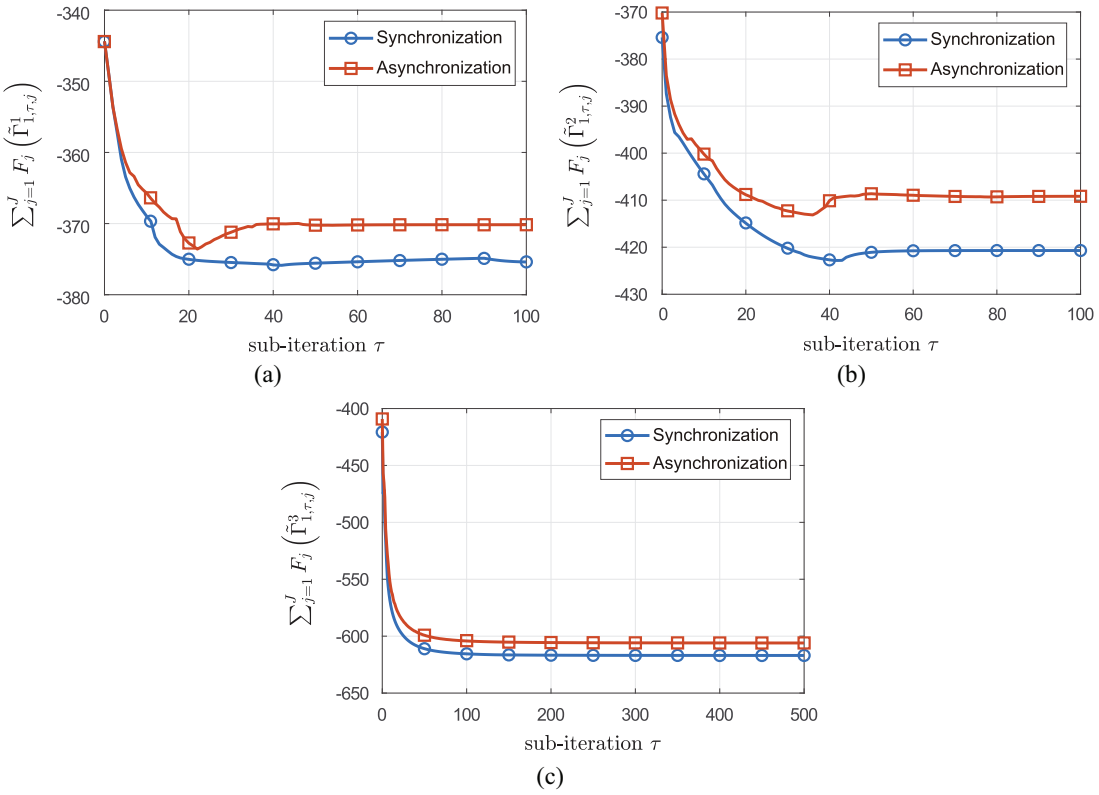

Fig. 12. Performance comparison between update-synchronization and update-asynchronization for (a) $q = 1$, (b) $q = 2$, and $q = 3$.

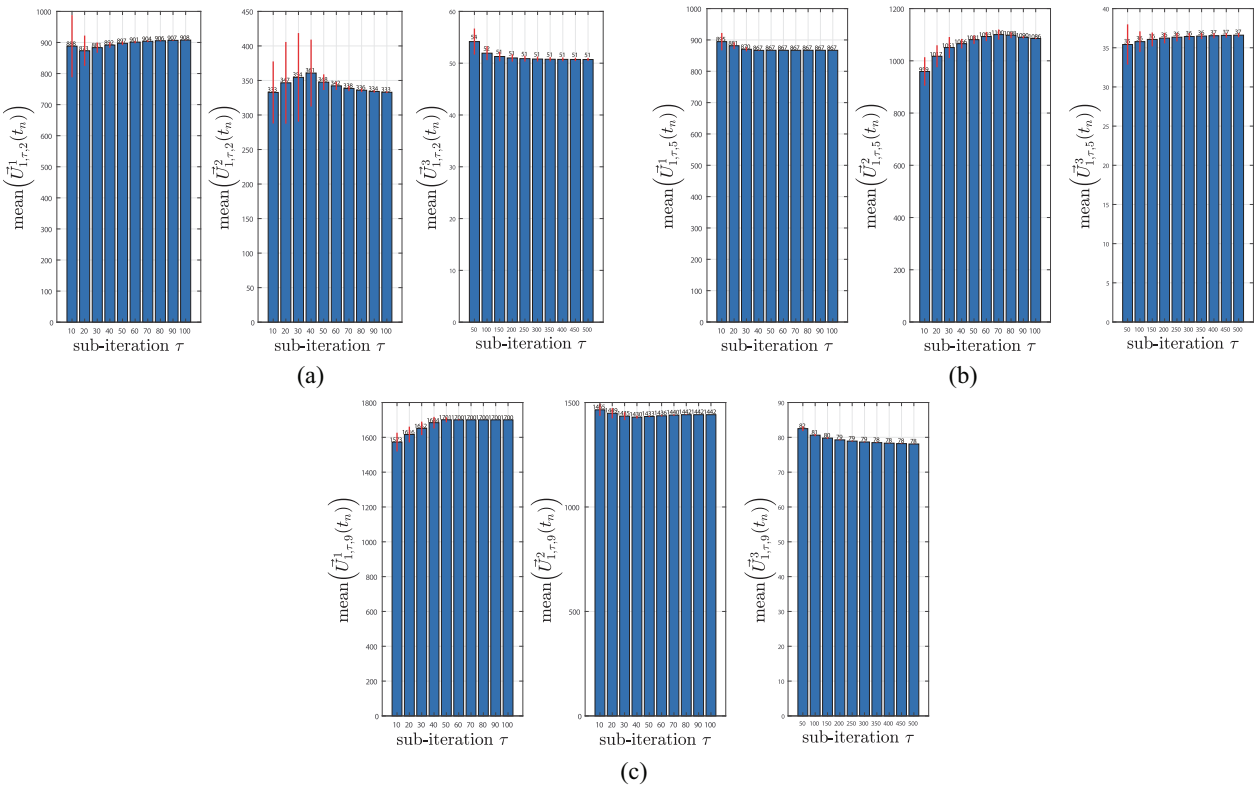

Fig. 13. Mean-bars with confidence interval of the updated coordinates in three dimensions by the GSDA subject to update asynchronization (from left to right: $q = 1, 2, 3$) for (a) UAV 2, (b) UAV 5, and (c) UAV 9.

50 dB (corresponding to $\lambda = 5$ m), and 48 dB (corresponding to $\lambda = 10$ m), are compared and the corresponding results are plotted in Fig. 14. It can be observed that all learning curves in Fig. 14 decrease and converge in the long run. In our empirical experience, our proposed GSDA is robust against location errors with the LDR less than 48 dB (in terms of

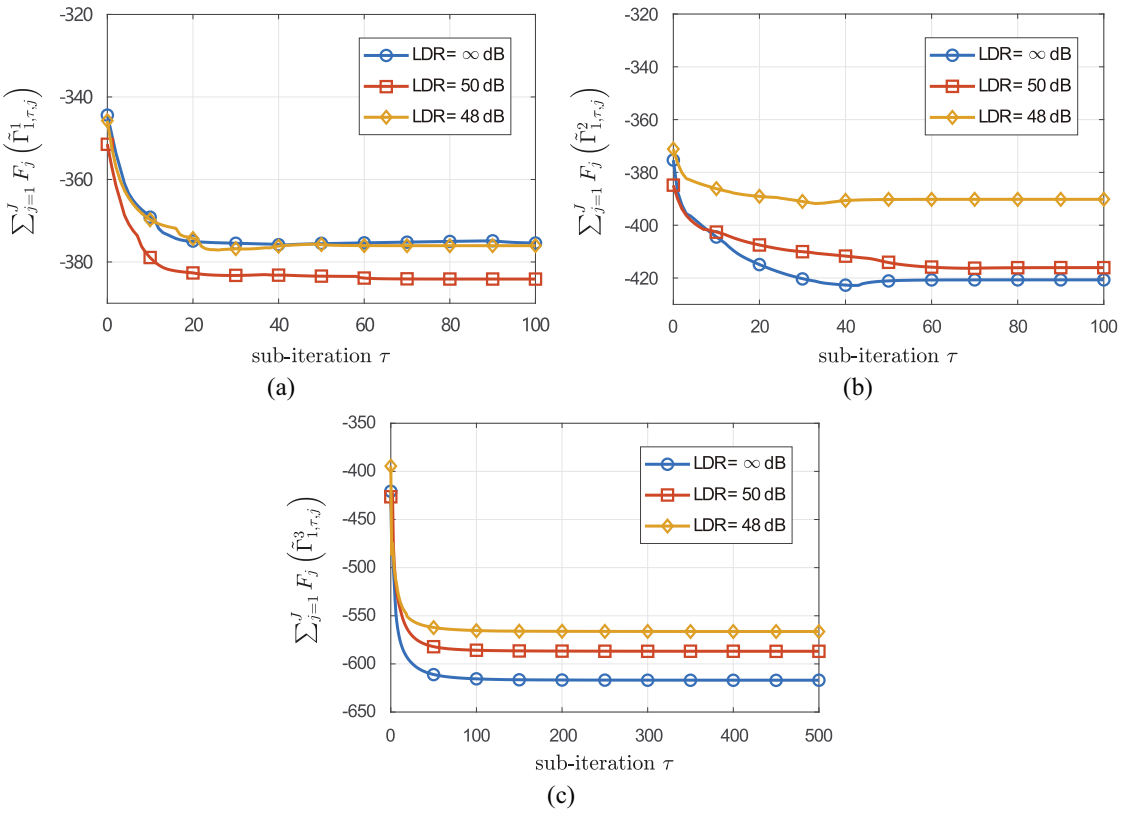

Fig. 14. Learning curves resulting from the UAVs' location matrix $\tilde{\mathcal{L}}_k^{\lambda}(t_n)$ subject to different LDRs (∞, 50, and 48 dB corresponding to $\lambda = 0$, 5, and 10 m, respectively) for (a) $q = 1$, (b) $q = 2$, and (c) $q = 3$.

10 m localization error) which are within the common GPS localization-precision range.

V. Conclusion

In this work, we study a timely wireless communication research topic, namely, UAV-aided wireless ad hoc network. A novel robust distributed UAV-deployment algorithm, namely, the GSDA, is proposed to maximize the overall channel capacity, which can be proven to converge under certain practical assumptions. The network configuration is assumed to be partitioned into square cells, each of which is served by a UAV as the base station. Besides, we have also designed a distributed genetic algorithm for comparison. The simulation results demonstrate that our proposed GSDA outperforms the distributed genetic algorithm while the latter scheme cannot converge satisfactorily. Moreover, we also propose a polynomial-regression-based predictor to facilitate a look-ahead version of GSDA which can take advantage of the predicted users' future locations. Important factors, including update asynchronization and location errors are also studied through simulations. Our proposed GSDA can serve for robust UAV-aided 3-D wireless ad hoc networks in the future.

References

[1] R. Shahzadi, M. Ali, and M. Naeem, "UAV placement and resource management in public safety networks: An overview," in *Intelligent Unmanned Air Vehicles Communications for Public Safety Networks* (Unmanned System Technologies). Singapore: Springer Nat., May 2022, pp. 19–49.

[2] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.

[3] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based Internet of Things services: Comprehensive survey and future perspectives," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 899–922, Dec. 2016.

[4] N. Wang, P. Wang, A. Alipour-Fanid, L. Jiao, and K. Zeng, "Physical-layer security of 5G wireless networks for IoT: Challenges and opportunities," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8169–8181, Oct. 2019.

[5] P. K. Deb, A. Mukherjee, and S. Misra, "XiA: Send-it-anyway Q-routing for 6G-enabled UAV-LEO communications," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 2722–2731, Oct.–Dec. 2021.

[6] W. Feng, J. Wang, Y. Chen, X. Wang, N. Ge, and J. Lu, "UAV-aided MIMO communications for 5G Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1731–1740, Apr. 2019.

[7] 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.

[8] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 434–437, Aug. 2017.

[9] W. Zhang, Q. Wang, X. Liu, Y. Liu, and Y. Chen, "Three-dimension trajectory design for multi-UAV wireless network with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 600–612, Jan. 2021.

[10] F. Fazel, J. Abouei, M. Jaseemuddin, A. Anpalagan, and K. N. Plataniotis, "Secure throughput optimization for cache-enabled multi-UAVs networks," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7783–7801, May 2022.

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

[12] J. Baek, S. I. Han, and Y. Ha, "Optimal UAV route in wireless charging sensor networks," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1327–1335, Feb. 2020.
[13] I. Bucaille, S. Héthuin, A. Munari, R. Hermenier, T. Rasheed, and S. Allsopp, "Rapidly Deployable network for tactical applications: Aerial base station with opportunistic links for unattended and temporary events ABSOLUTE example," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2013, pp. 1116–1120.
[14] A. V. Savkin and H. Huang, "Deployment of unmanned aerial vehicle base stations for optimal quality of coverage," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 321–324, Feb. 2019.
[15] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.
[16] H. Huang and A. V. Savkin, "Deployment of heterogeneous UAV base stations for optimal quality of coverage," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16429–16437, Feb. 2022.
[17] E. Kalantari, I. Bor-Yaliniz, A. Yongacoglu, and H. Yanikomeroglu, "User association and bandwidth allocation for terrestrial and aerial base stations with backhaul considerations," in *Proc. IEEE 28th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.
[18] F. Cheng et al., "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6732–6736, Jul. 2018.
[19] R. Duan, J. Wang, C. Jiang, H. Yao, Y. Ren, and Y. Qian, "Resource allocation for multi-UAV aided IoT NOMA uplink transmission systems," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7025–7037, Aug. 2019.
[20] B. Jiang, J. Yang, H. Xu, H. Song, and G. Zheng, "Multimedia data throughput maximization in Internet of Things system based on optimization of cache-enabled UAV," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3525–3532, Apr. 2019.
[21] X. Huang, K. Yan, H.-C. Wu, and Y. Wu, "Unmanned aerial vehicle hub detection using software-defined radio," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Jun. 2019, pp. 1–6.
[22] Z. Gao, D. Chen, S. Cai, and H.-C. Wu, "OptDynLim: An optimal algorithm for the one-dimensional RSU deployment problem with nonuniform profit density," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1052–1061, Feb. 2019.
[23] Z. Gao, D. Chen, S. Cai, and H.-C. Wu, "Optimal and greedy algorithms for the one-dimensional RSU deployment problem with new model," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7643–7657, Aug. 2018.
[24] P. Yang, X. Cao, C. Yin, Z. Xiao, X. Xi, and D. Wu, "Proactive drone-cell deployment: Overload relief for a cellular network under flash crowd traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2877–2892, Oct. 2017.
[25] R. Ghanavi, E. Kalantari, M. Sabbaghian, H. Yanikomeroglu, and A. Yongacoglu, "Efficient 3D aerial base station placement considering users mobility by reinforcement learning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.
[26] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 2898–2904.
[27] H. Dai, H. Zhang, M. Hua, C. Li, Y. Huang, and B. Wang, "How to deploy multiple UAVs for providing communication service in an unknown region?" *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1276–1279, Aug. 2019.
[28] S. S. Ram, V. V. Veeravalli, and A. Nedic, "Distributed non-autonomous power control through distributed convex optimization," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 3001–3005.
[29] S. S. Ram, V. V. Veeravalli, and A. Nedić, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, Jul. 2010.
[30] "Floating-point math speed vs precision." 2014. [Online]. Available: http://nicolas.limare.net/pro/notes/2014/12/16_math_speed/
[31] K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 22–37, Nov. 1996.
[32] K. Choi, D.-H. Jang, S.-I. Kang, J.-H. Lee, T.-K. Chung, and H.-S. Kim, "Hybrid algorithm combing genetic algorithm with evolution strategy for antenna design," *IEEE Trans. Magn.*, vol. 52, no. 3, pp. 1–4, Mar. 2016.
[33] I. Ciornei and E. Kyriakides, "Hybrid ant colony-genetic algorithm (GAAPI) for global continuous optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 1, pp. 234–245, Feb. 2012.
[34] R. D. Villarroel, D. F. García, M. A. Dávila, and E. F. Caicedo, "Particle swarm optimization vs genetic algorithm, application and comparison to determine the moisture diffusion coefficients of pressboard transformer insulation," *IEEE Trans. Dielectr. Electr. Insul.*, vol. 22, no. 6, pp. 3574–3581, Dec. 2015.
[35] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2012.

Xiao Yan (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in instrument science and technology from the University of Electronic Science and Technology of China, Chengdu, Sichuan, China, in 2003, 2006, and 2010, respectively.

In July 2010, he joined the faculty of the School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, where he is currently an Associate Professor. From August 2015 to August 2016, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Michigan–Dearborn, Dearborn, MI, USA. Besides, he is currently with the Aircraft Swarm Intelligent Sensing and Cooperative Control Key Laboratory of Sichuan Province, Chengdu. His research interests include the areas of cognitive radio, wireless sensing network, communication signal processing, and machine learning.

Yehan Lin received the B.S. degree in applied mathematics and the M.S. degree in electronic and communication engineering from the University of Electronic Science and Technology of China, Chengdu, Sichuan, China, in 2018 and 2021, respectively.

His current research interests include wireless communication, communication signal processing, and distributed UAV networks.

Hsiao-Chun Wu (Fellow, IEEE) received the B.S.E.E. degree from National Cheng Kung University, Tainan City, Taiwan, in 1990, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 1993 and 1999, respectively.

From March 1999 to January 2001, he had worked with Motorola Personal Communications Sector Research Labs as a Senior Electrical Engineer. In January 2001, he joined the faculty of the Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA, USA, where he is currently a Distinguished Professor. From July to August 2007, he had been a Visiting Assistant Professor with Television and Networks Transmission Group, Communications Research Centre, Ottawa, ON, Canada. From August to December 2008, he was a Visiting Associate Professor with the Department of Electrical Engineering, Stanford University, Stanford, CA, USA. He has published more than 300 peer-refereed technical journal and conference articles in electrical and computer engineering. His research interests include the areas of wireless communications and signal processing.

Dr. Wu currently serves as an Associate Editor for IEEE TRANSACTIONS ON BROADCASTING and IEEE TRANSACTIONS ON SIGNAL PROCESSING and an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE TRANSACTIONS ON MOBILE COMPUTING. Besides, he is an Academic Editor for *Sensors*. He used to serve as an Editor and the Technical Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and *IEEE Communications Magazine* and an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE COMMUNICATIONS LETTERS, IEEE SIGNAL PROCESSING LETTERS, and *IEEE Communications Magazine*. He has also served for numerous textbooks, IEEE/ACM conferences, and journals as the technical committee, the symposium chair, the track chair, or the reviewer in signal processing, communications, circuits, and computers. He is an IEEE Distinguished Lecturer.

Qian Wang received the B.S., M.S., and Ph.D. degrees in instrument science and technology from the University of Electronic Science and Technology of China, Chengdu, China, in 2005, 2008, and 2015, respectively.

In 2008, she joined the faculty of the School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu, where she is currently an Associate Professor. From September 2017 to September 2018, she was a Visiting Scholar with the Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA, USA. Her research interests include wireless sensing network, wireless communications, machine learning, and communication signal processing.

Shenglong Zhu received the B.S. degree in Internet of Things engineering from the University of Electronic Science and Technology of China, Chengdu, Sichuan, China, in 2020, where he is currently pursuing the master's degree in control science and engineering with the School of Aeronautics and Astronautics.

His current research interests include wireless ad hoc networks, communication signal processing, and distributed optimization.