Deployment Algorithms for UAV Airborne Networks towards On-demand Coverage

Haitao Zhao, Senior Member, IEEE, Haijun Wang, Weiyu Wu, and Jibo Wei, Member, IEEE

Abstract—Due to the flying nature of Unmanned Aerial Vehicles (UAVs), it is very attractive to deploy UAVs as aerial base stations and construct airborne networks to provide service for on-ground users at temporary events (such as disaster relief, military operation and so on). In the constructing of UAV airborne networks, a challenging problem is how to deploy multiple UAVs for on-demand coverage while at the same time maintaining the connectivity among UAVs. To solve this problem, we propose two algorithms: a centralized deployment algorithm and a distributed motion control algorithm. The first algorithm requires the positions of user equipments (UEs) on the ground and provides the optimal deployment result (i.e., the minimal number of UAVs and their respective positions) after a global computation. This algorithm is applicable to the scenario that requires minimum number of UAVs to provide desirable service for already known on-ground UEs. Differently, the second algorithm requires no global information or computation, instead, it enables each UAV to autonomously control its motion, find the UEs and converge to on-demand coverage. This distributed algorithm is applicable to the scenario where using given number of UAVs to cover UEs without UEs’ specific position information. In both algorithms, the connectivity of the UAV network is maintained. Extensive simulations validate our proposed algorithms.

Index Terms—Unmanned Aerial Vehicle (UAV), airborne network, deployment algorithms, on-demand coverage, connectivity.

I. INTRODUCTION

Due to the flying nature of Unmanned Aerial Vehicles (UAVs), they can have line-of-sight (LoS) connections towards ground users leading to an improved coverage and rate performance. Therefore, it is very attractive that using UAVs as aerial base stations (BSs) to boost the wireless capacity and enhance the coverage for hotspot areas (such as conference center, open-air concert and football stadium) or at temporary events (such as disaster relief and military operation) [1], [2].

Compared to terrestrial base stations, mobile UAVs can flexibly change their locations, therefore, how to optimally deploy their locations is an interesting problem. On one hand, since the user equipments (UEs) may distribute unevenly on the ground, UAVs should be deployed to provide on-demand coverage for UEs. On the other hand, to provide end-to-end communication between far-away UEs and enable a robust UAV airborne network, UAVs should keep connected (i.e., each UAV has at least one route to any other UAVs in the network). Therefore, it is crucial to provide on-demand coverage and at the same time guarantee connectivity in the deployment of UAV airborne networks.

There are inspiring researches on terrestrial BS (or small BS) deployment in cellular network. In [3], the authors investigated three different cell sorting criteria for the cell switch-off approach in energy saving and further propose a centralized set cover based algorithm. In [4], the authors aimed to plan small cells by maximizing the number of traffic demand nodes with a limited budget. In [5], the authors achieved the energy-optimal density of BSs corresponding to a given user density based on stochastic geometry. In [6], we proposed a solution to the small cell deployment problem in cellular networks to save energy. However, the solutions in these works cannot be applied directly to UAV scenarios mainly based on the three facts: (i) in terrestrial BS deployment, the BSs are fixed, while UAVs are moving; (ii) the terrestrial BS deployment can be done by using statistical traffic model, while UAV deployment mostly faces temporary events (any two temporary events may be quite different) and therefore the statistical model cannot stand; (iii) in the terrestrial BS deployment, the connection among BSs is not a concern since they are usually connected via fibers.

Recent researches that focus on deploying UAV networks can be generally classified into three categories. The first category mainly considered the network topology of UAV networks, for instance to maximize network coverage [7], [8], to maintain the connectivity [9]–[11] or to form a specific topology [12]–[15]. In this category, the UAV network itself is the research focus while the distribution of on-ground UEs is not considered. The second category mainly considers the application of UAVs for target covering or tracking [16]–[18]. In this category, although the on-ground targets are considered, the targets and the UAVs are non-cooperative because UAVs use sensors to detect or track the target(s), and therefore there is no communication between target(s) and UAVs. The third category considers using the UAVs as BSs to serve the mobile UEs, where UAVs are usually all connected to and controlled by control stations [19]. And therefore, the connection among UAVs is not a concern. In this research, we are considering a new application scenario where massive on-ground UEs access the UAVs in the sky, requiring the UAVs to form a bi-connected airborne network and provide on-demand coverage to the UEs.

There are two opposing problems in this application scenario. First, given the position of each UE, how to use the minimal number of UAVs to satisfy UE’s service requirements?
Second, without the specific position of each UE, how to make given number of UAVs autonomously move and cover UEs as many as possible? In both problems, the connectivity of UAV network should be maintained for robustness. To solve the first problem, we present an off-line centralized algorithm. Given the positions of on-ground UEs, this algorithm can obtain the minimal number of UAVs and their respective positions, while satisfying the coverage requirements of UEs on one hand and maintaining the connectivity among UAVs on the other hand. This algorithm suits the static network scenario and outputs the deployment results via global optimization. And to solve the second problem, we propose a distributed algorithm that does not require the position of each UAV beforehand. In this algorithm, each UAV autonomously controls its motion based on local information, i.e., the on-ground UEs that it finds and the status of neighboring UAVs, and iteratively flies to the proper position. This algorithm suits the dynamic network scenario that requires autonomous control of UAVs, outputting not only the final UAV positions but also the motion track of each UAV. The main contributions of this work are as follows.

- We propose solutions to the on-demand deployment problem with both a centralized algorithm and a distributed algorithm to suit different application scenarios.
- We consider the uneven distribution of on-ground UEs and their requirements on quality of service (QoS).
- We enable the obstacle avoidance in the process of UAV deployment.
- The distributed algorithm suits both the static scenario, where the UEs are static, and the dynamic scenario, where the UEs are moving.
- We conduct extensive experiments to validate the proposed algorithms. Results show that both algorithms achieve the bi-connected airborne network and provide on-demand coverage.

The rest of this paper is organized as follows. We review the related work in Section II and specify the system model as well as the problem in Section III. We then propose the centralized deployment algorithm and the distributed motion control algorithm respectively in Section IV and Section V. Simulation validation is presented in Section VI. And Section VII concludes this research.

II. RELATED WORK

The deployment problem of UAVs has recently attracted extensive researches in literatures, where various solutions are proposed towards different objectives considering different requirements. These objectives are described as follows.

A. Maximizing the coverage area

Given the number of UAVs, how to maximize the coverage area is a practical and interesting problem. Al-Hourani [20] proposed a model of the LoS communication probability between a UAV and a UE receiver and evaluated the optimal altitude for a single UAV that maximizes the coverage region. [21] further extended to the scenario with two UAVs and investigated the impact of altitude on transmission power while considering interference between them. In [22] and [23], both authors considered the efficient placement of one UAV in 3D space to maximize the coverage of a UAV. As to the deployment of multiple UAVs, the goal of [24] is to evenly distribute the UAVs in the area. In [7], an efficient deployment approach was proposed based on the circle packing theory that leads to a maximum coverage while each UAV uses the minimum transmit power. In [8], the authors developed a heuristic algorithm based on particle swarm optimization, and the algorithm suboptimally finds the minimum number of UAVs and their locations to serve all the users in a particular region with different user densities. Chen [25] proposed a novel framework that leverages user-centric information, such as mobility patterns, to effectively deploy UAVs while maximizing the users quality of experience using the minimum total transmit power of the UAVs. Yet, these works do not take the connectivity among the UAVs into consideration while deploying multiple UAVs. On the contrary, both [9] and [26] proposed a simple motion control algorithm for robotic sensor networks (the robotic sensors can be UAVs) that can maximize the coverage area while maintaining the connectivity of the sensors in the network, and in [10], the authors used multiple UAVs as wireless relays to serve ground sensors and addressed the tradeoff between UAV connectivity and maximizing coverage. However, in [9], [10] and [26], the network does not act as a backbone network and there is no terminal access from the ground.

B. Maintaining the topology or connectivity

In order to construct a robust network, it is important to maintain the connectivity of the network. To this end, authors in [26] developed a simple formation maneuvering control of multi-agent systems with connectivity maintenance properties, based on a distance-based potential function. And [27] proposed gradient-based flocking control protocols that could guarantee the desired distance stabilization and the connectivity of the underlying communication network simultaneously. In [28], the authors presented a distributed controller for multiple UAVs to make a target-centric formation while maintaining the network connectivity. In [29], the proposed algorithm can find the topology holes and eliminate them, resulting in a more robust network. Besides keeping the network connected only, the mobile robotic network keeping a regular topology is useful in many scenarios. Therefore, many researches have been conducted to form a specific topology, for instance, a line topology for relay [12], a ring topology for target surveillance [13], a triangular lattice topology for mapping, navigation or maximizing coverage [9], [14], a square lattice topology for collecting data, delivering payloads or constructing in parallel [30]. Similar to the researches towards maximizing the coverage area, the basic idea in these works towards maintaining the topology or connectivity is to take the UAVs as flying ad hoc network and does not consider the distribution of terminals (traffics) on the ground.

C. Target tracking

This category considers multiple UAVs tracking targets on the ground. Moon [16] presented a framework of information-theoretic based task assignment for multiple UAVs to track
moving targets. In [17], a distributed controller for a multi-agent system is proposed to track one target. In [18], the authors proposed a distributed framework for joint optimization of target detecting and target tracking. In [32], a hierarchical model based on multiple Gaussians uncertainty theory was employed for multi-object tracking. When multiple UAVs fly in the air cooperatively to track mobile targets, they usually move in a flock manner. Therefore, the flocking algorithms are recently reported to address the issues of coordination among multiple UAVs or robotic sensors [30]–[33]. The result of flocking algorithms in target tracking is a mass of sensors around one target. In [34], the researchers introduced a semi-flocking algorithm, which is a modified version of the flocking algorithm. It allows some of the sensors to flock around the targets and the others to freely move in open area for target searching. And authors in [35] developed static and dynamic safety index maps respectively to explore a suitable path for UAVs.

In the target tracking scenarios, the targets and the UAVs (or mobile robotic sensors) are un-cooperative that there are no communications between targets and UAVs. Furthermore, the number of UAVs is usually much bigger than that of targets.

In this research, we are considering massive on-ground UEs accessing the UAVs in the sky. The UAVs will form a bi-connected airborne network and provide on-demand coverage to the terminals. To the best of our knowledge, the deployment problem towards on-demand coverage and connectivity for UAV airborne networks has not been well studied previously. Table I summarizes the differences of the centralized algorithm (CA) and the distributed algorithm (DA) that proposed in this research, comparing to previously reported studies.

III. SYSTEM MODEL AND PROBLEM DEFINITION

A. System model

There are three kinds of communication objects in the considered network: UAV, UE and controller, which are illustrated in Fig. 1. The UAVs can move flexibly in the sky to any position, except where there are obstacles (such as skyscrapers, mountains or high trees). In [8], [20]–[23], the authors have studied the optimal altitude for maximum coverage from the sky. In this research, we assume that UAVs are flying at the fixed optimal height. The UE can be any on-ground terminal device, such as a sensor, a mobile phone and so on, which uses the aerial UAV-based backbone network to transmit information. The UEs are unevenly distributed on the ground because of the operation requirements or the terrain limitations. And also we consider that all the UAVs and UEs are equipped with omni-directional antennas because they have a natural advantage to transmit and receive signals in highly mobile environments. One or multiple controllers are located inside or at the edge of the area of interest (AoI), and they can be either ground-based or sky-based (for instance, satellites, high altitude platforms or bigger UAVs). The controller can communicate with at least one UAV to transmit control packets or connect the airborne network with other network structures, for instance, the satellite networks or the Internet.

B. UAV-to-UEV channel model

The UAV-to-UEV channels are mainly dominated by the LoS link [1]. Therefore, the pathloss between UAV i and UAV j can be considered as the free space propagation loss (FSPL):

$$L_{\text{LoS}}^{i,j} = 20 \log d_{i,j} + 20 \log f_0 + 20 \log(4\pi/c), \quad (1)$$
where $d_{i,j}$ is the distance between UAV $i$ and UAV $j$, $f_0$ is the carrier frequency of UAV-to-UAV channel and $c$ is the speed of light. Given the transmission power of UAVs in the airborne network, the maximum communication range is determined, which is denoted by $R_c$ in this research.

C. UAV-to-UE channel model

Generally, the air to ground propagation channel is modelled by jointly considering the LoS and NLoS components along with their occurrence probabilities separately [21]. Note that for NLoS links due to the shadowing effect and reflection of signals from obstacles, the pathloss is higher than that of the LoS links. The probability of UAV $i$ having LoS connections to UE $k$ at an elevation angle (see the $\theta_{i,k}$ in Fig. 1) is given by [22], [36]

$$P(\text{LoS}, \theta_{i,k}) = \frac{1}{1 + \alpha \exp(-\beta(\theta_{i,k} - \alpha))}, \quad (2)$$

where $\alpha$ and $\beta$ are constant values which depend on the environment (rural, urban, dense urban and etc.). Therefore, the pathloss between UAV $i$ and UE $k$ can be expressed as

$$g_{i,k} = P(\text{LoS}, \theta_{i,k}) \times L_{i,k}^{\text{LoS}} + (1 - P(\text{LoS}, \theta_{i,k})) \times L_{i,k}^{\text{NLoS}}, \quad (3)$$

where $L_{i,k}^{\text{LoS}}$ and $L_{i,k}^{\text{NLoS}}$ are the average pathloss for LoS and NLoS links, and they can be expressed as [37], [38]

$$L_{i,k}^{\text{LoS}} = 20 \log d_{i,k} + 20 \log f + 20 \log(4\pi/c) + \eta_{\text{LoS}},$$

$$L_{i,k}^{\text{NLoS}} = 20 \log d_{i,k} + 20 \log f + 20 \log(4\pi/c) + \eta_{\text{NLoS}},$$

$$FSPL$$

where $\eta_{\text{LoS}}$ and $\eta_{\text{NLoS}}$ are the average additional pathloss to the FSPL under LoS and NLoS respectively, which depend on the environment. $f$ is the carrier frequency of UAV-to-UE channel and $d_{i,k}$ is the distance between UAV $i$ and UE $k$.

We assume that if the received signal to interference plus noise ratio (SINR) of UE $k$ from UAV $i$ exceeds a threshold, denoted by $\Lambda_{th}$, UE $k$ is covered and its transmission rate and QoS can be supported by UAV $i$, and the SINR is calculated by [6]

$$\gamma_{i,k} = \frac{p_{i,k}g_{i,k}}{\sum_{j \neq i} p_{j,k}g_{j,k} + N_{gw}} \geq \Lambda_{th}, \quad (5)$$

where $p_{i,k}$ is the transmission power of UAV $i$ to UE $k$, $g_{i,k}$ is the channel gain between them, and $N_{gw}$ indicates the power of the Gaussian white noise.

When the UEs are mobile, there may exist Doppler effect. The Doppler effect will finally affect the SINR when the UEs receive signals from the UAVs. There are many researches in PHY layer trying to handle Doppler effect, for instance via adopting the frequency shift estimation or diversity technology [39], [40]. In this paper, we do not explicitly consider the Doppler effect.

D. Problem definition

In the deployment of UAVs, there are two opposing problems as follows.

Problem 1: Given the number and positions of UEs, how to use the minimum UAVs to provide on-demand coverage for the UEs?

Problem 2: Given the number of UAVs, how to optimally deploy them to obtain the best utilization, i.e., cover as many UEs as possible?

And in both problems, the final deployment of UAVs should satisfy the following requirements:

(i) The UAV airborne network is bi-connected for robustness, i.e., each UAV can communicate with at least other two UAVs.

(ii) The coverage outage proportion, i.e., the proportion of uncovered UEs, should be restricted below a predefined threshold $\tau$. And the QoS requirements of the covered UEs should be satisfied.

(iii) Considering the capacity of each UAV, the number of UEs served by one UAV should be less than a threshold $M_{\text{max}}$.

(iv) Any two UAVs cannot be closer than a threshold $R_{\text{opt}}$, in order to avoid possible collision.

(v) The deployment of UAVs should avoid obstacles.

For Problem 1, we model it as a centralized discrete deployment problem which reduces the search space since that we only care about the minimal number of UAVs and their final respective locations. For Problem 2, we model it as a continuous programming problem since we care about not only the UAV positions but also the continuous motion track of each UAV. As solutions to these two problems, we respectively propose a centralized deployment algorithm in Section IV and a distributed motion control algorithm in Section V.

IV. CENTRALIZED DEPLOYMENT ALGORITHM

A. Basic idea

The basic idea of this centralized deployment algorithm is that we first assume that enough candidate UAVs are initially deployed with obstacle avoidance which can be easily achieved by using the geographical information. Here ‘enough’ means that every UE is covered by at least one UAV and the UAVs are bi-connected. (For instance, we initially deploy candidate UAVs at all the cross points in a high-density grid that covers the AoI). And then we iteratively delete the redundant UAVs (a UAV is called redundant if deleting this UAV will neither affect the network connectivity nor violate the coverage outage restriction). And when no redundant UAVs can be further deleted, we get the final deployment result.
B. Problem formulation

Assuming there are $M$ on-ground UEs, and initially we deploy $N$ candidate UAVs. We introduce matrices $a = [a_i]_{1 \times N}$, $b = [b_{i,k}]_{N \times M}$ and $c = [c_{i,j}]_{N \times N}$ to denote the states of candidate UAVs, the associations between UAVs and UEs, and the links among UAVs respectively, where $a_i, b_{i,k}, c_{i,j} \in \{0,1\}$. Specifically, $a_i = 1$ indicates that the candidate UAV $i$ is active, and $a_i = 0$ indicates that it is deleted; $b_{i,k} = 1$ means that UE $k$ is associated with UAV $i$, otherwise, $b_{i,k} = 0$; $c_{i,j} = 1$ means that UAV $i$ is within UAV $j$’s communication range, otherwise, $c_{i,j} = 0$. Note that $c$ is a symmetric matrix, and if $a_i = 0$, $\sum_{j=1}^{N} c_{i,j} = \sum_{j=1}^{N} c_{j,i} = 0$. With these denotations, we can calculate the degree of UAV $i$ and UE $k$ by $\sum_{k=1}^{M} b_{i,k}$ and $\sum_{i}^{N} b_{i,k}$ respectively, which indicate the number of UEs it serves for UAV $i$ and the number of UAVs which serves it for UE $k$, respectively. And also we can calculate the neighbor quantity of UAV $i$ by $\sum_{j=1,i \neq j}^{N} c_{i,j}$.

The objective of the deployment is to minimize the number of finally deployed UAVs (i.e., the final number of active UAVs). Therefore, we can formulate the UAV deployment problem as follows:

$$\min \sum_{i=1}^{N} a_i,$$

s.t. 
$$C1 : a_i, b_{i,k}, c_{i,j} \in \{0,1\}, \quad \forall i, k, j,$$

$$C2 : \sum_{k=1}^{M} b_{i,k} \leq 1, \quad \forall i,$$

$$C3 : \sum_{k=1}^{M} b_{i,k} \leq M_{i}, \quad \forall i,$$

$$C4 : b_{i,k} \leq a_i, \quad \forall i, k,$$

$$C5 : a_i b_{i,k} c_{i,k} \geq b_{i,k} a_i \sum_{j=1}^{N} a_j p_{j,i} k g_{j,k} + N g_{w}, \quad \forall i, k,$$

$$C6 : \sum_{j=1}^{N} b_{i,k} \geq (1 - \tau) M, \quad \forall i,$$

$$C7 : \sum_{j=1}^{N} c_{i,j} \geq 2, \quad \forall a_i = 1,$$

$$C8 : 3n < N, \quad (c^a)_{i,j} \neq 0, \quad \forall i \neq j,$$

where $C1$ is the Boolean constraint for UAV deployment. $C2$ indicates that a UE can only be served by at most one UAV. $C3$ imposes an upper bound to the amount of UEs one UAV could cover. $C4$ means that associations to deleted UAVs are not permitted for any UE. $C5$ is a transformation of (5) to ensure the QoS requirement from a covered UE. $C6$ stipulates that the percentage of unserted UEs should be less than a predefined threshold $\tau$. For a deployed UAV, $C7$ guarantees the bi-connection and $C8$ ensures that there is at least one route between any two finally deployed UAVs. Note that $C7$ cannot ensure $C8$, which will be explained later.

C. Algorithm design

The optimization problem in (6) is NP-hard, which is difficult to solve optimally for a large-scale deployment [8], [41]. Therefore, in this paper we propose a heuristic algorithm which reduces the search space by specifying the candidate deployment locations. The heuristic algorithm includes five steps, which are illustrated in Fig. 2. And the denotations used in the algorithm are listed in Table II.

**Step 1:** Initialization (deploy enough candidate UAVs and calculate $c$)

Set enough candidate UAVs all over the AoI while avoiding obstacles. Initialize the elements of $a$ to be 1 and those of $\zeta$ and $\xi$ to be 0, which means that all the candidate UAVs are active by default. Then, we calculate the distances among all the UAVs and get $c$. The detailed process is shown in Algorithm 1.

**Algorithm 1:** Initialization

1. Set enough candidate UAVs, i.e., predefine $U$, avoiding obstacles.
2. $\zeta_i := 0, a_i := 1, \forall i \in U; \xi_k := 0, \forall k \in M; U_r = U, U_f = \emptyset$
3. for $\forall i \in U$ and $\forall j \in U$
4. Calculate the distance $d_{i,k}$ between UAV $i$ and UAV $j$.
5. if ($d_{i,j} \geq R_c$) then $c_{i,j} := 1$.
6. else $c_{i,j} := 0$.
7. end if
8. end for

**Step 2:** Build/update connection graph
To build (or update) the connection graph, we calculate the SINR between each UAV and UE, and then compare it with the threshold $\Lambda_{th}$. In this way, we obtain all the possible connections between UAVs and UEs. At the same time, the degrees of all the UAVs and UEs, i.e., $\zeta$ and $\xi$, can also be obtained. The detailed process is shown in Algorithm 2.

**Algorithm 2**: Build/update connection graph

```plaintext
1: for $\forall i \in U$ and $\forall k \in M$ do
2:   Calculate the SINR $\gamma_{i,k}$ between UAV $i$ and UE $k$.
3:   if $\gamma_{i,k} \geq \Lambda_{th}$ then
4:     $\beta_{i,k} := 1$, $\zeta_i := \zeta_i + 1$, $\xi_k := \xi_k + 1$.
5:   else
6:     $\beta_{i,k} := 0$.
7:   end if
8: end for
```

**Step 3**: Delete the redundant connections between UAVs and UEs

If there exists at least one UE whose degree is larger than 1, it means that there are redundant connections. The purpose of this step is to delete these redundant connections. To begin with, we find the index of the UAV, say $j$, which has the maximum degree, and set it to be active. If its degree is larger than the maximum threshold ($M_{\text{max}}$), we delete its connection to the UE with the largest degree iteratively until the degree of the UE meets the constraint C3. Our rationale is that the UEs with larger degrees have more choices to connect to other UAVs even if we delete their connections to UAV $j$. The detailed process is shown in Algorithm 3.

**Algorithm 3**: Delete redundant connections between UAVs and UEs.

```plaintext
1: while (The maximum value in $\xi$ is larger than 1) do
2:   Find the UAV index with the maximum degree, say $j$.
3:   Find all UEs that are connected to this UAV $j$, and record all these UEs in a set $\phi$.
4:   Save all degrees of UEs in $\phi$ to a new set $\psi$.
5:   while (The length of $\phi$ is larger than $M_{\text{max}}$) do
6:     Find the UE with the largest degree in $\phi$, say UE $k$.
7:     Delete the connection between UAV $j$ and the UE $k$.
8:     Update $\phi$ and $\psi$.
9:   end while
10: Delete the connections of the remaining UEs to other UAVs.
11: Update $b$, $\zeta$ and $\xi$.
12: end while
13: Calculate $a$ based on $b$.
```

**Step 4**: Delete the idle UAVs after making sure all constraints are satisfied

We must take three problems into consideration before making the final decision on whether to delete an idle UAV. See Fig. 3, assuming that there are four idle UAVs, i.e., 1, 2, 3 and 4, that have no UEs to cover (their UAV degrees are zero) after the redundant connections are deleted. The three problems in deleting idle UAVs are as follows. (i) If UAV 1 is deleted, UAV 5 will have only one neighbor (UAV 6), which would break the bi-connection constraint, i.e., C7 in (6); (ii) If UAV 2 is deleted, although C7 for all remained UAVs are not broken, C8 is no longer satisfied since the UAVs are split into two isolated group (i.e., the information-isolated island problem); (iii) For the two neighboring idle UAVs 3 and 4, we may fail in deleting UAV 3 due to constraint C7 for UAV 4 when UAV 4 has not been deleted yet. However, after UAV 4 is deleted, we can delete UAV 3 successfully. In other words, the order of deleting UAVs matters.

![Fig. 3. Potential problems resulting from deleting a candidate UAV location.](image)

The first problem could be handled by checking the neighbor quantities of all the neighbors of an idle UAV before it is deleted. For the second problem, we use Algorithm 4 where the breadth-first search (BFS) [42] is adopted to detect whether this problem will occur for a given idle UAV. $\Theta$ saves the neighbor set of the given idle UAV and the algorithm outputs a Boolean value $e = 1$ if constraint C8 is satisfied. The third problem can be solved by introducing a retrospecting mechanism. The basic idea is that, once a redundant UAV location is deleted successfully, we need to recheck whether its neighbors with history of failing to be deleted could be deleted again as described in Algorithm 5.

**Algorithm 4**: Check if information-isolated island problem will happen after deleting UAV $i$

**Input**: $N$, $i$, $c$, $a$.

**Output**: $e$.

1: Initialization: $e=1$.
2: Get remaining neighbor set of UAV $i$ via $c$ and $a$, saved in $\Theta$.
3: for $\forall r \in \Theta$ do
4:     $\Theta_{i} := \Theta \setminus \{r\}$
5:     Using BFS to find paths which avoid $i$ from $r$ to each UAV in $\Theta_{i}$, based on $c$.
6:     if (A path from $r$ to any UAV in $\Theta_{i}$ is found) then
7:         Continue.
8:     else
9:         $e=0$ and break.
10: end if
11: end for

If there is an idle UAV emerging after Algorithm 3, Algorithm 5 is responsible for checking the constraints C6, C7 and C8 to decide whether to delete it. If these three constraints are all satisfied, we can regard this idle UAV as redundant and
Algorithm 5: Delete the idle UAVs after checking the constraints and retrospect

1: if (An idle UAV emerges, say m) then
2: Execute Algorithm 4 to check constraint C8;
3: if (C6, C7 and C8 hold simultaneously for m) then
4: Set the transmit power of the idle UAV m to 0.
5: \( a_m := 0, c_{i,m} = c_{m,i} := 0, \forall i, U_r := U_r \setminus \{m\}. \)
6: Retrospect to update \( U_r \) and \( U_f \).
7: else
8: \( U_f := [U_f, m], U_r := U_r \setminus \{m\}, a_m := 1, \) and recover the transmit power of UAV m.
9: end if
10: end if

delete it, and then the retrospecting mechanism is executed to update \( U_r \) and \( U_f \).

Step 5: Check if one more active UAV can be deleted

If there is no idle UAV emerging after Step 3 or the idle UAV cannot be deleted by Step 4, it means that for the moment any remaining UAV is associated with at least one UE or act as the connection to eliminate the information-isolated island problem. However, there is possibility that the UEs associated to one UAV (say UAV i) can all be re-associated to other neighboring UAVs without violating the constraints defined in (6), especially C3. Under this case we can delete UAV i to decrease the amount of required UAVs. Therefore, after Step 5, we would try to check if the UAV with the minimum degree in \( U_r \) can be further deleted by setting its transmit power to be 0, and then go back to Step 2 for a new iteration.

The whole algorithm ends when \( U_r = \emptyset \), which means that all the candidate UAVs have been checked at least once and no more active UAVs could be further deleted.

Although it is hard to calculate the optimality gap of the heuristic algorithm, given the total number of UEs, their respective positions and the maximum number of UEs that one UAV could serve, we can obtain the theoretical minimum number of finally deployed UAVs. This minimum UAV number is the lower bound of the optimization objective, which means that the optimality gap is smaller than the gap between the results of the heuristic algorithm and the lower bound.

D. Algorithm convergence and complexity

Considering the retrospect mechanism after deleting a redundant UAV, we need to recheck whether its neighbors with history of failing to be deleted could be deleted again. For example, after some iterations, the number of UAs in \( U_r \) is assumed to be \( N' \), and if we delete a redundant UAV in \( U_r \) successfully, the maximum number of UAs we need to recheck is \( N' - 1 \). The end conditions of the algorithm is \( U_r = \emptyset \), so we can conclude that the algorithm would converge within \( N(1 + N')/2 \) iterations theoretically under all distributions of UEs. The complexity of the initialization is \( O(N^2) \). In each iteration, the complexity of the next four steps (i.e., Step 2-5) are \( O(MN) \), \( O(MN + M^2) \), \( O(N^2) \) and \( O(1) \) respectively. Thus, the computational complexity of the whole deployment algorithm is \( O(N^4 + MN^3 + M^2N^2) \).

This off-line centralized algorithm can provide the optimal number and positions of UAVs under a given scenario where all the positions of UEs are known.

V. DISTRIBUTED MOTION CONTROL ALGORITHM

In this section, we present an on-line distributed motion control algorithm to optimally deploy the given number of UAVs to cover as many UEs as possible while maintaining the connectivity among UAVs. The distributed algorithm requires no position of each UE in advance, instead, the UAVs will discover the UEs by themselves using target recognition sensors when they fly over the UEs, and autonomously control their movements towards the optimal deployment. The basic idea is that each UAV models the demands for optimal coverage as a virtual force field, only based on its local information, i.e., the information through sensing, communicating with the UEs under its coverage and one-hop information exchange with neighboring UAVs. And each UAV will follow the force field to move towards its proper position in a distributed manner.

A. Virtual forces modelling for on-demand coverage

The force field is composed of four kinds of virtual forces:

(i) The attractive forces (\( F_{a1} \)) to drag all the UAVs together towards the center (or multiple hotspots if there exist) in the AoI and connect the UAVs covering around different hotspots, which are illustrated in Fig. 4a; (ii) The attractive forces towards the UEs (\( F_{a2} \)) for on-demand coverage, which are illustrated in Fig. 4b; (iii) The repulsive forces (\( F_{r} \)) to push away two UAVs between which the distance is closer than a desired value \( R_{opt} \), which are illustrated in Fig. 4c; and (iv) The repulsive forces (\( F_{obs} \)) to avoid obstacles, which are illustrated in Fig. 4d.

1) The attractive forces for connectivity (\( F_{a1} \))

Taking the scenario in Fig. 4 as an example, we denote \( F_{a1} \) and \( F_{a2} \) as the attractive force towards respectively Hotspot 1 and Hotspot 2 in the AoI, which will drag the UAVs to cover around the hotspots. And \( F_{a2} \) is the attractive force towards the virtual connection line between Hotspot 1 and Hotspot 2, which will drag the UAVs to form a multi-hop connection between the flocks of UAVs respectively covering around Hotspot 1 and Hotspot 2. Inspired by the law of universal gravitation, we model the attractive forces in the force field as follows,

\[
\vec{F}_a = K_a \times \frac{1}{(d_i)^2}, \quad i = 1, 2, 3,
\]

where \( d_i \) is the distance of the UAV to the hotspots or to the line between two adjacent hotspots; \( K_a \) is the attractive force factor decided by the importance of the hotspot (for the attractive force towards the line between hotspots, this factor is a constant). Since \( F_{a1} \) is to attract all the UAVs in the AoI towards or connecting the hotspots, the selection of \( K_a \) essentially depends on the size of AoI. And the direction of this attractive force is always towards the hotspot or the line between hotspots. This model is reasonable in that the attractive forces will decrease with distance, and therefore,
the UAVs will be dragged towards the nearest hotspots or the
proper positions along the connections between hotspots, and
hence will save the travel distance.

Therefore, the overall attractive force can be expressed by
\[
\vec{F}_a = F_a^1 + F_a^2 + F_a^3.
\]
(8)
And UAVs at any point in the AoI will be affected by this
resultant attractive force.

2) The attractive forces towards UEs ($\vec{F}_a$)

The attractive force $\vec{F}_a$ is towards the hotspots but not
the individual UEs. Considering the possible uneven distribution
of UEs around the hotspots, we further introduce the attractive
force towards UEs (denoted by $\vec{F}_a$) to make UAVs deployed
more precisely to cover the UEs.

The attractive force towards a UE also uses the universal
gravitation model and it can be expressed as
\[
\vec{F}_a = k_a \times \frac{1}{d_{ue}} \cdot d_{ue} < R_s,
\]
(9)
where $k_a$ is the UE force factor, $d_{ue}$ is the distance
between the UAV and the UE, and $R_s$ is the sense range of UAVs
determined by the sensors equipped on them. Since an individ-
ual UE should have less impact on UAVs than the hotspot,
therefore, the force factor ($k_a$) is less than that used in $\vec{F}_a$.

3) The repulsive forces to keep UAVs at desirable distance
($\vec{F}_r$)

The attractive forces will drag UAVs that initially are
scattered over the AoI together towards the hotspots and
the connections between them. To keep UAVs from being too
crowded and possible colliding with each other, we introduce
the repulsive forces (denoted by $\vec{F}_r$) [41]. The repulsive forces
will take effect when the distance between two UAVs is less
than a predefined optimal range, denoted by $R_{opt}$ ($R_{opt} \leq R_c$),
as illustrated in Fig. 4c. Inspired by Hooke’s Law, we model
the repulsive force between UAV $i$ and UAV $j$ as follows,
\[
\vec{F}_r = \begin{cases} 
K_r \times (R_{opt} - d_{ij}) , & d_{ij} < R_{opt} \\
0 , & \text{otherwise}
\end{cases},
\]
(10)
where $K_r$ is the repulsive force factor, $d_{ij}$ is the distance
between UAV $i$ and UAV $j$. The direction of this repulsive
force is always to push two neighboring UAVs away to the
opposite directions.

This model is reasonable in that it will lead the UAVs to
move away from their close neighbors while ensuring com-
munication. The repulsive force $\vec{F}_r$ takes effect only when the
distance between two UAVs is less than $R_{opt}$, however, once
$\vec{F}_r$ takes effect it means that the two neighboring UAVs should
be pushed away immediately to avoid collision. Therefore, $K_r$
should be much bigger than the attractive force factor $K_a$.

4) The repulsive forces to avoid obstacle ($\vec{F}_{obs}$)

Considering that there may exist obstacles which can be de-
dected by using the sensors (such as radars) or the geometrical
information, each UAV should also be affected by a repulsive
force (denoted by $\vec{F}_{obs}$) from the obstacle once they are closer
than a safe distance, in order to avoid hitting the obstacle. We
also adopt the universal gravity to model $\vec{F}_{obs}$ as follows,
\[
\vec{F}_{obs} = \begin{cases} 
k_r \times (1/d_{obs})^2 , & d_{obs} < d_{safe} \\
0 , & \text{others}
\end{cases},
\]
(11)
where $d_{safe}$ is a predefined safe distance to the obstacle and
$d_{obs}$ is the current distance between the UAV and the obstacle.
The direction of this repulsive force is always to push the UAV
away from the obstacle.

The general principles to set the three force factors, i.e.,
$K_a$, $k_a$ and $K_r$ are as follows. $K_a$ is bigger than $k_a$, so that
all the UAV will head for the hotspots generally at first; $K_r$
is bigger than $K_a$ so that the too-close neighboring UAVs could
be pushed away immediately to avoid collisions. In summary,
we have $K_r > K_a > k_a$.

B. Motion control algorithm

The resultant force that affects any UAV can be calculated
by the following vector addition,
\[
\vec{F} = \vec{F}_a + \vec{F}_r + \vec{F}_{obs}.
\]
(12)
And the resultant force contains each UAV’s relations with
the hotspots, the neighbor UAVs, the UEs and the obstacles,
which contributes to the optimal deployment with collision
and obstacle avoidance.

According to Newton’s second law of motion, the change
of velocity in $\Delta t$ seconds is governed by the resultant force
acting on UAV $i$ (denoted by $\vec{F}_i$), and that is
\[
\Delta \vec{v}_i = \frac{\vec{F}_i \cdot \Delta t}{m} = \vec{F}_i \cdot \Delta t,
\]
(13)
where \( m \) is the virtual mass of the UAV and is normalized as 1 in this research.

Specifically, the velocity is controlled iteratively, for instance, once a second (\( \Delta t=1s \)). In each iteration, UAV \( i \) computes the resultant force \( \vec{F}_i \) according to (12), and then changes its velocity. According to (13), the velocity change is larger if \( \vec{F}_i \) has larger magnitude, and vice versa. Since the magnitude of \( \vec{F}_i \) varies from 0 to \( +\infty \), it is necessary to map this magnitude to a finite velocity from 0 to \( V_i \), where \( V_i \) is the maximum speed of UAV \( i \). To perform this mapping, we select the trigonometric function \( \arctan() \) so that the moving velocity is given by

\[
\vec{v}_i = \arctan(\vec{F}_i) \times 2/\pi \times V_i. \quad (14)
\]

This transformation is reasonable based on the following two facts. First, the transfer function is an increasing function. Second, when the force has larger magnitude, the moving distance is less sensitive to any further increase in the force magnitude.

Therefore, each UAV can just adjust its velocity iteratively according to (14), and specifically, this distributed motion control is done in three stages as below.

1) **Initialization**: Each UAV is informed about the center coordination of AoI or the hotspots if there are in AoI.

2) **Motion control**: Motion control is carried out iteratively. In each iteration, any UAV (say UAV \( i \)) broadcasts a HELLO message which contains its position information. Its one-hop neighbors can receive this message to get UAV \( i \)'s position information and similarly UAV \( i \) can obtain the position information of all its one-hop neighbors. Using the obtained local information, UAV \( i \) calculates the resultant force according to (12) and then get the desired velocity according to (14). The above steps are repeated until the stop condition is satisfied (see below).

3) **Stop condition**: If the UEs are static, the motion will stop when the resultant force on every UAV is zero in ideal condition. But it may take a large number of iterations before the resultant force is perfectly zero. We therefore specify that UAV \( i \) stops moving when the following two conditions are satisfied: (i) the resultant force is smaller than a threshold \( F_{th} \); and (ii) UAV \( i \) has at least two neighbors, i.e., \( C7 \) in (6).

The detailed motion control algorithm for UAV \( i \) is given in Algorithm 6, in which the control algorithm is composed of three parallel threads: Threads A, B and C. Thread A and Thread C are two iteration processes which run respectively in the communication module and in the control module. Therefore, they run independently, and their iteration intervals can be different. Thread B is an interrupt process, triggered by the reception of a HELLO packet.

### VI. Simulation Validations and Discussions

#### A. Simulation setup

We use JAVA 1.6 to implement the proposed algorithms as a simulator which shows the dynamics of the deployment process. Readers can download the simulator and demo videos and from [43]. In the evaluation, we consider 200 UEs distributed in a 2000m×2000m AoI. And to evaluate the scalability of the proposed algorithms, we consider varied network scenarios, which are explained below.

1) **Distribution of UEs**: We adopt two typical traffic distribution patterns, i.e., the random pattern (where the UEs are randomly distributed in the AoI) and the clustered pattern (where the UEs are divided into several clusters).

2) **Motion of the UEs**: We consider the static scenario and the dynamic scenario. In the static scenario, all the UEs do not move. In the dynamic scenario, we consider two cases. First, the UEs are moving to form a clustered pattern from a random pattern, under which case UEs are moving along quite different routes. Second, a cluster of UEs is moving from one place to another as a group, under which case UEs are moving along similar routes.

3) **Obstacles**: We consider both environments with and without obstacles in the AoI.

4) **Initial states of UAVs**: In the distributed motion control algorithm, the initial positions of UAVs will affect the deployment results. Therefore, we also differentiate the initial state where all UAVs are scattered over the AoI from the initial state where they are gathered at a base.

Primary simulation parameters are listed in Table III [3], [6], [20].

#### B. Random pattern with no obstacles (Figures 5 and 6)

Fig. 5 shows the iteration process and the final deployment results when the centralized algorithm is used. Fig. 5a depicts a random distribution of UEs and 33 initial candidate UAV locations under the random pattern (the 33 candidate UAV
TABLE III
PRIMARY SIMULATION PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum UEs served by one UAV</td>
<td>(N_{U}^{\text{max}})</td>
<td>20</td>
</tr>
<tr>
<td>Height of UAVs</td>
<td>(h)</td>
<td>100 m</td>
</tr>
<tr>
<td>Communication range of UAVs</td>
<td>(R_c)</td>
<td>500 m</td>
</tr>
<tr>
<td>Sense range of UAVs</td>
<td>(R_s)</td>
<td>500 m</td>
</tr>
<tr>
<td>Desired value of neighboring distance</td>
<td>(R_{\text{opt}})</td>
<td>230 m</td>
</tr>
<tr>
<td>Maximum Tx power of UAV</td>
<td>(p_{i,k})</td>
<td>40 dBm</td>
</tr>
<tr>
<td>Maximum speed of UAV</td>
<td>(V_i)</td>
<td>10 m/s</td>
</tr>
<tr>
<td>Carrier frequency of UAV-to-UE channel</td>
<td>(f)</td>
<td>2 GHz</td>
</tr>
<tr>
<td>(\alpha)</td>
<td></td>
<td>9.6</td>
</tr>
<tr>
<td>(\beta)</td>
<td></td>
<td>0.28</td>
</tr>
<tr>
<td>Additional pathloss under LoS</td>
<td>(\eta_{\text{LoS}})</td>
<td>1 dB</td>
</tr>
<tr>
<td>Additional pathloss under NLoS</td>
<td>(\eta_{\text{NLoS}})</td>
<td>20 dB</td>
</tr>
<tr>
<td>Power of Gaussian white noise</td>
<td>(N_{\text{gw}})</td>
<td>-174 dBm/Hz</td>
</tr>
<tr>
<td>SINR threshold</td>
<td>(\eta_{\text{th}})</td>
<td>-7 dB</td>
</tr>
<tr>
<td>Coverage outage proportion</td>
<td>(\tau)</td>
<td>0.02</td>
</tr>
<tr>
<td>Attractive force factor towards the center</td>
<td>(K_a)</td>
<td>800</td>
</tr>
<tr>
<td>UE attractive force factor</td>
<td>(k_a)</td>
<td>7</td>
</tr>
<tr>
<td>Repulsive force factor</td>
<td>(K_r)</td>
<td>3500</td>
</tr>
</tbody>
</table>

locations are dense enough to cover all the UEs). Fig. 5(b-c) show the dynamic process of deleting redundant UAVs, where the deleted UAVs are denoted by yellow triangles while the remained ones are green (for brevity, the links among active UAVs are not plotted). We can see that with the proposed algorithm, redundant UAVs are deleted iteratively. For instance, 15 UAVs have been deleted after the 40th iteration, see Fig. 5c. After 81 iterations, the algorithm stops and therefore outputs the final deployment results as shown in Fig. 5d. In Fig. 5d we show both the connections among UAVs (green line) and the associations between each UAV and the UEs associated with it (red line) but leave out all the deleted UAVs for a clearer illustration. It shows that to cover all these 200 UEs we need only 13 UAVs. And we can also see that all the UEs are associated with a UAV, any UAV is associated with no more than 20 UEs (which is the predefined threshold, see \(M_{U}^{\text{max}}\) in Table III) and all the UAVs are bi-connected.

For the same UE distribution as in Fig. 5, Fig. 6 shows the dynamic motion process of 20 UAVs governed by the distributed motion control algorithm. We can see that with the effects of attractive forces to UEs and repulsive forces between too-close UAVs, the distributed algorithm can also obtain the same result as that obtained by the centralized algorithm, i.e., each UE is covered by a UAV and the UAVs are bi-connected. An interesting phenomenon observed from the final result (Fig. 6d) is that the airborne network mainly consists of triangle connections, which is robust and at the same time can maximize the coverage.

The difference between these two algorithms is that in the centralized algorithm, the candidate UAVs are fixed, and the algorithm iteratively deletes redundant UAVs with global optimization, while in the distributed algorithm, the UAVs are not static, instead, they move with the effect of the resultant force calculated by local information and no UAVs are deleted. Therefore, these two algorithms are applicable to different scenarios. The centralized one is applicable to the scenario where the minimum number of UAVs is required to provide desirable services for already known on-ground UEs. Differently, the distributed one is applicable to the scenario where given number of UAVs try to cover UEs without their specific position information.

C. Clustered pattern with no obstacles (Figures 7 and 8)

In this subsection, we consider the clustered pattern where there are 3 clusters with respective 60, 70 and 70 UEs. In the clustered pattern, the UEs are more unevenly distributed than in the random pattern. Therefore, it is tougher to keep the UAV bi-connected. The results presented in Fig. 7 and Fig. 8 validate that the proposed algorithm can also work well with UEs under clustered pattern.

Fig. 7(a-c) illustrate the iteration process under the clustered traffic pattern when the centralized algorithm is used. Note that in Fig. 7b, although there are several UAVs with zero degree during the first iteration, at most one idle UAV can be deleted in each iteration to avoid the potential information-isolated problem. And Fig. 7d shows the final deployment results, which shows that all the UEs are covered and the UAVs are bi-connected (three clusters of UEs are connected together by UAV 18 and UAV 30 whose degrees are zero). Comparing Fig. 7d with Fig. 5d we find that the clustered pattern usually needs more UAVs (in this case, 14 UAVs) than the random pattern (13 UAVs) although the number of UEs are the same. This is caused by the uneven distribution of UEs and the requirement to guarantee connectivity among clusters.

Fig. 8 shows the dynamic motion process of 20 UAVs governed by the distributed motion control algorithm, with the same UE distribution as in Fig. 7. It demonstrates that the distributed algorithm can also work well under this clustered pattern distribution. Similar to Fig. 6, we can find that UAVs in each cluster form a hexagon network, which improves the load balance. Moreover, triangle structure of connection line between hotspots enhances the robustness of the whole airborne network.

D. Obstacle avoidance (Figures 9 and 10)

In this subsection, we evaluate the proposed algorithms when there exists an obstacle. For the centralized algorithm, we assume that the position of the obstacle is known, and therefore the obstacle can be easily avoided by not setting candidate UAVs in the obstacle area (see Fig. 9a). However, in the distributed algorithm, UAVs can only avoid the obstacle when encountering it by the effect of repulsive force from the obstacle.

In this evaluation, we only report the results with random pattern UEs in Fig. 9 and Fig. 10 (similar results can be obtained with the clustered UEs). The experiment results show that both algorithms can avoid the obstacle while maintaining the desirable properties in the final deployment results: all the UEs are covered and the UAV airborne network is bi-connected.

E. Distributed motion control when UAVs are initially gathered at a base airport (Figure 11)

In the former evaluations, we all consider that the UAVs are initially scattered throughout the AoI for distributed motion
control. This may happen when we need on-demand coverage just after the UAVs complete a task with scattered network topology. In this part, we consider another realistic scenario where all the UAVs are initially located at a base airport.

Fig. 11 illustrates the dynamic process of the distributed motion control when 20 UAVs depart from a base (which is located at the center of the AoI) to cover 200 randomly distributed UEs. Fig. 11(a-c) show that with the distributed motion control algorithm, UAVs disperse immediately and fly towards uncovered UEs, with the effects of the repulsive forces from other UAVs and the attractive forces from UEs nearby. And Fig. 11d plots the final deployment of UAVs, as well as the associations between them and the on-ground UEs. This experiment validates that the distributed algorithm can also suit the initial scenarios where the UAVs are gathered at a base.

F. Covering dynamic UEs

UEs may move on the ground as individuals or as a group. Under these dynamic scenarios, it is desirable that UEs are continuously tracked and covered by the flying UAVs. Therefore, in this subsection, we check if the proposed distributed motion control algorithm can properly track and cover the dynamic UEs. We first consider the scenario where UEs are moving to form a clustered pattern from a random pattern, under which case UEs are moving along quite different routes. And this is corresponding to the UEs’ grouping procedure. Second, we consider that the UEs are moving from one place...
to another as a group, under which case UEs are moving along similar routes. This is corresponding to the global mitigation of UEs.

1) **UEs are moving to form a clustered pattern from the random pattern (Figure 12)**

In the initial state, UEs are all distributed randomly in the AoI and covered by the bi-connected UAVs in the air (Fig. 12a). And then UEs move with the constant speed of 5 m/s to their respective final positions generated in advance under the clustered pattern and finally form three clusters (Fig. 12b-d). We can see that through the whole dynamic process the following desirable network properties are maintained: (i) all the UEs are covered and (ii) the UAVs are bi-connected. It is also notable that with the proposed algorithm, the UAV airborne network is comprised of roughly triangular structures, enhancing the robustness and stabilization of the whole topology.

2) **A cluster of UEs are moving under the clustered pattern as a group (Figure 13)**

Fig. 13 shows the deployment process when a cluster of UEs are moving from a place to another. Fig. 13a shows the initial distribution of 200 UEs in three clusters and 20 randomly distributed UAVs, which are exactly the same as that in Fig. 8a. And then the UE cluster on the top starts to move as a group rightwards at the constant speed of 5 m/s while the other two clusters do not move (Fig. 13b-d). And at t=200 s the moving UEs stop around their destination (Fig. 13d) and the UAVs stop moving 194 seconds later to form the desirable on-demand coverage (Fig. 13e).

These two experiments show that the proposed distributed motion control algorithm can also apply to dynamic scenarios where UEs are moving.

**G. Performance metric statistics**

In this subsection, we report the performance statistics of the proposed algorithms to give a deeper look at the scalability of the algorithms. In these performance statistics, we assume that there are no obstacles in the area, and the UE distributions under the random pattern and the clustered pattern are the same as before (i.e., Fig. 5a and Fig. 7a respectively).

On one hand, since the proposed algorithms both complete the on-demand deployment step by step, we care about the time consumed by these two algorithms. On the other hand, these two algorithms are designed for different application scenarios with different objectives and requirements, therefore they are also evaluated with different performance metrics, which are described as follows:

i) **Time consumed**: For the centralized algorithm, it is evaluated in terms of number of iterations \(N_{\text{iter}}\), while for the distributed algorithm it is evaluated via motion control time \(T_{\text{iter}}\).

ii) **The number of finally deployed UAVs \(N_f\)**: The main goal of the centralized algorithm is to obtain the deployment with minimum UAVs, therefore, we need to check the number of finally deployed UAVs. This indicates the effectiveness and the stability of the centralized algorithm.

iii) **Motion fairness**: This metric is to check the travel variance among UAVs in the distributed motion control, and it is defined by the Jain’s fairness index

\[
P_m = \frac{(\sum_{i=1}^{N_d} D_{im}^i)^2}{N_d \cdot \sum_{i=1}^{N_d} (D_{im}^i)^2},
\]

where \(U_Es\) are moving.
where $D^i_m$ is the moving distance of UAV $i$ throughout the algorithm. Note that the value of $\Psi_m$ varies from $1/N$ to 1 and is maximized when the moving distances of all the UAVs are the same.

iv) Coverage fairness ($\Psi_c$): It indicates the fairness of UAVs serving UEs. If a UAV covers much more UEs than the others, it will also consume energy faster and hence limit the overall network lifetime. Therefore, we use metric $\Psi_c$ to check if all the UAVs serve similar amount of UEs, and $\Psi_c$ is also defined according to the Jain’s fairness index as in (15).

1) Performance of the centralized algorithm

In Fig. 14a, we plot the dynamic output of the centralized algorithm, i.e., the number of active UAVs ($N_a$) and total number of served UEs ($M_s$), in the iterative process. There are 200 UEs in AoI and initial 33 candidate UAVs (see Fig. 5 and Fig. 7). We can see that the number of remaining UAVs converges to 13 and 14 respectively under the random pattern and the clustered pattern, which are the final deployment results. The flat periods of the blue dotted curve in Fig. 14a indicate that at these iterations idle UAVs have failed to be deleted due to the coverage outage or information-isolated island problem. However, through the retrospecting process, further redundant UAVs can be found and then deleted. Fig. 14a also shows the variation of the total number of covered UEs in each iteration. We can observe that the coverage outage is satisfied all the time although the number of active UAVs deceases.

In the centralized algorithm, we request that there are enough initial candidate locations so that the algorithm can converge to the optimal deployment with minimum UAVs. But how the number of initial UAV candidate locations affect the final results? We therefore further check the effect of this initial number to the final results in Fig. 14b, which plots the number of finally deployed UAVs ($N_f$) and iterations ($N_{iter}$) varying with the number of initial candidate UAV locations ($N$) (the result is the average value of 500 simulations). From Fig. 14b, we can see that the number of iterations increases with the number of initial candidate UAV locations, but the number of finally deployed UAVs has little variation. From Fig. 14b, we can also see that the number of finally deployed UAVs under both patterns is slightly more than the theoretical bound of the minimum deployed UAVs (200/20=10). This is resulted from the uneven distribution of UEs in our simulation (some UAVs are needed to keep the connectivity of the airborne network even if it has no UEs to cover). The results in Fig. 14b validate that when there are enough initial candidate UAV locations the final deployment result is affected little by the specific initial number, which indicates the feasibility and stability of the centralized algorithm.

2) Performance of the distributed algorithm

In the performance evaluation for the distributed algorithm, we consider 13-30 UAVs covering 200 UEs under four cases:
(i) Random UEs with scattered UAVs similar to Fig. 6a, where UEs are randomly distributed in the AoI and UAVs are initially scattered over AoI in a random manner; (ii) Random UEs with gathered UAVs similar to Fig. 11a, where UEs are still distributed randomly while UAVs are initially gathered around a base which is located at the center of AoI; (iii) Clustered UEs with scattered UAVs similar to Fig. 8a, where UEs are distributed in three clusters and UAVs are initially scattered over AoI in a random manner; and (iv) Clustered UEs with gathered UAVs, where UEs are distributed in three clusters as those in Fig. 8a and UAVs are initially gathered around the base as those in Fig. 11a.

Fig. 15a shows the motion control time ($T_{iter}$) varying with the number of UAVs ($N_d$). We can see that the required control time does not increase with the number of UAVs. Specifically, the motion control time keeps stable when UAVs depart from a base, however, when UAVs are randomly distributed, the required control time decreases with $N_d$ when $N_d \leq 20$ and keeps stable when $N_d > 20$. This is because that when the number of UAVs is too small (less than 20), scattered UAVs will travel long distances to achieve on-demand coverage and bi-connection for a certain traffic pattern, which results in longer control time. Moreover, we can see that the algorithm performs better (i.e., consumes less time) when UEs are randomly distributed than when UEs are distributed in the clustered pattern. This is because that with 200 UEs in the 2000m×2000m AoI, the UEs are distributed more unevenly in the clustered pattern than in the random pattern.

Fig. 15b gives the results of the motion fairness ($\Psi_{m}$) versus the number of UAVs, from which we can see that the algorithm is fair to UAVs (the fairness index is more than 0.97) in that all UAVs move almost the same distance. This is resulted from the nature of the virtual forces that drag each UAV to its respectively nearest proper position.

3) Performance comparison of the two algorithms

In both the centralized algorithm and the distributed algorithm, we consider the SINR exceeding a threshold as the criteria of any UE to access a UAV, please see Equation (5), which means that the transmit rate of all covered UEs can be satisfied. As an example, in Fig. 16, we illustrate the SINR distribution under the UAV deployment of Fig. 8d. We can see that the SINR distribution is coherent with the distribution of UEs (which are represented by little red ‘+’) and therefore can formulate the on-demand coverage. To evaluate the coverage property of the proposed algorithms, we define that the proportion of uncovered UEs should be restricted below a predefined threshold $\tau$, which guarantees that the probability of UE coverage is at least $1 - \tau$. We have
plotted the statistics result of the coverage probability with 500 simulations in Fig. 17, where we set $\tau = 0.02$. It shows that the statistical coverage probability of both algorithms can satisfy the requirement on coverage probability, i.e., higher than 98%. Specifically, the proportion of covered UEs is 99% in average for the centralized algorithm and 99.7% in average for the distributed algorithm with different number of UEs under both two patterns.

Figure 18 gives coverage fairness ($\Psi_c$) of the finally deployed UAVs versus the number of UEs, where we set the 33 initial candidate locations as in Fig. 5a for the centralized algorithm and the 20 UAVs as in Fig. 6a for the distributed algorithm. (Note that the UAVs only acting as connections between clusters are excluded in the evaluation). From Fig. 18, we can draw the following conclusions: (i) Both algorithms can obtain satisfactory coverage fairness ($\Psi_c > 0.7$ in most cases); (ii) In general, $\Psi_c$ increases with the number of UEs, since the difference of the served UE numbers among UAVs is smaller when the number of UEs increases; (iii) The distributed algorithm can obtain higher $\Psi_c$ than the centralized algorithm. This is because that the centralized algorithm essentially selects the final deployed UAV from fixed candidate UAVs (i.e., the UAVs do not move), while in the distributed algorithm UAVs can move with the effect of virtual forces and therefore they have higher probability to be deployed right at the optimal positions.

VII. Conclusion

In this paper, we propose two UAV deployment algorithms, i.e., the centralized algorithm and the distributed algorithm, to achieve on-demand coverage and maintain interconnection among UAVs at the same time. The centralized algorithm adopts a heuristic method to select UAVs from all candidate UAV locations iteratively while jointly considering the connectivity among UAVs and the associations between the UAVs and UEs. This centralized algorithm is applicable to the scenario that requires the minimum number of UAVs to provide desirable service for known on-ground UEs. Differently, the distributed algorithm requires no global information of UEs but autonomously controls the motion of each UAV in a distributed manner with the effect of virtual forces. This distributed algorithm is applicable to the scenario that requires the optimal coverage by utilizing given number of UAVs without any UE information. Extensive simulations validate that both algorithms can achieve a bi-connected airborne network, satisfying on-demand coverage and the UEs QoS requirements, and they can both avoid obstacles that may exist in the AoI. Besides, the results of the centralized algorithm could be a referential baseline for the distributed algorithm in deciding how many UAVs should be released to do the autonomous searching and covering. The distributed algorithm can further apply to the dynamic scenarios where the on-ground UEs are moving.

References


[43] “UAV deployment demo,” https://pan.baidu.com/s/1o7He78i.

Haitao Zhao (SM’18) received his B.E., M.Sc. and Ph.D. degrees all from the National University of Defense Technology (NUDT), P. R. China, in 2002, 2004 and 2009 respectively. And he is currently an associate professor in the Department of Cognitive Communications at NUDT. Prior to this, he visited the Institute of ECIT, Queen’s University of Belfast, UK and Hong Kong Baptist University. His main research interests include cognitive radio networks and self-organized networks. He has served as a TPC member of IEEE ICC’14-17, Globecom’16-17, and guest editor for IEEE Communications Magazine. He is a senior member of IEEE.

Ghaith Alshamrani received the B.S. degree in computer engineering from Fahd Bin Abdulaziz University, AlUla, Saudi Arabia, in 2012, the M.S. degree in computer science from University of Nebraska-Lincoln, Lincoln, NE, USA, in 2014, and the Ph.D. degree in computer science from the University of Siegen, Siegen, Germany, in 2019. He is currently a postdoctoral researcher at University of Twente, Enschede, The Netherlands. His research interests include computer networks and cyber-physical systems.

Haijun Wang received the B.S. degree in internet of things engineering from Shandong University, Jinan, China, and the M.S. degree in information and communication engineering from National University of Defense Technology, Changsha, China. He is currently pursuing the Ph.D. degree with the department of cognitive communications, National University of Defense Technology, Changsha, China. His research interests include cognitive radio networks, unmanned aerial vehicle communications.

Weiyu Wu received the B.S. degree in communication engineering from the Hunan Normal University, Changsha, China. She is now pursing the M.S. degree with College of Electrical and Information Engineering, Hunan University, Changsha, China. She is also in the Joint Cultivation with the College of Electronic Science, National University of Defense Technology, Changsha, China. Her research interests aim at wireless communication networks.

Jibo Wei (M’04) received the B.S. and M.S. degrees from the National University of Defense Technology (NUDT), Changsha, China, in 1989 and 1992, respectively, and the Ph.D. degree from Southeast University, Nanjing, China, in 1998, all in electronic engineering. He is currently a Professor with the Department of Cognitive Communications, NUDT. His research interests include Software Defined Radios (SDR), Cognitive communications and networking, OFDM and MIMO, cooperative communications. Dr. Wei is a member of the IEEE Communication Society and also a member of the IEEE Vehicular Technology Society. He is a council member of the China Institute of Communications. He is also an editorial committee member of the Journal on Communications.