TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 22/24 pp589–604
DOI: 10.26599/TST.2023.9010074
Volume 29, Number 2, April 2024

IQABC-Based Hybrid Deployment Algorithm for Mobile Robotic Agents Providing Network Coverage

Shuang Xu, Xiaojie Liu*, Dengao Li*, and Jumin Zhao

Abstract: Working as aerial base stations, mobile robotic agents can be formed as a wireless robotic network to provide network services for on-ground mobile devices in a target area. Herein, a challenging issue is how to deploy these mobile robotic agents to provide network services with good quality for more users, while considering the mobility of on-ground devices. In this paper, to solve this issue, we decouple the coverage problem into the vertical dimension and the horizontal dimension without any loss of optimization and introduce the network coverage model with maximum coverage range. Then, we propose a hybrid deployment algorithm based on the improved quick artificial bee colony. The algorithm is composed of a centralized deployment algorithm and a distributed one. The proposed deployment algorithm deploy a given number of mobile robotic agents to provide network services for the on-ground devices that are independent and identically distributed. Simulation results have demonstrated that the proposed algorithm deploys agents appropriately to cover more ground area and provide better coverage uniformity.

Key words: wireless robotic networks; network coverage; deployment algorithm; improved quick artificial bee colony

1 Introduction

With the great prosperity and the increasing popularity of smart mobile devices, the network connecting these devices is getting more significant[1, 2]. When the terrestrial infrastructure is out of work due to the heavy damage in a sudden disaster, such as earthquake, flood, fire, and so on, it is important to deploy an emergency network to provide communication services timely and quickly. Moreover, when the communication links are heavily congested due to a large-scale event, such as a concert, a basketball game, and so on, a temporary network with low cost is required. In these circumstances, mobile robotic agents such as Unmanned Aerial Vehicles (UAVs) have been widely utilized to enhance wireless communication systems, since they are characterized by flexible placement and strong Line-of-Sight (LoS) communication links[3]. In addition, communications utilizing mobile robotic agents have been an important part in the Fifth Generation (5G) and beyond 5G wireless networks[4].

By deploying a group of mobile robotic agents flexibly and rapidly, we can form a Wireless Robotic Network (WRN) to provide wireless communication services anywhere and anytime. Mobile robotic agents have the potential to alleviate traffic congestion in mobile internet by working as floating relays, especially when there are numerous Internet of Things (IoT) devices transmitting messages simultaneously[5]. Additionally, working as flying base stations, mobile robotic agents can provide wireless communication with network performance gains for rescue vehicles to achieve disaster relief[6]. Therefore, the deployment of mobile robotic agents

• Shuang Xu and Dengao Li are with College of Computer Science and Technology (College of Data Science), Taiyuan University of Technology, Taiyuan 030024, China. E-mail: xiaoshuang_0320@163.com; lidengao@tyut.edu.cn.

• Xiaojie Liu is with Pengcheng Laboratory, Shenzhen 518055, China. E-mail: liuxiaojie0112@gmail.com.

• Jumin Zhao is with College of Information and Computer, Taiyuan University of Technology, Taiyuan 030024, China. E-mail: zhaojumin@tyut.edu.cn.

∗ To whom correspondence should be addressed.

Manuscript received: 2023-05-18; revised: 2023-07-01; accepted: 2023-07-20

attracted much attention as an effective approach to achieve rapid and efficient network coverage[7].

Many deployment algorithms have been proposed for mobile robotic agents to provide network coverage, which can be classified into three categories: centralized, distributed, and hybrid algorithms. Centralized algorithms rely on a few resourceful agents in computing, but suffer from low scalability, single high computation complexity, and individual communication overhead. Distributed algorithms can deploy agents without requiring prior knowledge, but may suffer from local optimization. As a result, centralized algorithms are typically employed when a resourceful satellite or agent is available to collect information about the mobile robotic agents and compute deployment for them. By contrast, distributed algorithms are preferred when it is infeasible for an agent to obtain the entire information. Hybrid algorithms can take advantage of the accuracy of centralized decision-making, as well as the coordination and adaptability among mobile robotic agents.

Therefore, based on the Improved Quick Artificial Bee Colony (IQABC) algorithm[8], we propose a hybrid deployment algorithms, including a centralized algorithm and a distributed one, to address the network coverage problem in a space-air-ground WRN. The contributions can be summarized as follows:

(1) We decouple the 3D deployment problem into vertical and horizontal dimensions without any loss of optimality and introduce both centralized and distributed network coverage problems. A hybrid deployment algorithm is proposed to deploy a given number of mobile robotic agents to improve the covered area even when the centralized controller is failure.

(2) For the situation where the distributed mobile agents are controlled with the assistance of a satellite, we propose an IQABC-based centralized coverage algorithm (IQABC-C) to deploy mobile robotic agents with the objective of minimizing the uncovered area and optimizing the coverage uniformity.

(3) When the satellite is failure, we model the coverage problem as a continuous programming problem and propose an IQABC-based distributed coverage algorithm (IQABC-D) to deploy each agent utilizing only local information iteratively.

(4) The proposed deployment algorithm can deploy agents to cover more area so that more devices can be covered and served. Coverage uniformity is considered in our proposed algorithms to deploy agents evenly and uniformly.

The rest of this paper is organized as follows. Section 2 reviews the related work. The problem formulation is described in Section 3. Section 4 presents the proposed hybrid deployment algorithm. Simulation results are presented in Section 5. Section 6 concludes this paper.

2 Related Work

In recent years, much attention have been paid to deploy mobile robotic agents to provide network coverage for ground users. The related works are summarized in terms of on-demand coverage, area coverage, and Swarm Intelligence (SI) based coverage.

2.1 On-demand coverage

From the perspective of providing coverage services for each ground users, many on-demand coverage deployment algorithms were proposed. In Ref. [9], the deployment of UAV base stations for achieving on-demand full communication coverage in wireless communication was studied, taking into account network robustness and service quality. The UAV deployment algorithm was proposed to minimize the number of UAVs required while achieving on-demand full coverage. Alzenad et al.[10] proposed an energy-efficient deployment algorithm that deploys agents at the optimal altitude by using the minimum required transmit power while maximizing the number of covered users. In Ref. [11], an on-demand coverage solution based on UAV base stations was proposed to tackle temporary wireless network congestion in tourism areas. An automatic UAV base station deployment algorithm was designed to determine the minimal number of UAV base stations and their two-dimensional coordinates simultaneously, addressing the challenge of irregular tourist distribution.

Lyu et al.[12] proposed a polynomial-time algorithm with successive agent deployment to minimize the number of agents required to provide wireless coverage for ground users. However, the user locations must be known in advance and the user mobility was not considered. A centralized algorithm and a distributed algorithm were proposed in Ref. [13] to deploy mobile robotic agents to provide on-demand service for ground user devices. On one hand, the centralized algorithm can provide desirable services with the minimum number of mobile robotic agents, but the positions of user devices have to be known in advance. On the other hand, the distributed algorithm can find user devices autonomously and provide them with on-demand network coverage. However, the deployment changes along with the mobility of user devices, leading to network instability especially when certain mobile devices move into or outside the target area.

2.2 Area coverage

Aiming at a target area where user devices are located, some researchers proposed algorithms to deploy mobile robotic agents to provide good services by improving the covered area. In Ref. [7], a distributed deployment algorithm was proposed for WRNs that utilizes Particle Swarm Optimization (PSO) and Voronoi diagram to optimize coverage area with a minimum number of agents. It only required local information and did not rely on any prior information. Trotta et al.[14] formulated a constrained coverage and persistence aerial network deployment problem and proposed two agent deployment approaches, a centralized optimal approach, and a distributed game theory based approach, to maximize the coverage area while scheduling the recharging operations of these agents. The centralized approach maximized the coverage area by deploying agents in regular hexagon patterns, while the distributed approach placed agents by utilizing virtual spring force to achieve regular hexagon patterns.

Ruan et al.[15] proposed an energy-efficient coverage deployment algorithm based on spatial adaptive play to deploy multiple agents to maximize the covered area and achieve power control. In Ref. [16], an emergency network coverage deployment problem was formulated and two algorithms were proposed to minimize the deployment delay till covering the whole target area for two dispatching scenarios, including dispatching agents from the same location and different locations, respectively. In Ref. [17], two fast UAV deployment problems were formulated to cover the target area, considering the delay among all agents for fairness and the total delay for efficiency. They also presented algorithms to address these two problems considering the aforementioned two dispatching scenarios.

2.3 Coverage based on swarm intelligence

Swarm intelligence optimization algorithms such as Genetic Algorithm (GA), PSO, Artificial Bee Colony (ABC), etc., have been utilized in addressing the coverage deployment problem. In Ref. [18], a novel genetic algorithm was developed to deploy heterogeneous sensors in Wireless Sensor Networks (WSN) to improve the coverage area. Reina et al.[19] proposed a Multi-Layout Multi-subpopulation Genetic Algorithm (MLMPGA) to solve the NP-hard problem of deploying UAVs for optimal coverage of ground nodes. The simulation demonstrated that the MLMPGA outperformed other well-known meta-heuristic optimization algorithms, such as PSO and GA, in terms of performance results. However, all these GA-based algorithms are centralized.

Cao et al.[20] proposed a PSO-based strategy to deploy heterogeneous directional sensors and relay nodes to maximize coverage area and prolong network lifetime in complex 3D industrial WSNs. In Ref. [21], a PSO-based energy efficient coverage control algorithm is proposed, which achieves a balance between coverage ratio and energy cost by adjusting the sensing radius of sensors in different grids. Du et al.[22] proposed a network-based heterogeneous particle swarm optimization algorithm to deploy UAVs for UAV downlink communication coverage, which achieved the deployment by maximizing the total required data rates of users.

Qasim et al.[23] proposed a modified Ant Colony Optimization (ACO) based framework for WSN deployment in a realistic 3-D environment, which outperformed ACO-based algorithms in terms of the size of the solution for node deployment. It achieved quick convergence and avoided the time overhead problem that arises in standard ACO-based algorithms in 3-D environments. In Ref. [24], a U-ABC algorithm is proposed to deploy agents to provide network access for user devices in a post-disaster environment. The U-ABC algorithm improved the network throughput and achieved a high user coverage rate with a given number of UAVs. Furthermore, the covered users are supposed to be static in this scenario.

As mentioned above, almost all the SI-based algorithms utilized in solving the coverage problem can solve the coverage problem effectively and efficiently. However, they are centralized, which are not suitable in distributed networks, especially when the centralized controller is failed. Considering both centralized and distributed algorithms have their advantages respectively, we will use SI algorithms for the coverage problem in both centralized and distributed way.

3 System Model and Problem Formulation

3.1 Network model

We consider a WRN providing communication coverage for ground user devices that are independent and identically distributed in a target region. As shown in Figs. 1a and 1c, there are n mobile UAV agents hovering above the target region. The satellite has a wide observation range and can dispatch UAVs to provide communication coverage for the target region. The UAV agents can be controlled by the satellite flying above. The satellite primarily serves as a centralized controller responsible for making deployment decisions for the agents when it operates effectively. The satellite controller runs the centralized deployment algorithm based on the collected information. However, in cases where UAV agent-to-satellite communications are interrupted due to severe weather, electromagnetic interference, or satellite failure, the UAV agents can operate in self-organized networking way. Each agent can obtain the local information by sensing and communication through one-hop connections. The UAV agents provide communication coverage for the target region by making deployment decisions through distributed collaboration among themselves. Important notations are shown in Table 1.

Suppose that the target area is a square area sized $L \times W$. The set of UAV agents is denoted by $\boldsymbol{N} = \{N_i(\boldsymbol{p}_i^{\mathrm{a}}, \boldsymbol{v}_i^{\mathrm{a}}) | i = 1, 2, \ldots, n\}$, where $\boldsymbol{p}_i^{\mathrm{a}} = (p_{i,1}^{\mathrm{a}}, p_{i,2}^{\mathrm{a}}, p_{i,3}^{\mathrm{a}})$ ($p_{i,1}^{\mathrm{a}} \in [0, L]$, $p_{i,2}^{\mathrm{a}} \in [0, W]$, and $p_{i,3}^{\mathrm{a}} \in [h_{\min}, h_{\max}]$) is the three dimensional position of agent N_i and $\boldsymbol{v}_i^{\mathrm{a}} = (v_{i,1}^{\mathrm{a}}, v_{i,2}^{\mathrm{a}}, v_{i,3}^{\mathrm{a}})(|\boldsymbol{v}_i^{\mathrm{a}}| \leqslant v_{\max}^{\mathrm{a}})$ is the velocity vector of agent N_i, and the speed of agent is supposed to be no greater than the maximum speed $v_{\max}^{\mathrm{a}}$. The Euclidean distance between agents N_i and N_j, denoted by d_{ij}^{a2a}, can be calculated as follows:

$$d_{ij}^{\mathrm{a2a}} = |\boldsymbol{p}_i^{\mathrm{a}} - \boldsymbol{p}_j^{\mathrm{a}}| \tag{1}$$

When providing communication coverage for ground user device, the UAV comprises two main parts of communication. One is the agent-to-user connections which is modeled as the coverage model, and the other is the agent-to-agent connections which is modeled as the communication model.

3.2 Coverage model

The coverage model presented in Ref. [25] is adopted, which describes the agent-to-user communication channels. Each UAV agent uses the Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) connections simultaneously. As illustrated in Fig. 1b, there is an agent N_i located at $\boldsymbol{p}_i^{\mathrm{a}} = (p_{i,1}^{\mathrm{a}}, p_{i,2}^{\mathrm{a}}, p_{i,3}^{\mathrm{a}})$ with altitude $p_{i,3}^{\mathrm{a}}$, and a user device U_k located at $\boldsymbol{p}_k^{\mathrm{g}} = (p_{k,1}^{\mathrm{g}}, p_{k,2}^{\mathrm{g}}, p_{k,3}^{\mathrm{g}})$ with altitude $p_{k,3}^{\mathrm{g}} = 0$. Let r_{ik}^{a2g} and $\theta_{ik}^{\mathrm{a2g}}$ denote the horizontal distance and the elevation angle (in radian) of agent N_i with respect to user device U_k, respectively. Therefore, $r_{ik}^{\mathrm{a2g}} = \sqrt{(p_{i,1}^{\mathrm{a}} - p_{k,1}^{\mathrm{g}})^2 + (p_{i,2}^{\mathrm{a}} - p_{k,2}^{\mathrm{g}})^2}$, $\theta_{ik}^{\mathrm{a2g}} = \arctan \frac{p_{i,3}^{\mathrm{a}}}{r_{ik}^{\mathrm{a2g}}}$, and $d_{ik}^{\mathrm{a2g}} = \sqrt{(r_{ik}^{\mathrm{a2g}})^2 + (p_{i,3}^{\mathrm{a}})^2}$.

Hence, the path loss between agent N_i and user device U_k can be calculated using Eq. (2).

$$L(p_{i,3}^{\mathrm{a}}, r_{ik}^{\mathrm{a2g}}) = \xi_{\mathrm{LoS}}^{i,k} \times L_{\mathrm{LoS}}^{i,k} + \xi_{\mathrm{NLoS}}^{i,k} \times L_{\mathrm{NLoS}}^{i,k} \tag{2}$$

where $L_{\mathrm{LoS}}^{i,k}$ is the path loss model for LoS connections, and it is calculated using Eq. (3)[10]. $L_{\mathrm{NLoS}}^{i,k}$ is the path

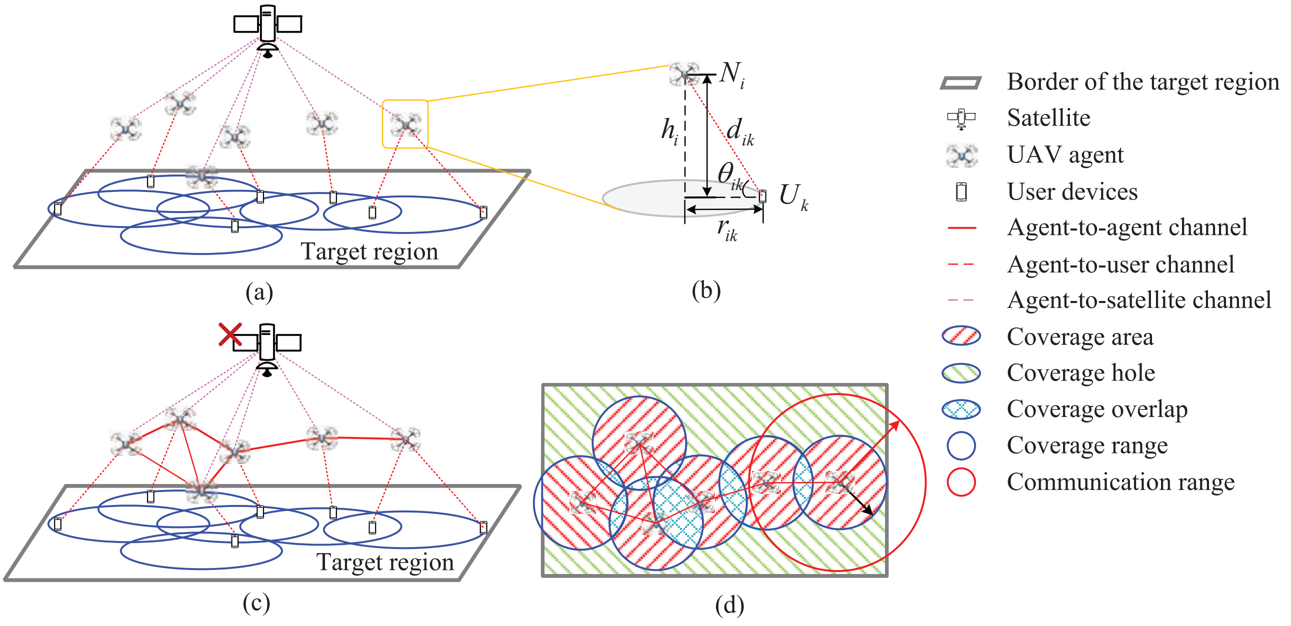


Fig. 1 Network model of a WRN providing network coverage.

Table 1 Important notations.

Notation	Description
$L \times W$	Size of target area
$\boldsymbol{N}$	Set of all UAV agents
$\boldsymbol{P}$	Set of positions of all UAV agents
$\boldsymbol{p}_i^{\mathrm{a}}$	Three dimensional position of agent N_i
$\boldsymbol{v}_i^{\mathrm{a}}$	Velocity vector of agent N_i
d_{ij}^{a2a}	Distance between agents N_i and N_j
r_{ik}^{a2g}	Horizontal distance of agent N_i with respect to user device U_k
$\theta_{ik}^{\mathrm{a2g}}$	Elevation angle of agent N_i with respect to user device U_k
$L_{\mathrm{LoS}}^{i,k}$	Path loss model for LoS
$L_{\mathrm{NLoS}}^{i,k}$	Path loss model for NLoS
$\xi_{\mathrm{LoS}}^{i,k}$	Probability of having a LoS connection
$\xi_{\mathrm{NLoS}}^{i,k}$	Probability of having an NLoS connection
f	Carrier frequency of agent-to-user channel
η_{LoS}	Average additional pathloss for LoS connection
η_{NLoS}	Average additional pathloss for NLoS connection
c	Speed of light
a and b	Constant values that depend on the environment
P_{t}	Signal transmission power of agent N_i
P_{r}	Received power of user device U_k
P_{th}	Receiving power threshold
L_{th}	Path loss threshold
R_{a2g}	Radius of the maximum coverage region of agent
h_{opt}	Altitude at which an agent can provide the maximum coverage range
f'	Carrier frequency of agent-to-agent channel
R_{a2a}	Maximum communication range
α_i^1	Local polygon area
α_i^2	Local polygon region within the proxy coverage region
α_i^3	Local polygon area outside the proxy coverage area
$\boldsymbol{V}_i$	Set of Voronoi vertices of agent N_i
$\boldsymbol{E}_i$	Set of Voronoi edges of agent N_i
k_i	Number of Voronoi vertices
$\boldsymbol{N}_i^v$	Set of neighbor Voronoi agents of agent N_i
$v_{\max}^{\mathrm{a}}$	Maximum speed of agents
$f_{\mathrm{h}}(\boldsymbol{N})$	Global hole ratio
$f_{\mathrm{u}}(\boldsymbol{N})$	Global coverage uniformity
δ_i	Standard deviation of distances between the i-th agent and its neighbor Voronoi agents
m_i	Mean value of distances between the i-th agent and its neighbor Voronoi agents
c_1 and c_2	Impact factors

loss model for NLoS connections, and it is calculated using Eq. (4)[13]. $\xi_{\mathrm{LoS}}^{i,k}$ is the probability of having an LoS connection, and it is calculated using Eq. (5). $\xi_{\mathrm{NLoS}}^{i,k}$ is the probability of having an NLoS connection, and it is calculated using Eq. (6).

$$L_{\mathrm{LoS}}^{i,k} = 20 \log d_{ik}^{\mathrm{a2g}} + 20 \log f + 20 \log \left(\frac{4\pi}{c}\right) + \eta_{\mathrm{LoS}} \tag{3}$$

$$L_{\mathrm{NLoS}}^{i,k} = 20 \log d_{ik}^{\mathrm{a2g}} + 20 \log f + 20 \log \left(\frac{4\pi}{c}\right) + \eta_{\mathrm{NLoS}} \tag{4}$$

$$\xi_{\mathrm{LoS}}^{i,k} = \frac{1}{1 + a^{-b(\theta_{ik}^{\mathrm{a2g}} - a)}} \tag{5}$$

$$\xi_{\mathrm{NLoS}}^{i,k} = 1 - \xi_{\mathrm{LoS}}^{i,k} \tag{6}$$

where f denotes the carrier frequency of the agent-to-user channel, η_{LoS} and η_{NLoS} are the additional path losses for LoS and NLoS connections, respectively, c is the speed of light, a and b are constant values that depend on the environment (rural, urban, dense urban, etc.). By substituting Eqs. (3)–(6) into Eq. (2), we obtain Eq. (7).

$$L(p_{i,3}^{\mathrm{a}}, r_{ik}^{\mathrm{a2g}}) = \frac{A}{1 + a^{-b(\frac{180}{\pi}\theta_{ik}^{\mathrm{a2g}} - a)}} + 20 \log \frac{r_{ik}^{\mathrm{a2g}}}{\cos \theta_{ik}^{\mathrm{a2g}}} + B \tag{7}$$

where $A = \eta_{\mathrm{LoS}} - \eta_{\mathrm{NLoS}}$ and $B = 20 \log \frac{4\pi f}{c} + \eta_{\mathrm{LoS}}$.

Assuming agent N_i transmits signal with power P_{t}, the received power P_{r} of user device U_k can be calculated using Eq. (8).

$$P_{\mathrm{r}} = P_{\mathrm{t}} - L(p_{i,3}^{\mathrm{a}}, r_{ik}^{\mathrm{a2g}}) \tag{8}$$

In order to have a guaranteed quality of service, the received power P_{r} must exceed a certain threshold P_{th}. That is, the user device U_k is covered if its path loss is no greater than a certain threshold L_{th}, i.e., $L(p_{i,3}^{\mathrm{a}}, r_{ik}^{\mathrm{a2g}}) \leqslant L_{\mathrm{th}}$. Obviously, the radius of the communication coverage region of agent N_i can be defined as $R_{\mathrm{a2g}} = r_{ik}^{\mathrm{a2g}}|_{L(p_{i,3}^{\mathrm{a}}, r_{ik}^{\mathrm{a2g}}) = L_{\mathrm{th}}}$. For a particular environment and a given agent at optimal altitude h_{opt} (i.e., the altitude at which an agent can provide the maximum coverage range), all the ground points located at coverage range $r < R_{\mathrm{a2g}}$ experience a path loss no greater than L_{th}[26]. The proof refers to the Appendix A.

3.3 Communication model

If the satellite controller is failed, the distributed communication among UAV agents is enabled to respond the changes of network status rapidly. The communication model describes the agent-to-agent communication channels. The agent communicates with each other without obstructions resulted by high buildings or some other obstacles. Thus, the agent-to-agent links are mainly dominated by LoS connections, which can be expressed as the Free Space Propagation Loss (FSPL). Hence, for any two agents $N_i \in \boldsymbol{N}$ and $N_j \in \boldsymbol{N}, j \neq i$, the path loss between them can be calculated as follows:

$$L_{\text{LoS}}(i,j)=20\log d_{ij}^{\text{a2a}}+20\log f'+20\log\left(\frac{4\pi}{c}\right) \quad (9)$$

where f' is the carrier frequency of agent-to-agent channel and c is the speed of light. The path loss increases with the increase in their distance. Similarly, given the maximum transmit power, the maximum communication range R_{a2a} can be determined[13]. Due to the constrained communication range, each agent N_i can only communicate with agents in its vicinity called neighbor agents, denoted by $N_{\text{eib}_i} = \{N_j | d_{ij}^{\text{a2a}} \leqslant R_{\text{a2a}}, j = 1,2,\ldots,n\}$.

3.4 Voronoi partition

To evaluate the covered area, the Voronoi partition[27] is utilized to cut up the target area into many subareas, also called Voronoi polygons. Figures 2a and 2b illustrate the instances of Voronoi partition and Voronoi polygon of agent N_i. After the Voronoi partition, each Voronoi polygon contains only one agent and any point inside this polygon is closer to its involved agent than to any other agent. For each agent N_i, the area of its corresponding Voronoi polygon is referred to as the local polygon area, which is denoted by α_i^1. Furthermore, the local coverage area and local coverage hole of this agent is the local polygon area inside and outside its coverage range respectively, which are represented by α_i^2 and α_i^3, respectively.

Let $\boldsymbol{V}_i = \{V_1, V_2, \ldots, V_{k_i}\}$ and $\boldsymbol{E}_i = \{E_1, E_2, \ldots, E_{k_i}\}$ represent the sets of Voronoi vertices and Voronoi edges of agent N_i, where k_i is the number of Voronoi vertices. Let $\boldsymbol{N}_i^{\text{v}}$ denoted the set of neighbor Voronoi agents of agent N_i. The agent N_i has k_i neighbor Voronoi agents. As illustrated in Fig. 2b, the agent N_i has 5 Voronoi vertices, so it also has 5 neighbor Voronoi agents.

3.5 Problem formulation

We aim to use a given number of mobile robotic agents to provide a target region with enough and uniform coverage. Assume that: (1) all the agents are identical and have the same coverage range R_{a2g} and radio communication range R_{a2a}; (2) There are no obstacles at the optimal altitude h_{opt} so that an agent can be deployed at any location in the horizontal dimension. Consequently, this problem can be simplified into a two-dimensional one with all mobile robotic agents flying at the same fixed optimal altitude[26, 28].

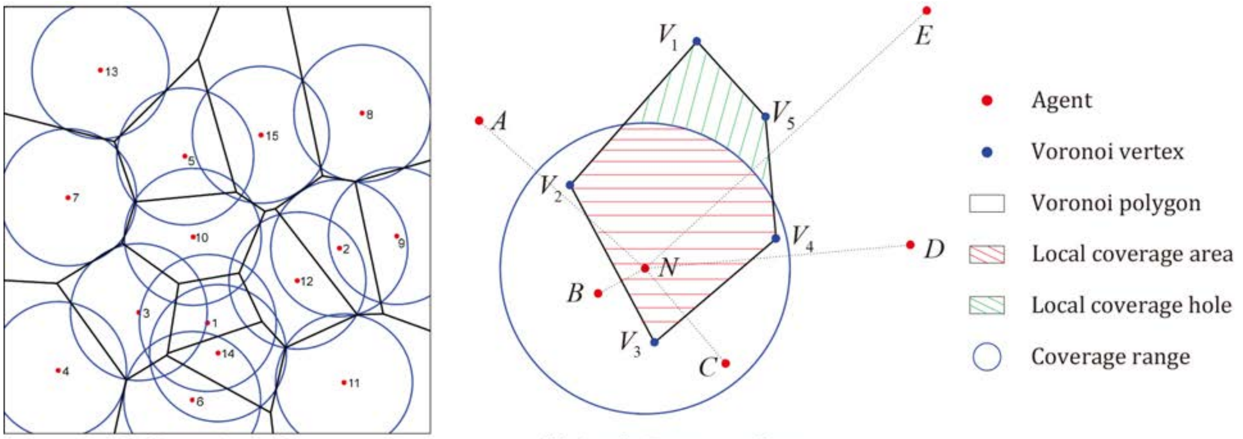
The following definitions are given to evaluate coverage area and coverage uniformity.

Definition 1 (global hole ratio). The global hole ratio, denoted by $f_{\text{h}}(\boldsymbol{N})$, is the ratio of the sum of all local coverage holes to the target area, which is formulated as Eq. (10).

$$f_{\text{h}}(\boldsymbol{N}) = \frac{\sum\limits_{i=1}^{n} \alpha_i^3}{L \cdot W} \quad (10)$$

Definition 2 (global coverage uniformity). The global coverage uniformity, denoted by $f_{\text{u}}(\boldsymbol{N})$, is calculated by the standard deviation of the distance between every two neighbor Voronoi agents, which is formulated as Eqs. (11)–(13).

$$f_{\text{u}}(\boldsymbol{N}) = \frac{1}{n}\sum_{i=1}^{n} \delta_i \quad (11)$$



(a) Voronoi partition

(b) Single Voronoi polygon

Fig. 2 Voronoi partion and Voronoi polygon.

$$\delta_i = \sqrt{\frac{1}{k_i}\sum_{j=1}^{k_i}(d_{ij}^{\mathrm{a2v}} - m_i)^2} \tag{12}$$

$$m_i = \frac{1}{k_i}\sum_{j=1}^{k_i} d_{ij}^{\mathrm{a2v}} \tag{13}$$

where δ_i and m_i are the standard deviation and mean value of distances between the i-th agent and its neighbor Voronoi agents, respectively. Additionally, the global coverage uniformity $f_{\mathrm{u}}(\boldsymbol{N})$ is normalized within the range [0, 1] through Eq. (14).

$$f_{\mathrm{u}}^{'}(\boldsymbol{N}) = 1 - \mathrm{e}^{-f_{\mathrm{u}}(\boldsymbol{N})} \tag{14}$$

For optimizing the network coverage, we take both the coverage area maximization and coverage uniformity into consideration. Therefore, the global hole ratio and global coverage uniformity are combined to construct an objective function, as shown in Eq. (15). The positions of agents $\boldsymbol{P} = \{\boldsymbol{p}_i^{\mathrm{a}} | i = 1, 2, \ldots, n\}$ are optimized as follows:

$$\min_{\boldsymbol{P}} \ c_1 \cdot f_{\mathrm{h}}(\boldsymbol{N}) + c_2 \cdot f_{\mathrm{u}}'(\boldsymbol{N}) \tag{15}$$

$$\begin{aligned}
\text{s.t.,}\quad & \boldsymbol{v}_i^{\mathrm{a}} < v_{\max}^{\mathrm{a}}, && i = 1, 2, \ldots, n && (16)\\
& 0 \leqslant p_{i,1}^{\mathrm{a}} \leqslant L, && i = 1, 2, \ldots, n && (17)\\
& 0 \leqslant p_{i,2}^{\mathrm{a}} \leqslant W, && i = 1, 2, \ldots, n && (18)\\
& p_{i,3}^{\mathrm{a}} = h_{\mathrm{opt}}, && i = 1, 2, \cdots, n && (19)
\end{aligned}$$

where c_1 and c_2 are their impact factors, and $c_1 + c_2 = 1$. Since the coverage area is more significant than coverage uniformity, c_1 and c_2 are set to be 0.9 and 0.1, respectively.

4 IQABC-Based Hybrid Deployment Algorithm

For the communication coverage with a centralized satellite controller, we model it as an optimization problem to maximize the covered area. However, when the centralized satellite controller is failed, the distributed communication coverage is modeled as a continuous programming problem to deploy each agent step by step. The ABC algorithm was successfully applied to achieve multivariable numerical and combinatorial optimization efficiently, and it has the advantages of easy implementation and fewer requirement of control parameters[29, 30]. Moreover, to further improve the stand ABC algorithm, the improved and enhanced variants, such as Quick ABC (QABC) algorithm[31] and IQABC algorithm[8], have been proposed. Compared with other variants, the IQABC algorithm performs best in terms of convergence speed and solution quality[8]. Thus, the IQABC is exploited to design a hybrid deployment algorithm to address the above two problems. The hybrid deployment algorithm is composed of an IQABC-based centralized deployment algorithm and an IQABC-based distributed deployment algorithm. They are exploited to solve the coverage deployment with centralized controller and without centralized controller, respectively.

4.1 IQABC algorithm

There are four fundamental phases in the IQABC algorithm, i.e., (1) initialization phase, (2) employed bee phase, (3) onlooker bee phase, and (4) scout bee phase.

4.1.1 Initialization phase

In the initialization phase, a set of food sources formed as $X = \{x_{ij} | i = 1, 2, \ldots, \mathrm{sn} \text{ and } j = 1, 2, \ldots, D\}$ are initialized utilizing Eq. (20).

$$x_{ij} = x_j^{\min} + \phi_{ij} \cdot (x_j^{\max} - x_j^{\min}) \tag{20}$$

where sn is the number of initialized food sources and D is the dimension of each food source, $x_j^{\min}$ and $x_j^{\max}$ are the lower and upper bounds for the j-th dimension, respectively, and ϕ_{ij} is a random number within the range [0, 1]. The initialized food sources are evaluated by utilizing the fitness function shown in Eq. (21).

$$\mathrm{fit}(x_i) = \frac{1}{1 + f(x_i)}, \quad f(x_i) \geqslant 0 \tag{21}$$

where $f(x_i)$ is the objective function value for a minimization problem of food source $x_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$. Then, the food source with the largest fitness value is selected as the global best food source, and it is recorded as x_{b}. The index of global best food source is recorded as i_{b}, and the total number of evaluations N_{eval} is initialized as 1.

Furthermore, counters t_{EBP} and t_{OBP} are defined to record the times that the global best food source has not improved in the employed bee phase and in the onlooker bee phase, respectively, t_{EBP} and t_{OBP} are initialized to 0. Counters of food sources that record the times each food source has not improved, denoted by $t_i, i = 1, 2, \ldots, \mathrm{sn}$, are also initialized to 0. Particularly, control parameters L_{best} and L_{init} are initialized as $2 \cdot \mathrm{sn} \cdot D$ and $\mathrm{sn} \cdot D$, respectively. The former control parameter is utilized to explore new food sources in the employed and onlooker bee phases, while the latter one is utilized to abandon exhausted food sources in the scout bee phase.

4.1.2 Employed bee phase

In the employed bee phase, each employed bee $i, (i = 1, 2, \ldots, \text{sn})$ is only responsible for one food source, and attempts to find a new food source $v_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]$ depending on its corresponding food source x_i and the counter t_{EBP}. If $t_{\text{EBP}} \leqslant L_{\text{best}}$, the employed bee i will explore the global best food source x_{b} to produce a candidate source using Eq. (22).

$$v_{ij} = x_{\text{b}j} + \omega \cdot (x_{\text{b}j} - x_{ij}) \tag{22}$$

where v_{ij}, $x_{\text{b}j}$, and x_{ij} are the j-th dimension of the new food source, the global best food source, and the food source, respectively. Note that the parameters of v_i are the same with x_{b} except the j-th dimension. ω is random number within the range $[-1, 1]$. If $t_{\text{EBP}} > L_{\text{best}}$, the employed bee i will explore its corresponding food source x_i to produce a candidate source using Eq. (23).

$$v_{ij} = x_{ij} + \omega \cdot (x_{ij} - x_{kj}) \tag{23}$$

where j and k are randomly selected from D dimensions and sn food sources, and the remarkable thing is that k must be different from i, i.e., $j \in \{1, 2, \ldots, D\}$, $k \in \{1, 2, \ldots, \text{sn}\}$, and $k \neq i$. Note that the parameters of v_i are the same with x_k except the j-th dimension. The fitness value of the food source v_i is caculated using Eq. (21). If $\text{fit}(v_i) \geqslant \text{fit}(x_{\text{b}})$, replace the global best food source x_{b} with the new food source v_i, and reset the counter $t_{i_{\text{b}}}$ to 0, i.e., $t_{i_{\text{b}}} = 0$. Otherwise, if $\text{fit}(v_i) < \text{fit}(x_{\text{b}})$, keep the global best food source x_{b} unchanged, and add one to the counters $t_{i_{\text{b}}}$ and t_{EBP}, i.e., $t_{i_{\text{b}}} = t_{i_{\text{b}}} + 1$ and $t_{\text{EBP}} = t_{\text{EBP}} + 1$. If $\text{fit}(v_i) \geqslant \text{fit}(x_i)$, replace the food source x_i with the new food source v_i, and reset the counter t_i to 0, i.e., $t_i = 0$. Otherwise, if $\text{fit}(v_i) < \text{fit}(x_i)$, keep the food source x_i unchanged, and add one to the counter t_i, i.e., $t_i = t_i + 1$. Particularly, the employed bee phase is performed for each employed bee, only when the termination criteria are not reached.

4.1.3 Onlooker bee phase

In the onlooker bee phase, the selection possibility of each food source x_i, denoted by $\text{pro}(x_i)$, is firstly determined using Eq. (24).

$$\text{pro}(x_i) = \frac{\text{fit}(x_i)}{\sum\limits_{j=1}^{\text{sn}} \text{fit}(x_j)} \tag{24}$$

Each onlooker bee $i, (i = 1, 2, \ldots, \text{sn})$ selects food sources depending on the calculated selection possibilities. The roulette wheel selection scheme is utilized in the selection. That is, each onlooker bee i generates a random number pro_{r} within the range $[0, 1]$, if $\text{pro}_{\text{r}} \leqslant \text{pro}(x_i)$, food source x_i is selected and onlooker bee m will produce a new food source v_i depending on the selected food source x_i and the counter t_{OBP}. If $t_{\text{OBP}} \leqslant L_{\text{best}}$, the onlooker bee i will produce a new food source around the global best food source x_{b} using Eq. (22). Otherwise, if $t_{\text{OBP}} > L_{\text{best}}$, Eq. (23) is utilized to generate the new food source by onlooker bee. Then, each onlooker bee m evaluates the new food source v_i using Eq. (21). If $\text{fit}(v_i) \geqslant \text{fit}(x_{\text{b}})$, replace the global best food source x_{b} with the new food source v_i, and reset the counter $t_{i_{\text{b}}}$ to 0, i.e., $t_{i_{\text{b}}} = 0$. Otherwise, if $\text{fit}(v_i) < \text{fit}(x_{\text{b}})$, keep the global best food source x_{b} unchanged, and add one to the counters $t_{i_{\text{b}}}$ and t_{OBP}, i.e., $t_{i_{\text{b}}} = t_{i_{\text{b}}} + 1$ and $t_{\text{OBP}} = t_{\text{OBP}} + 1$. If $\text{fit}(v_i) \geqslant \text{fit}(x_i)$, replace the food source x_i with the new food source v_i, and reset the counter t_i to 0, i.e., $t_i = 0$. Otherwise, if $\text{fit}(v_i) < \text{fit}(x_i)$, keep the global best food source x_{b} unchanged, and add one to the counter t_i, i.e., $t_i = t_i + 1$. Particularly, the onlooker bee phase is performed for each onlooker bee, only when the termination criteria are not reached.

4.1.4 Scout bee phase

In the scout bee phase, the global best food source achieved so far and its index are recorded. Then, for each food source x_i, if $t_i > L_{\text{init}}$, this food source is abandoned. And its employed bee turns to be a scout bee to generate a new food source using Eq. (20).

4.2 IQABC-based centralized deployment algorithm

In case where the centralized satellite controller work properly, the IQABC-based centralized deployment algorithm is proposed to improve the communication coverage area while enhancing the coverage uniformity.

4.2.1 Mapping IQABC into centralized coverage

The mapping relationship between IQABC and centralized coverage is illustrated in Fig. 3. Bees in IQABC find good food sources with high profitability in their hive, where the profitability is a value calculated

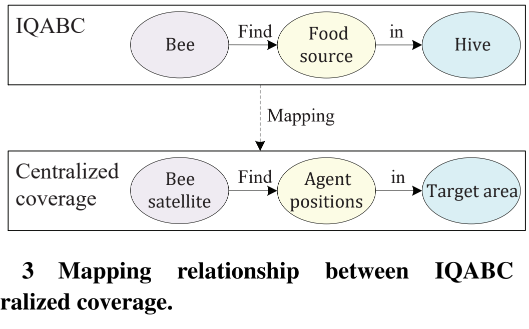


Fig. 3 Mapping relationship between IQABC and centralized coverage.

by utilizing the fitness function. On the other hand, artificial bees are created in satellite controller to find good positions for all agents with high coverage area in the target area.

Particularly, for IQABC and centralized coverage, four special instructions are given below. (1) Positive feedback. The food sources with higher profitability attract more onlooker bees to visit them. Similarly, the positions with better coverage performance engage more onlooker bees in satellite controller to exploit them. (2) Negative feedback. The exhausted food sources are abandoned by scout bees, while the perishing positions are yielded by the scout bees in satellite controller. (3) Fluctuations. The scout bees find a new food source randomly, while the scout bees in satellite controller find new positions for all agents stochastically. (4) Multiple interactions. A honey bee shares the food source it finds with other bees. Similarly, the bee in satellite controller shares the positions of agents with other bees in satellite controller.

4.2.2 Food source encoding

Each food source encodes a solution for the centralized communication coverage problem. The optimized food source is the final positions of all agents. The horizontal position of agent N_i $(i = 1, 2, \ldots, n)$ flying at the optimal altitude h_{opt} is denoted by $(p^{\text{a}}_{i,1}, p^{\text{a}}_{i,2})$. The food source encoding n mobile robotic agents is denoted by $x_i = (p^{\text{a}}_{1,1}, p^{\text{a}}_{1,2}, p^{\text{a}}_{2,1}, p^{\text{a}}_{2,2}, \ldots, p^{\text{a}}_{n,1}, p^{\text{a}}_{n,2})$, where the dimension of the food source is twice the number of agents, i.e., $D = 2n$. Furthermore, each agent is supposed to be deployed inside the target area, i.e., $0 \leqslant p^{\text{a}}_{i,1} \leqslant L$ and $0 \leqslant p^{\text{a}}_{i,2} \leqslant W$. In additon, based on the objective function Eq. (15), the fitness function is constructed to evaluate the quality of each solution using Eq. (21).

4.3 IQABC-based distributed deployment algorithm

For the situation that the centralized satellite controller suffers breakdown, we propose an IQABC-based distributed deployment algorithm utilizing Voronoi partition to deploy each agent distributedly utilizing only local information.

4.3.1 Mapping IQABC into distributed coverage

As illustrated in Fig. 4, the mapping relationship between IQABC and distributed coverage consists of three aspects: bees correspond to agent bees, food source corresponds to the candidate position of each agent, and hive corresponds to its local polygon. Particularly, the four special instructions mentioned above for IQABC and centralized coverage also apply to the distributed coverage where bees in satellite controller are replaced by agent bees.

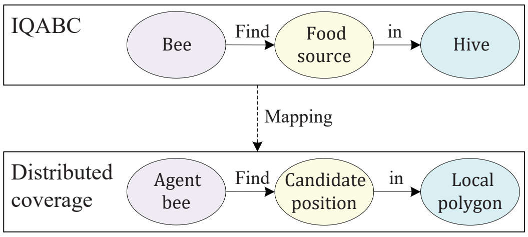


Fig. 4 Mapping relationship between IQABC and distributed coverage.

4.3.2 Framework of IQABC-D

The IQABC-D algorithm is performed to deploy agents iteratively. To better reflect the deployment process of the IQABC-D algorithm, the framework of IQABC-D for each agent is given in Fig. 5. In each iteration, each agent first updates its local information by broadcasting its local information and receiving one-hop messages. Then, each agent calculates its local polygon information including local polygon area, local coverage area, and local coverage hole through the Voronoi partition. Finally, according to the calculated local coverage hole, each agent moves with the proposed moving scheme, which is presented in Section 4.3.3.

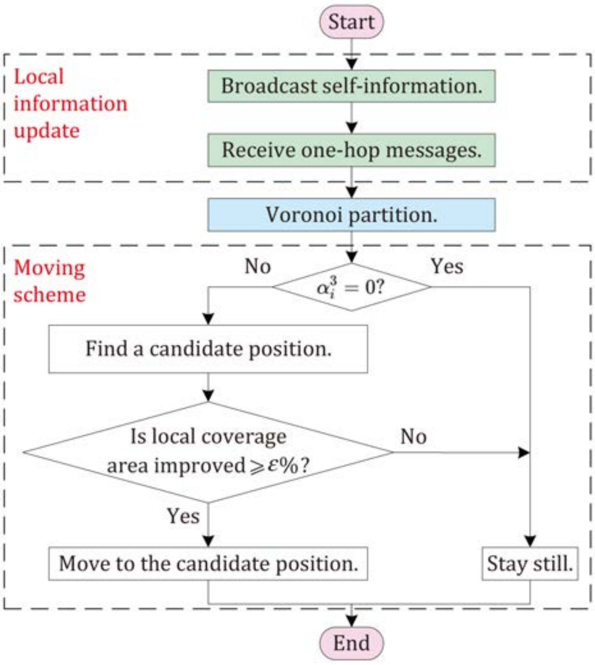

Start
Local information update
Broadcast self-information.
Receive one-hop messages.
Voronoi partition.
Moving scheme
No
Yes
$\alpha_i^3 = 0?$
Find a candidate position.
Is local coverage area improved ⩾ ε%?
No
Yes
Move to the candidate position.
Stay still.
End

Fig. 5 Framework of IQABC-D for each agent in each iteration.

4.3.3 Moving scheme

Each agent finds a candidate position by utilizing the IQABC algorithm. If the local coverage area of the candidate position is improved by $\varepsilon\%$ or more, the agent will move to this candidate position. Otherwise, the agent will keep still. Herein, each food source is a position inside its Voronoi polygon, i.e., $x_i = (p^{\mathrm{a}}_{i,1}, p^{\mathrm{a}}_{i,2})$. The objective of finding a candidate position is to minimize its local coverage hole, which is expressed in Eq. (25).

$$f(x_i) = \alpha_i^3 \tag{25}$$

Particularly, there are two adaptive adjustments for each food source: position adjustment and speed adjustment.

(1) Position adjustment. Since the candidate position is supposed to be inside the Voronoi polygon, the position generated in IQABC-D should be adjusted when it is outside the Voronoi polygon. The position adjustment can be expressed as follows:

$$x_i = \overline{x_i x_o} \cap \boldsymbol{E}_i,\ x_i \text{ is outside its Voronoi polygon} \tag{26}$$

where $\overline{x_i x_o}$ denotes the line from the old position x_o to the generated position x_i.

(2) Speed adjustment. Another constraint is that the speed of each agent should be no greater than the maximum speed. Let $\boldsymbol{v}_i$ denote the velocity moving to the generated position x_i, i.e., $\boldsymbol{v}_i = x_i - x_o$, the velocity adjustment can be expressed as follows:

$$\boldsymbol{v}_i' = \frac{\boldsymbol{v}_i}{|\boldsymbol{v}_i|} \tag{27}$$

5 Simulation Result

5.1 Simulation setting

The simulation experiments are conducted on a computer with Intel(R) Xeon(R) Intel(R) E5-2609 CPU processor and 64.00 G RAM. The IQABC-based hybrid deployment algorithm is conducted on MATLAB 2015b platform. To evaluate the performance of the IQABC-based centralized deployment algorithm (IQABC-C) and IQABC-based distributed deployment algorithm (IQABC-D), ABC[30] and QABC[31] are employed for comparison with the IQABC based algorithm. In the simulation, a given number of UAV agents hovering at optimal altitudes are deployed above a target area. Main parameters[7, 25] are shown in Table 2.

Performance metrics including objective function value, coverage rate, and coverage uniformity are adopted to evaluate the proposed deployment algorithm.

Table 2 Simulation parameters.

Algorithm/ scenario	Parameter	Symbol	Value
Scenario	Size of target area	$L \times W$	2000 m× 2000 m
	Network coverage range	R_{a2g}	250 m
	Communication range	R_{a2a}	700 m
	Number of UAV agents	n	30
IQABC-C	Number of initialized food sources	sn	20
	Maximum cycle number	$C^{\mathrm{a}}_{\max}$	1000
	Neighbourhood radius in QABC	r	1
IQABC-D	Maximum speed of agents	$v_{\max}$	50 m/cycle

The coverage rate C_{r} is defined as follows:

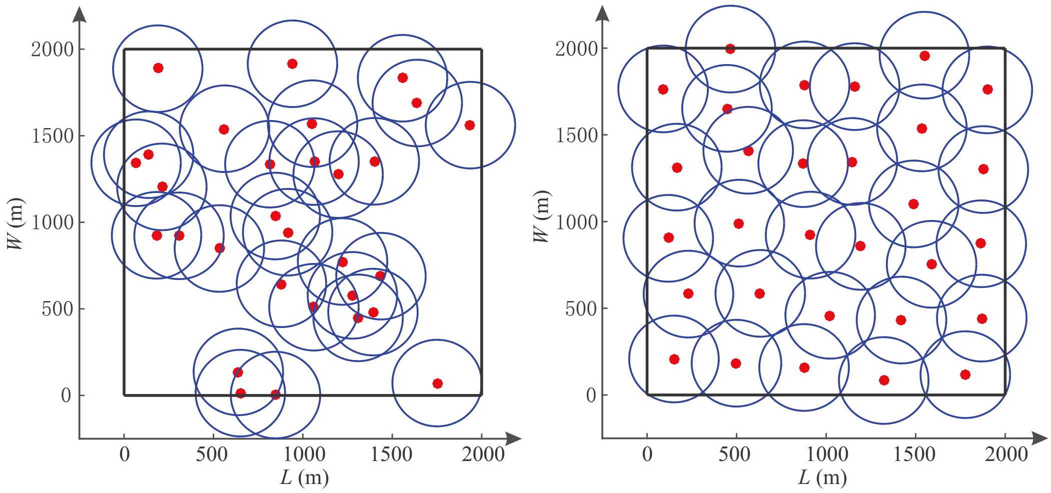
$$C_{\mathrm{r}} = \frac{\sum\limits_{i=1}^{n} \alpha_i^2}{\sum\limits_{i=1}^{n} \alpha_i^1} \tag{28}$$

5.2 Evaluation of IQABC-C

The IQABC-C is performed with parameters shown in Table 2. The initial random deployment is shown in Fig. 6a, it can be seen that there are many coverage holes and coverage overlaps that interfere with improving the whole ground coverage area. After performing IQABC-C, the final deployment is obtained and illustrated in Fig. 6b. It is easy to find that agents are deployed rather evenly in the target area and the total coverage areas are improved significantly.

Moreover, ABC and QABC are exploited as comparison with IQABC-C with the same problem mapping and food source encoding. As illustrated in Figs. 7a, IQABC-C performs better than ABC and QABC on both objective function and convergence speed. Furthermore, as illustrated in Figs. 7b and 7c, IQABC-C can obtain maximal coverage rate and minimal coverage uniformity quickly compared with the other two algorithms. Particularly, due to the nature of the objective function, coverage uniformity and coverage rate changes along with cycles are not monotonic. This proves the advantage of effectively exploiting the IQABC in converage deployment with centralized controller.

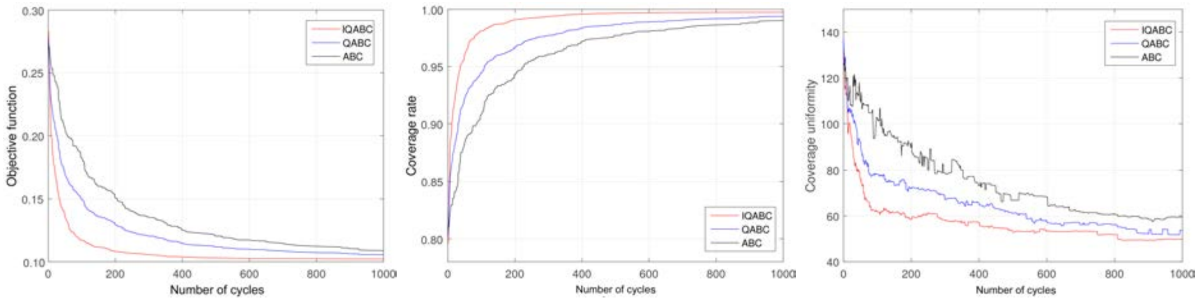
In addition, using the optimal solution obtained by ABC-based deployment algorithm as the reference value, we calculated the number of cycles required for the IQABC-C deployment algorithm and QABC-based deployment algorithm to converge to this optimal solution. The results are shown in Table 3. It can be



(a) Initial deployment

(b) Final deployment

Fig. 6 IQABC-C deployment instance with 30 agents.



(a) Objective function vs. number of cycles

(b) Coverage rate vs. number of cycles

(c) Coverage uniformity vs. number of cycles

Fig. 7 Comparison of IQABC-C and ABC variants.

Table 3 Ratio of number of cycles for different deployment algorithms.

Index	Result
	ABC: QABC: IQABC-C
Number of cycles	1: 0.70: 0.20

seen that the IQABC-C deployment algorithm has the fastest convergence speed, followed by the QABC-based deployment algorithm and the ABC-based deployment algorithm. Additionally, it can be observed from Fig. 7 that the IQABC-C deployment algorithm produces better or at least similar results compared to the QABC-based and ABC-based deployment algorithms within the same cycle.

Finally, the impact of agent number on final deployment by performing IQABC-C is evaluated in terms of objective function, coverage rate, and coverage uniformity, as illustrated in Fig. 8. It can be seen clearly that with the increase of agent number, objective function value decreases while coverage rate increases. Furthermore, coverage uniformity first decreases and then increases. Coverage uniformity reaches the minimal value with 25 agents, after which coverage rates are almost the maximal values. In other words, with the given environmental conditions, the optimal number of agents is 25 to get maximal coverage rate and minimal coverage uniformity.

5.3 Evaluation of IQABC-D

To evaluate the local deployment for a particular agent in IQABC-D, we compare IQABC-D with ABC and QABC. The agent is randomly initialized in the target area at $(845.77, 1036.10)$ with 5 Voronoi neighbors located at $(1197.05, 1275.42)$, $(1062.67, 1350.66)$, $(815.24, 1335.66)$, $(532.94, 848.69)$, and $(914.85, 942.71)$, respectively. Its Voronoi polygon is shown in Fig. 9a. These three algorithms are executed for 40 cycles and run 30 times to obtain the average local coverage rate. The average local coverage rate changes with cycles, as shown in Fig. 9b. It is evident that IQABC-D deployment algorithm converges faster than the QABC-based deployment algorithm and the

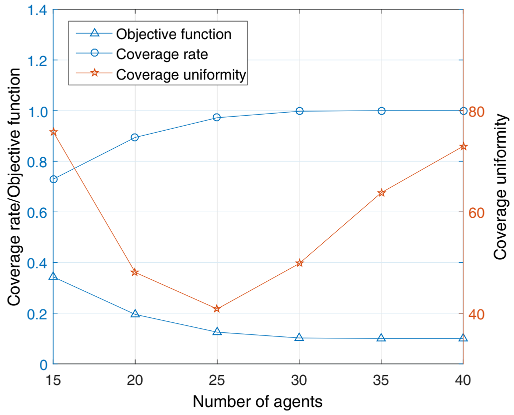
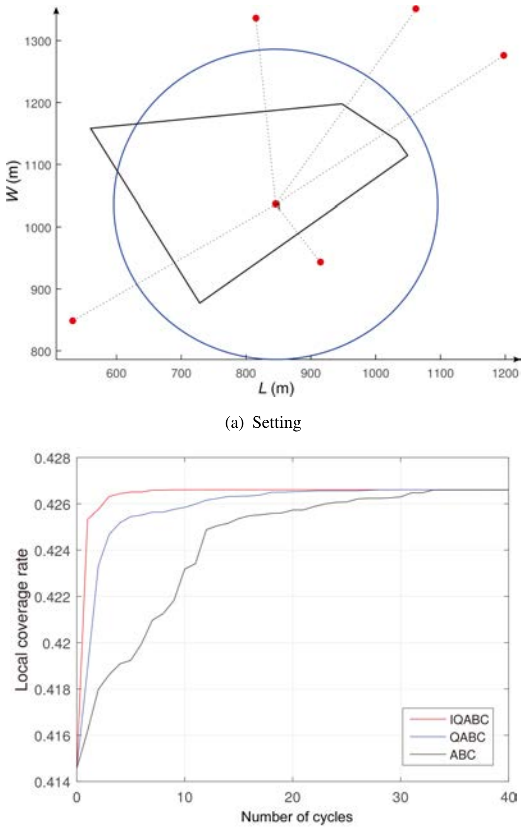


Fig. 8 Performance of IQABC-C vs. number of agents.



(a) Setting

(b) Average local coverage rate vs. number of cycles

Fig. 9 Comparison of IQABC-D with ABC variants for a particular agent.

ABC-based deployment algorithm. Moreover, IQABC-D deployment algorithm produces better or at least similar results compared to the QABC-based deployment algorithm and ABC-based deployment algorithm within the same cycle.

As mentioned in Section 4.3.2, with different settings of stop control parameter ε, the proposed moving scheme in IQABC-D performs differently. To set the appropriate stop control parameter, namely, coverage threshold ε, coverage rate, and number of stop cycles are evaluated with parameters shown in Table 2. As shown in Fig. 10, with the increase of coverage threshold, coverage rates and number of stop cycles decrease, because it is easier to meet the stop condition with larger coverage threshold, and coverage rate improves with the increase of cycles. In order to stop the algorithm with high coverage rate and low stop cycle, we set coverage threshold $\varepsilon = 0.1$.

In addition, the IQABC-D is performed with parameters shown in Table 2. The initial random deployment is shown in Fig. 11a, it can be seen that there are many coverage holes and coverage overlaps that interfere with improving the whole ground coverage area. After performing IQABC-D iteratively, as shown in Figs. 11b–11e, the whole coverage rate improves with the decrease of each agent's local coverage hole. Finally, we obtain the final deployment after 34 cycles, which is illustrated in Fig. 11f, where agents are deployed rather evenly in the target area and the total coverage areas are improved significantly. This demonstrates that the proposed IQABC-D algorithm can achieve a solid performance, even when the centralized control do not exist.

To evaluate the performance of IQABC-D, EVF, VVF, VEVF, and VEDGE[32, 33] are employed as comparison algorithms since they are both distributed algorithms for area coverage improvement. As shown in Figs. 12a and 12b, all of these distributed algorithms have better performance than the initial random deployment in terms of coverage rate and coverage uniformity. Moreover, IQABC-D has higher coverage rate and better coverage uniformity than other compared algorithms.

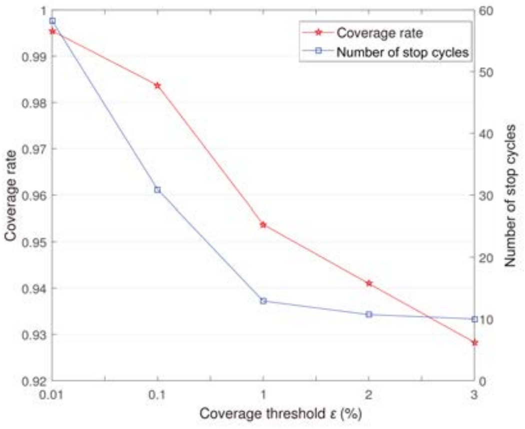
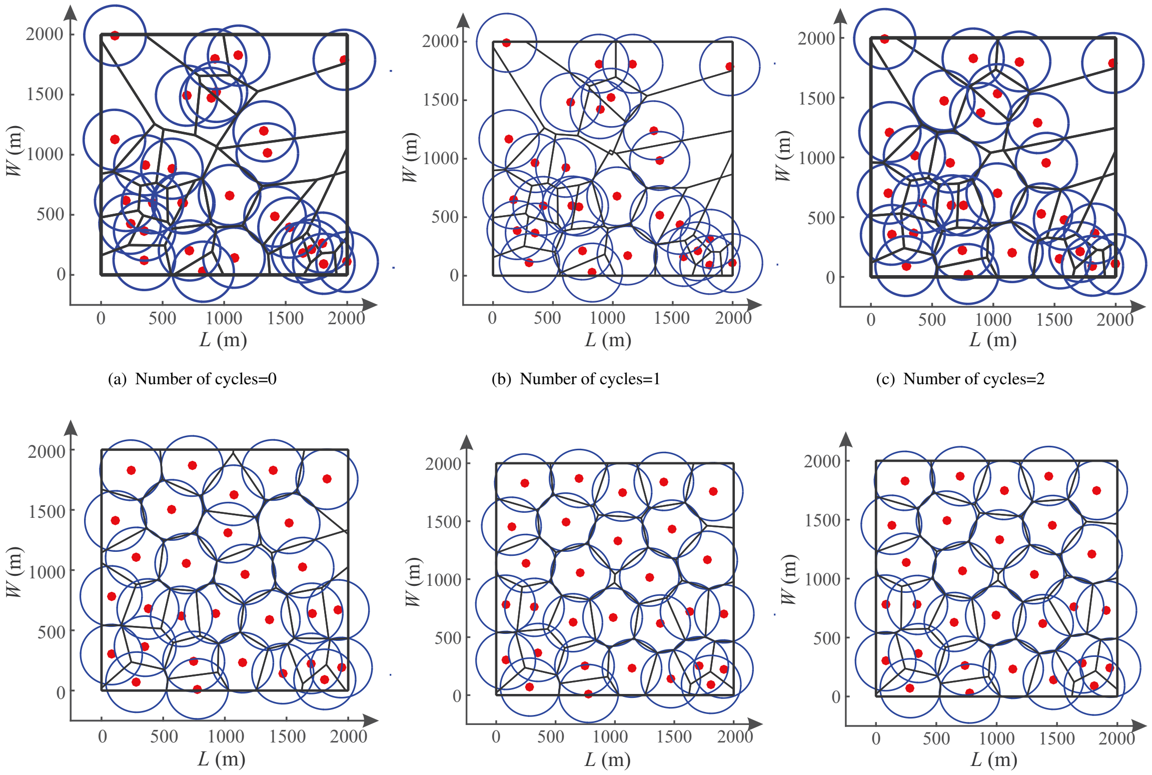


Fig. 10 Coverage rate and number of stop cycles vs. converage threshold ε.



(a) Number of cycles=0

(b) Number of cycles=1

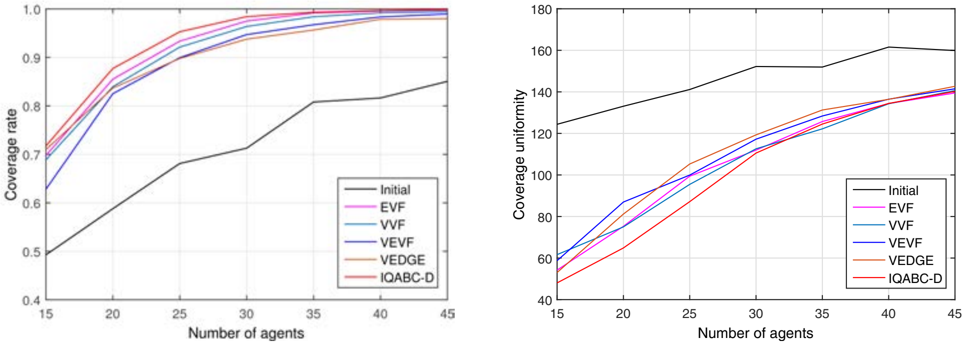
(c) Number of cycles=2

(d) Number of cycles=10

(e) Number of cycles=20

(f) Number of cycles=34

Fig. 11 IQABC-D deployment instance with 30 agents.



(a) Coverage rate vs. number of agents

(b) Coverage uniformity vs. number of agents

Fig. 12 Performance of IQABC-D vs. number of agents.

6 Conclusion

In this paper, we propose an IQABC-based hybrid deployment algorithms, including IQABC-C and IQABC-D, to deploy mobile robotic agents providing network coverage in a target region. In IQABC-C, an objective function is designed to maximize the coverage area while minimizing the coverage uniformity. In IQABC-D, only local information without any prior information are utilized. Simulations demonstrate the proposed algorithms can effectively deploy mobile robotic agents to provide network coverage. However, limited power constraint and environmental changes are not considered in this paper, which will be discussed in future.

Appendix

A Proof of that there is only one optimal altitude for a given agent to provide the maximum coverage range

To prove that there is only one optimal altitude h_{opt} for a given agent N_i in a particular environment to provide the maximum coverage range, we conduct experiments and analysis in various environments. Four different environments (suburban environment, urban environment, dense urban environment, and high-rise urban environment) are selected and the following parameters are set; the maximum allowable path loss $L_{\text{th}} = 90\,\text{dB}$, the carrier frequency of agent-to-user channel $f = 2\,\text{GHz}$, while using the following $(a, b, \eta_{\text{LoS}}, \eta_{\text{NLoS}})$ pairs $(4.88, 0.43, 0.1, 21)$, $(9.61, 0.21, 1, 20)$, $(12.08, 0.18, 1.6, 23)$, and $(27.23, 0.15, 2.3, 34)$ to describe suburban, urban, dense urban, and high-rise urban, respectively[26]. We can obtain the coverage range with respect to the altitude of agent according to Eq. (7), which is illustrated in Fig. A1.

As shown in Fig. A1, it can be observed that for an agent in a particular environment, its coverage range initially increases and then decreases with the increase of altitude. This indicates the existence of a single optimal altitude, denoted by h_{opt}, which provides the maximum coverage range. As illustrated in Fig. A1, the optimal altitudes for suburban, urban, dense urban, and high-rise urban environments are 127.75 m, 189.90 m, 196.96 m, and 199.52 m, while the corresponding maximum coverage ranges are 344.63 m, 251.82 m, 206.11 m, and 89.59 m, respectively. Therefore, for a given agent N_i at altitude h_{opt} in a particular environment, all the ground points located at a distance R_{a2g} experience the same path loss L_{th}, and all points located at a distance $r < R_{\text{a2g}}$ experience a path loss smaller than L_{th}.

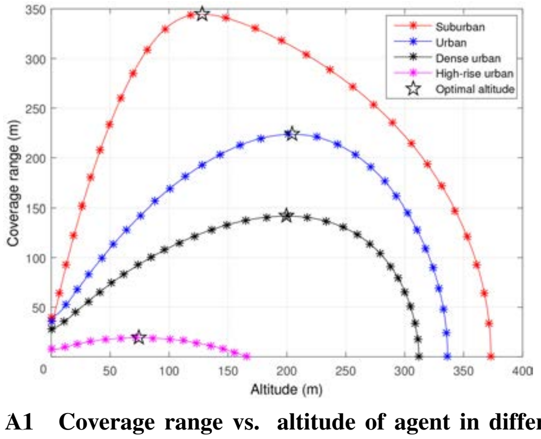


Fig. A1 Coverage range vs. altitude of agent in different environments.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 62102280), Fundamental Research Program of Shanxi Province (No. 20210302124167), Key Research and Development Program of Shanxi Province (No. 202102020101001), and National Major Scientific Research Instrument Development Project of China (No. 62027819).

References

[1] J. Li, A. M. V. V. Sai, X. Cheng, W. Cheng, Z. Tian, and Y. Li, Sampling-based approximate skyline query in sensor equipped IoT networks, *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 219–229, 2020.

[2] J. Li, M. Siddula, X. Cheng, W. Cheng, Z. Tian, and Y. Li, Approximate data aggregation in sensor equipped IoT networks, *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 44–55, 2019.

[3] R. Shahzadi, M. Ali, H. Z. Khan, and M. Naeem, UAV assisted 5G and beyond wireless networks: A survey, *J. Netw. Comput. Appl.*, vol. 189, p. 103114, 2021.

[4] 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, 2019.

[5] A. D. Boursianis, M. S. Papadopoulou, P. Diamantoulakis, A. Liopa-Tsakalidi, P. Barouchas, G. Salahas, G. Karagiannidis, S. Wan, and S. K. Goudos, Internet of things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: A comprehensive review, *Internet of Things*, vol. 18, p. 100187, 2022.

[6] S. Zhang and J. Liu, Analysis and optimization of multiple unmanned aerial vehicle-assisted communications in post-disaster areas, *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12049–12060, 2018.

[7] X. Liu, X. Wang, J. Jia, and M. Huang, A distributed deployment algorithm for communication coverage in wireless robotic networks, *J. Netw. Comput. Appl.*, vol. 180, p. 103019, 2021.

[8] S. Aslan, H. Badem, and D. Karaboga, Improved quick artificial bee colony (iqABC) algorithm for global optimization, *Soft Comput.*, vol. 23, no. 24, pp. 13161–13182, 2019.

[9] X. Liu, X. Wang, M. Huang, J. Jia, N. Bartolini, Q. Li, and D. Zhao, Deployment of UAV-BSs for on-demand full communication coverage, *Ad Hoc Netw.*, vol. 140, p. 103047, 2023.

[10] 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 Wirel. Commun. Lett.*, vol. 6, no. 4, pp. 434–437, 2017.

[11] L. Yin, N. Zhang, and C. Tang, On-demand UAV base station deployment for wireless service of crowded tourism areas, *Pers. Ubiquitous Comput.*, vol. 26, no. 4, pp. 1137–1149, 2022.

[12] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, Placement optimization of UAV-mounted mobile base stations, *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 604–607, 2017.

[13] H. Zhao, H. Wang, W. Wu, and J. Wei, Deployment algorithms for UAV airborne networks toward on-demand coverage, *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2015–2031, 2018.

[14] A. Trotta, M. D. Felice, F. Montori, K. R. Chowdhury, and L. Bononi, Joint coverage, connectivity, and charging strategies for distributed UAV networks, *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 883–900, 2018.

[15] L. Ruan, J. Wang, J. Chen, Y. Xu, Y. Yang, H. Jiang, Y. Zhang, and Y. Xu, Energy-efficient multi-UAV coverage deployment in UAV networks: A game-theoretic framework, *China Commun.*, vol. 15, no. 10, pp. 194–209, 2018.

[16] X. Zhang and L. Duan, Optimization of emergency UAV deployment for providing wireless coverage, in *Proc. GLOBECOM 2017 - 2017 IEEE Global Communications Conf.*, Singapore, 2018, pp. 1–6.

[17] X. Zhang and L. Duan, Fast deployment of UAV networks for optimal wireless coverage, *IEEE Trans. Mob. Comput.*, vol. 18, no. 3, pp. 588–601, 2019.

[18] N. T. Hanh, H. T. T. Binh, N. X. Hoai, and M. S. Palaniswami, An efficient genetic algorithm for maximizing area coverage in wireless sensor networks, *Inf. Sci.*, vol. 488, pp. 58–75, 2019.

[19] D. G. Reina, H. Tawfik, and S. L. Toral, Multi-subpopulation evolutionary algorithms for coverage deployment of UAV-networks, *Ad Hoc Netw.*, vol. 68, pp. 16–32, 2018.

[20] B. Cao, J. Zhao, Z. Lv, X. Liu, X. Kang, and S. Yang, Deployment optimization for 3D industrial wireless sensor networks based on particle swarm optimizers with distributed parallelism, *J. Netw. Comput. Appl.*, vol. 103, pp. 225–238, 2018.

[21] J. Wang, C. Ju, Y. Gao, A. K. Sangaiah, and G. Kim, A PSO based energy efficient coverage control algorithm for wireless sensor networks, *Comput., Mater. Contin.*, no. 9, pp. 433–446, 2018.

[22] W. Du, W. Ying, P. Yang, X. Cao, G. Yan, K. Tang, and D. Wu, Network-based heterogeneous particle swarm optimization and its application in UAV communication coverage, *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 4, no. 3, pp. 312–323, 2020.

[23] T. Qasim, M. Zia, Q. A. Minhas, N. Bhatti, K. Saleem, T. Qasim, and H. Mahmood, An ant colony optimization based approach for minimum cost coverage on 3-D grid in wireless sensor networks, *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1140–1143, 2018.

[24] J. Li, D. Lu, G. Zhang, J. Tian, and Y. Pang, Post-disaster unmanned aerial vehicle base station deployment method based on artificial bee colony algorithm, *IEEE Access*, vol. 7, pp. 168327–168336, 2019.

[25] X. Liu, X. Wang, J. Jia, J. Lv, and N. Bartolini, Distributed deployment in UAV-assisted networks for a long-lasting communication coverage, *IEEE Syst. J.*, vol. 16, no. 3, pp. 4130–4138, 2022.

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

[27] F. Aurenhammer and R. Klein, Voronoi diagrams, in *Handbook of Computational Geometry*, J. R. Sack and J. Urrutia, eds. Amsterdam, the Netherlands: North-Holland Publishing Co., 2000, pp. 201–290.

[28] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs, *IEEE Trans. Wirel. Commun.*, vol. 15, no. 6, pp. 3949–3963, 2016.

[29] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, A comprehensive survey: Artificial bee colony (ABC) algorithm and applications, *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, 2014.

[30] D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *J. Glob. Optim.*, vol. 39, no. 3, pp. 459–471, 2007.

[31] D. Karaboga and B. Gorkemli, A quick artificial bee colony (qABC) algorithm and its performance on optimization problems, *Appl. Soft Comput.*, vol. 23, pp. 227–238, 2014.

[32] H. Mahboubi and A. G. Aghdam, Distributed deployment algorithms for coverage improvement in a network of wireless mobile sensors: Relocation by virtual force, *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 4, pp. 736–748, 2017.

[33] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors, *IEEE Trans. Ind. Inform.*, vol. 10, no. 1, pp. 163–174, 2014.

Shuang Xu received the PhD degree in computer application technology from Northeastern University, Shenyang, China in 2019. She is currently a master's supervisor at College of Computer Science and Technology (College of Data Science), Taiyuan University of Technology, Shanxi, China. Her research interests include space-air-ground integrated networks, mobile edge computing, and big data.

Xiaojie Liu received the PhD degree in computer science and engineering from Northeastern University, China in 2022. She was a visiting PhD student in Sapienza University of Rome, Italy in 2021. She is currently working as a postdoctoral researcher in Pengcheng Laboratory, China. Her research interests include UAV-assisted communication networks, space-air-ground integrated networks, etc.

Dengao Li received the PhD degree from Taiyuan University of Technology, China in 2010. He is currently a professor at College of Computer Science and Technology (College of Data Science), Taiyuan University of Technology, China. His main research interests include the signal processing, wireless networks, mobile computing, cyber-physical system, and algorithms.

Jumin Zhao received the PhD degree from Taiyuan University of Technology, China in 2008. She is currently a professor at College of Information and Computer, Taiyuan University of Technology, China. Her main research interests include signal and information processing and wireless sensor network.