A Deployment Method Based on Artificial Bee Colony Algorithm for UAV-Mounted Base Stations

Chen Zhang, Leyi Zhang, Yanming Liu, Lipeng Zhu, and Zhenyu Xiao *

School of Electronic and Information Engineering, Beihang University, Beijing, 100191, China.
{moon_zc, 16231001, liuyanming, zhulipeng, xiaozy}@buaa.edu.cn

Abstract. With high mobility and low cost, unmanned aerial vehicles (UAVs) are widely used in wireless communication systems. Especially in emergencies, UAVs can be used as aerial base stations (BSs) to provide wireless communication services for ground users. Aiming to reduce cost, we prefer to minimizing the number of UAVs needed to serve all users. Compared with the existing works, we take the constraints of required quality of service (QoS) and the service ability of each UAV into consideration. To solve the formulated mixed-integer programming problem, we propose a three-step method. First, to ensure each UAV can serve more users, the maximum service radius of UAVs is derived according to users' QoS requirement. Second, we propose an artificial bee colony (ABC) algorithm based clustering method to cluster users into different groups in the horizontal direction. Third, we adjust the positions of UAVs to obtained a better communication performance of the wireless communication system. Finally, the simulation results are presented to demonstrate the superiority of the proposed method.

Keywords: Wireless communication · Unmanned aerial vehicles · Aerial base stations · Three-dimensional deployment.

1 Introduction

Because of the high mobility, high agility, and high stability of line-of-sight (LoS) channel [1] of unmanned aerial vehicles (UAVs), wireless communication assisted by UAVs has become more and more popular in recent years [2, 3]. Under some circumstances, terrestrial infrastructures are unable to maintain a wireless communication system. For example, earthquakes and floods may destroy those facilities, while concerts and competitions lead to increased traffic exceeding the service ability of the system. It is convenient to apply UAVs to resume wireless communication. A solution to use UAVs in emergencies is to apply UAVs as aerial base stations (BSs).

* This work was supported in part by the National Key Research and Development Program (Grant Nos. 2016YFB1200100), and the National Natural Science Foundation of China (NSFC) (Grant Nos. 61827901 and 91738301).

Applying UAVs as aerial-BSs has attracted lots of attention from academia, including several topics. Among all these topics, the deployment of the UAV is a basic one. In [4], the authors maximized the coverage region by optimizing the height of a single UAV. In [5], a three-dimensional (3D) deployment algorithm was proposed, which can maximize the coverage region of each UAV while satisfying the quality of service (QoS) of different users. In [6], a UAV was used as an aerial-BS to serve ground users as many as possible while consuming power as little as possible.

However, as users' number and demands for communication are increasing rapidly, using a single UAV to satisfy all users is becoming more and more difficult. Thus, many studies have discussed the problem of deploying multiple UAVs. In [7], a spiral deployment algorithm was proposed to deploy UAVs in two-dimensional (2D). An algorithm was designed to minimize the number of UAVs needed to serve all the ground users. Similarly, the authors in [8] utilized the elephant herding optimization algorithm [9] to minimize the number of UAVs. In [10], the number of UAVs was minimized in the condition of known and unknown user location. In [11], a low time complexity algorithm was proposed to minimize the number of UAVs and to optimize the 3D positions of UAVs to improve resource utilization.

In this paper, we study a downlink UAV network where multiple UAVs are deployed as BSs to serve the ground users with constraint on service ability. Different from the existing deployment studies, we are committed to using as few as possible UAVs to serve all the ground users and optimize the QoS of users at the same time. As mentioned before, the application of a single UAV is limited because of its finite service region and service ability. Besides, the air to ground (A2G) channel model is decided by the positions of UAVs and users, so the QoS can be improved by adjusting the 3D positions of UAVs.

The rest of this paper is organized as follows. In section 2, the system model is presented and the multiple UAVs deployment problem is formulated. In section 3, the solution to the problem is introduced. In section 4, the simulation performance compared with existing methods is provided to show the superiority of our method. Finally, we conclude this paper in section 5.

2 System Model and Problem Formulation

A downlink wireless communication network assisted by multiple UAVs is shown in Fig.1, where UAVs are used to transmit data to users randomly distributed in a 2D area $\mathcal{D} = [0, x_{max}] \times [0, y_{max}]$. Users are denoted by $\mathcal{K} = \{1, 2, \cdots, K\}$ and the position of each user is presented by $\boldsymbol{w_k} = [x_k, y_k]^{\mathrm{T}} \in \mathbb{R}^{2\times1}$. At the same time, the deployment area of the UAVs is limited to a 3D area $\mathcal{P} = \{[x_m, y_m, h_m] | x_{min} \leq x_m \leq x_{max}, y_{min} \leq y_m \leq y_{max}, h_m > 0\}$. $\mathcal{M} = \{1, 2, \cdots, M\}$ denotes the set of UAVs, and $\boldsymbol{p_m} = [x_m, y_m, h_m]^{\mathrm{T}} \in \mathbb{R}^{3\times1}$ represents the position of each UAV $m \in \mathcal{M}$.

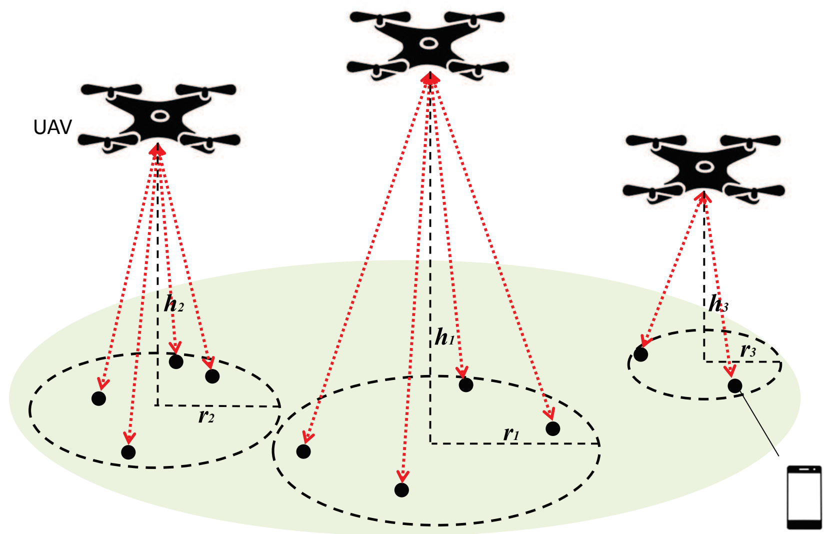


Fig. 1. The considered wireless communication system with UAVs

2.1 System Model

Since some obstacles like trees and buildings may block the link between UAVs and ground users in wireless communication, the channel between UAVs and users is usually a mixture of line-of-sight (LoS) link and none-line-of-sight (NLoS) link. Taking UAV m and user k for example, the large-scale channel gain $\beta_{m,k}$ between them in LoS environments and NLoS environments can be expressed as [12]:

$$\beta_{m,k}(d_{m,k}) = \begin{cases} \beta_0 \, {d_{m,k}}^{-\alpha} & \text{LoS environment,} \\ \kappa \, \beta_0 \, {d_{m,k}}^{-\alpha} & \text{NLoS environment,} \end{cases} \tag{1}$$

In (1), β_0 is the path loss of the reference distance in LoS environments. $\kappa \in (0, 1)$ is an attenuation coefficient for NLoS environments. $d_{m,k}$ represents the distance between user k and UAV m, which can be expressed as follow:

$$d_{m,k} = \sqrt{{h_m}^2 + {s_{m,k}}^2} = \frac{s_{m,k}}{cos\theta_{m,k}}, \tag{2}$$

where h_m is the height of UAV m, $s_{m,k} = \sqrt{(x_k - x_m)^2 + (y_k - y_m)^2}$ is the 2D distance between user k and UAV m, and $\theta_{m,k}$ is the evaluation angle between user k and UAV m.

Then, the probability of existing an LoS link between user k and UAV m can be given by [12]:

$$P_{LoS}(\theta_{m,k}) = \frac{1}{1 + a\ \exp(-b(\theta_{m,k} - a))}, \tag{3}$$

where a and b are parameters directly related to the environment. Then, the probability of NLoS links can be obtained as $P_{NLoS}(\theta_{m,k}) = 1 - P_{LoS}(\theta_{m,k})$.

Thus, we can obtain the channel gain between user k and UAV m:

$$\begin{aligned} \bar{g}_{m,k}(d_{m,k}, \theta_{m,k}) &\triangleq \mathbb{E}[|g_{m,k}|^2] \\ &= P_{LoS}(\theta_{m,k})\beta_0 {d_{m,k}}^{-\alpha} + P_{NLoS}(\theta_{m,k})\kappa\beta_0 {d_{m,k}}^{-\alpha} \\ &= \hat{P}_{LoS}(\theta_{m,k})\beta_0 {d_{m,k}}^{-\alpha}, \end{aligned} \tag{4}$$

where $\hat{P}_{LoS}(\theta_{m,k}) = P_{LoS}(\theta_{m,k}) + \kappa P_{NLoS}(\theta_{m,k})$.

In this paper, the received power of each user is used as the measurement of the QoS. The users' minimum required received signal power is denoted by P_0 and the received power of user k from UAV m can be given by:

$$P_{m,k} = \bar{g}_{m,k} \times P_t, \tag{5}$$

where P_t is UAVs' transmitting power. UAV m can successfully transmit data to user k only when $P_{m,k}$ is larger than threshold P_0. We can get the constraint about $\bar{g}_{m,k}$ according to (5), which can be given by:

$$\bar{g}_{m,k} \geq \bar{g}_0, \tag{6}$$

where $\bar{g}_0 = \frac{P_0}{P_t}$ represents the minimum channel gain for successful transmission. Only when the channel gain between UAV m and user k satisfies (6), can UAV m possibly serve user k.

2.2 Maximum Service Radius

According to the relationship $\cos\theta_{m,k} = \frac{s_{m,k}}{\sqrt{s_{m,k}^2 + h_m^2}}$, the channel gain $\bar{g}_{m,k}$ can be rewritten as a function of the 2D distance $s_{m,k}$ and UAV's height h_m. The relation is shown in Fig.2.

Fig.2 shows a typical plot of $\bar{g}_{m,k}$ versus h_m for different $s_{m,k}$ values. It can be seen that when h_m is fixed, $\bar{g}$ decreases with increasing $s_{m,k}$ and finally fails to meet the requirement (6) if $s_{m,k} > r_{max}$, where r_{max} represents the largest service radius of UAV. When $s_{m,k}$ is fixed, $\bar{g}$ increases to the highest point because of the increasing the probability of existing LoS link, and then decreases because of the attenuation of long distance. Therefore, given the minimum channel gain requirement $\bar{g}_0$, the optimal height can be calculated by [4]:

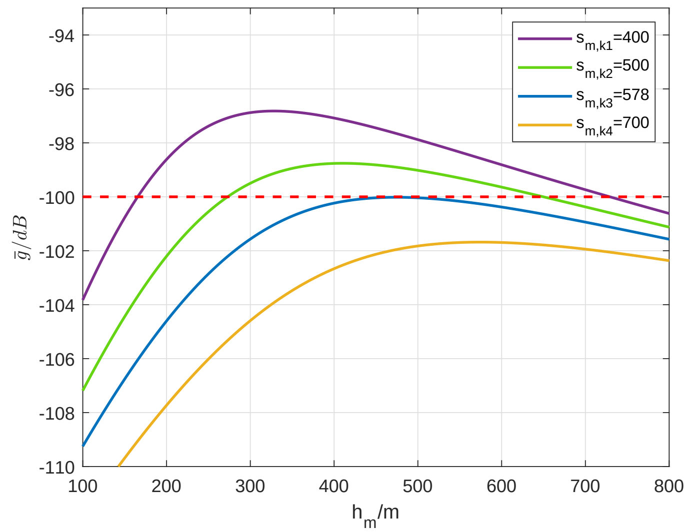


Fig. 2. Curve of channel gain $\bar{g}_0$ as a function of h_m when $s_{m,k}$ is fixed.

$$\frac{\partial s_{m,k}}{\partial h_m} = 0 \tag{7}$$

After deriving the optimal height h_m, the largest service radius r_{max} is obtained by solving the following equation [4]:

$$\bar{g}(r_{max}, h_m) = \bar{g}_0 \tag{8}$$

That is to say, when the 2D distance between UAV m and user k satisfies $s_{m,k} \leq r_{max}$, users k can be served by UAV m. However, if the distance between UAV m' and user k also satisfies the relation above, whether UAV m should serve user k becomes unclear. In order to solve this problem, we define an indicator function to ensure each user is served by only one UAV:

$$\gamma_{m,k} = \begin{cases} 1, & \text{User } k \text{ is served by UAV } m, \\ 0, & \text{Otherwise,} \end{cases} \tag{9}$$

where user k is served by UAV m for $\gamma = 1$, and $\gamma_{m,k} = 0$ otherwise.

2.3 Problem Formulation

The number of ground users served by each UAV is limited because of UAVs' limited service ability. N_{max} is used to represent the maximum number of ground users each UAV can serve. By jointly optimizing the connection between UAVs and ground users and the positions of UAVs, the problem for minimization of the number of UAVs is formulated as follows:

$$\begin{aligned}
\min_{\{\boldsymbol{p}_m\},\{\gamma_k^m\}} \quad & |\mathcal{M}| && (10)\\
\text{s.t.} \quad & \bar{g}_{m,k} \geq \bar{g}_0 \gamma_k^m \quad \forall k \in \mathcal{K}, \forall m \in \mathcal{M}. && (10a)\\
& \boldsymbol{p}_m \in \mathcal{P} \quad \forall m \in \mathcal{M} && (10b)\\
& \sum_{k \in \mathcal{K}} \gamma_{m,k} \leq N_{max} \quad \forall m \in \mathcal{M} && (10c)\\
& \sum_{m \in \mathcal{M}} \gamma_{m,k} = 1 \quad \forall k \in \mathcal{K} && (10d)
\end{aligned}$$

Constraint (10a) indicates that (6) must be satisfied when user k is served by UAV m. The constraints on the deployment area and service ability of each UAV are shown in (10b) and (10c). Constraint (10d) means that each user can only be served by one UAV.

There are integer and continuous variables in (10), meaning that it is a mixed-integer programming problem, which is difficult to solve [13]. In the next section, a suboptimal solution of (10) will be developed.

3 UAV Deployment Method

In this section, we first design an algorithm that combines the heuristic algorithm in [7] and the artificial bee colony (ABC) algorithm to cluster users into groups. This algorithm can minimize the number of UAVs required to serve all users. Then, we optimize the 3D position of each UAV to improve QoS.

3.1 User Clustering

We reformulate the problem of user clustering as follow:

$$\begin{aligned}
\min_{\{\gamma_{m,k}\}} \quad & |\mathcal{M}| && (11)\\
\text{s.t.} \quad & \gamma_{m,k} r_{m,k} \leq r_{max} \quad \forall k \in \mathcal{K}, \forall m \in \mathcal{M} && (11a)\\
& . \sum_{k \in \mathcal{K}} \gamma_{m,k} \leq N_{max} \quad \forall m \in \mathcal{M} && (11b)\\
& \sum_{m \in \mathcal{M}} \gamma_{m,k} = 1 \quad \forall k \in \mathcal{K} && (11c)
\end{aligned}$$

We aim at adjusting the serve indicator variable $\gamma_{m,k}$ to minimize the number of groups, which is also the number of UAVs. To solve (11), an Ordered ABC-based Placement (OAP) algorithm is designed, which combines ABC algorithm and heuristic algorithm. ABC algorithm can effectively find the optimal or suboptimal solution for difficult problem and heuristic algorithm is used to reduce the solution space of ABC algorithm to find the solution more quickly.

The main idea of the iterative algorithm OAP algorithm is to give priority to users located at the outmost periphery of all users. A circle with radius r_{max} is used to cover users, and users covered in each iteration will be clustered into the same group.

In each iteration, we first find users located at the outmost edge of the uncovered users, called boundary users $\mathcal{K}_{U,bo}$, while the other users are called inner users $\mathcal{K}_{U,in}$. In order to ensure that the clustering is performed in an order from outside to inside, a boundary user needs to be selected as the feature user k_0 of each group, so that each clustering work is carried out near the boundary of the uncovered user area. In the first iteration, a boundary user is randomly selected as k_0 from $\mathcal{K}_{U,bo}$, and in each subsequent iteration, the user on the boundary of updated uncovered users that is closest to k_0 in the last iteration will be selected as the new k_0.

After k_0 is selected, calculate the distances between all users and k_0. Boundary users and inner users with a distance of not greater than $2r_{max}$ from k_0 are grouped into the sets $\mathcal{K}_{local,bo}$ and $\mathcal{K}_{local,in}$. Only users in $\mathcal{K}_{local,bo}$ and $\mathcal{K}_{local,in}$ need to be considered when clustering users because users with distances greater than $2r_{max}$ are impossible to be clustered into the same groups. The process above is effective in reducing the solution space of ABC algorithm.

Then, ABC algorithm is applied to determine the cluster's center, trying to cluster as more users as possible into this group. The above iteration continues until all users are grouped. Finally, we derive the center position of each cluster and divide users into different groups. The set of groups is presented as $\mathcal{L} = \{\mathcal{L}^1, \cdots, \mathcal{L}^{|\mathcal{M}|}\}$. Every user k is guaranteed to belong to one subset of $\mathcal{L}$.

The process of clustering users into different groups is described in Algorithm 1. Step 6 depicts the process of applying ABC algorithm to decide the center of cluster, which is detailed in Algorithm 2.

ABC algorithm was designed to solve multivariable function optimization problems in 2005 [14]. The algorithm imitates the behavior of employed bees, onlooker bees and scout bees when they are searching for food to find the solution to the problem. The algorithm is presented as follows.

1. *Initialization*: Firstly, the initial solution set is randomly generated as $\mathcal{F}^{(0)} = \{F_1^{(0)}, \cdots, F_{N_p}^{(0)}\}$, where N_p stands for the total amount of solutions. Every initial solution $F_i^{(0)} = (x_i^{(0)}, y_i^{(0)}) \in \mathcal{F}^{(0)}$ is a possible location for the center of cluster m.

   Then we define a function $s(F_i^{(0)}, k_0) = \sqrt{(x_i^{(0)} - x_{k_0})^2 + (y_i^{(0)} - y_{k_0})^2}$ to represent the 2D distance $s(F_i^{(0)}, k_0)$ between each $F_i^{(0)}$ and k_0. If $s(F_i^{(0)}, k_0) > r_{max}$, which means the cluster cannot cover k_0, the distance $s(F_i^{(0)}, k_0)$ should be normalized to r_{max}. The new position $F_i^{\prime(0)}$ after normalization is determined as follow:

Algorithm 1: Ordered ABC-based User Clustering Algorithm

Input:
User set $\mathcal{K}$, user locations $\{w_k\}$
Output:
The number of groups $|\mathcal{M}|$ and set $\mathcal{L}$
1: Initialize $m = 1$, $\mathcal{L} = \emptyset$, $\mathcal{K}_U = \mathcal{K}$
2: **while** $\mathcal{K}_U \neq \emptyset$ **do**
3: Find boundary user set $\mathcal{K}_{U,bo} \subseteq \mathcal{K}_U$ and update inner user set $\mathcal{K}_{U,in} \leftarrow \mathcal{K}_U \backslash \mathcal{K}_{U,bo}$.
4: Choose the feature user k_0
5: For every $k_{U,bo} \in \mathcal{K}_{U,bo}$, calculate the distance between $k_{U,bo}$ and k_0. If the distance is not greater than $2r_{max}$, add $k_{U,bo}$ to $\mathcal{K}_{local,bo}$.
For every $k_{U,in} \in \mathcal{K}_{U,in}$, calculate the distance between $k_{U,in}$ and k_0. If the distance is not greater than $2r_{max}$, add $k_{U,in}$ to $\mathcal{K}_{local,in}$.
6: Use Algorithm 2 to obtain $\mathcal{L}^m$.
7: Set $\mathcal{K}_U \leftarrow \mathcal{K}_U \backslash \mathcal{L}^m$.
8: Update $m = m + 1$.
9: Add $\mathcal{L}^m$ to $\mathcal{L}$.
10: **end while**
11: $|\mathcal{M}| = m$ **return** $\mathcal{L}$.

$$\begin{cases} x_i'^{(0)} = \frac{r_{max}}{s(F_i^{(0)},k_0)}(x_i^{(0)} - x_{k_0}) + x_{k_0}, \\ y_i'^{(0)} = \frac{r_{max}}{s(F_i^{(0)},k_0)}(y_i^{(0)} - y_{k_0}) + y_{k_0}. \end{cases} \tag{12}$$

After that, we calculate the fitness value of every solution to find the best position to be the cluster's center. The fitness value is defined as follow:

$$f_{0_i}(x_0, y_0) = \begin{cases} \alpha_1 N_{bo} + \alpha_2 N_{in}, & N_{bo} + N_{in} \leq N_{max}, \\ 0.01, & N_{bo} + N_{in} > N_{max}, \end{cases} \tag{13}$$

where N_{bo} represents the number of boundary users in $\mathcal{K}_{local,bo}$ which are covered by the circle with $F_i^{(0)}$ or $F_i'^{(0)}$ as the center and r_{max} as the radius, N_{in} represents the number of inner users $\mathcal{K}_{local,in}$ covered by that circle. α_1 and α_2 are weights of N_{bo} and N_{in} respectively and satisfy $\alpha_1 > \alpha_2$. (13) means that if the number of users covered by the circle is no more than N_{max}, the number of covered users increases may lead to the increasing of fitness value and the bigger the fitness value is, the higher probability of $F_i^{(0)}$ to be the optimal solution. However, if the number of users covered by the circle is greater than N_{max}, we have $f_{0_i}(x_0, y_0) = 0.01$, which means that $F_i^{(0)}$'s fitness value is too small to be the optimal solution. By calculating the fitness value of each solution, the initial optimal solution F_c^m with the largest value of fitness function f_m^c can be obtained.

2. *Employed Bees Phase*: The role of Employed bees is to find other possible positions as group centers near the current locations. For every possible po-

Algorithm 2: ABC Procedure

Input: $\{w_k\}_{k\in\mathcal{K}} \in \mathbb{R}^{2\times 1}$, $\mathcal{K}_{local,bo}$, $\mathcal{K}_{local,in}$, k_0, N_{max}, r_{max}, N_p, T, T_s
Output: $\mathcal{L}^m$

1: Randomly initialize the set of possible positions for cluster's center $\mathcal{F}^{(0)}$.
For $F_{0_i} \in \mathcal{F}^{(0)}$, calculate its distance with feature user k_0. If the distance is greater than r_{max} then normalize it.
Calculate the fitness value of every position, and find the position F_c^m with the greatest fitness valuef_c^m
$t = 0$.
2: **while** $t \neq T$ **do**
3: Employed bees search for a better solution in the neighbourhood of current solution.
Update the solution set $\mathcal{F}^{(t)}$.
4: Onlooker bees search for a better solution according to the probability.
Update the solution set $\mathcal{F}^{(t)}$.
5: Scout bees generate a new solution if the current remains unmodified during T_s iterations.
6: Calculate fitness value of every solution in $\mathcal{F}^{(t)}$.
Find the greatest fitness value of solutions in $\mathcal{F}^{(t)}$ and compare it with f_c^m.
Choose the one with greater fitness value as F_c^m.
7: Update $t = t + 1$.
8: **end while**
9: Calculate the distance s_{k_i,F_c^m} between F_c^m and every user k_i. If $s_{k_i,F_c^m} < r_{max}$, add k_i to $\mathcal{L}^m$.
return $\mathcal{L}^m$.

sition $F_i^{(t-1)}$, an employed bee searches for a new position $F_i^{(t)} = (x_i^{(t)}, y_i^{(t)})$ as follow:

$$F_i^{(t)}(k) = F_i^{(t-1)}(k) + \phi(F_i^{(t-1)}(k) - F_j^{(t-1)}(k)), \tag{14}$$

where $k = 1, 2$ represents the x or y coordinate of the position, $F_j^{(t-1)}(k)$ represents another position in $\mathcal{F}^{(t-1)}$ differing from $F_i^{(t-1)}(k)$, $\phi \in [-1, 1]$ is a random number. After examining and adjusting the position $F_i^{(t)}$, we can compare its fitness value with $F_i^{(t-1)}$. If $F_i^{(t)}$ has a greater fitness value, $F_i^{(t-1)}(k)$ will be replaced by $F_i^{(t)}(k)$. Finally a new solution set $\mathcal{F}^{(t)}$ will be obtained.

3. *Onlooker Bees Phase*: Every onlooker bee selects a position in $\mathcal{F}^{(t)}$ according to the probability of every solution and starts searching for a better solution in its neighborhood. For every position $F_i^{(t)}$, the probability of being chosen by the onlooker bee is calculated as:

$$P_i = \frac{0.9 * f_i^{(t)}}{max(f_i^{(t)})} + 0.1, \tag{15}$$

where $max(f_i^{(t)})$ represents the largest fitness value of positions in $\mathcal{F}^{(t)}$. Every onlooker bee generates a random number $rand \in (0, 1)$. If $rand < P_i$, then the onlooker bee chooses $F_i^{(t)}$, and searches for a new position like (14). Then we can adjust its position and calculate its fitness value. If it has a greater fitness value, the current position will be replaced by the new one. Finally, we will get a new set $\mathcal{F}^{(t)}$. The largest fitness value in $\mathcal{F}^{(t)}$ is compared with f_c^m. If it is greater than f_c^m, then f_c^m and F_c^m should be updated.

4. *Scout Bees Phase*: If there is no position better than the current one in its neighborhood after T_s iteration, where T_s represents the largest searching time, the old position will be given up while the scout bee will randomly generate a new position and start its searching. It is an effective process to remove local optimums.

The iteration will be repeated until the iteration time is up to the maximum value T. The position F_c^m with the maximum fitness value is decided as the optimal position for the cluster's center. Users covered by the circle with center F_c^m and radius r_{max} are stored in the set $\mathcal{L}^m$. The algorithm is presented in Algorithm 2.

When operating the proposed user clustering algorithm, Algorithm 2 is called to decide which users should be clustered into a group. In Algorithm 2, the complexity of initialization is $\mathcal{O}(N_p\left|\mathcal{K}_{local}\right|)$. The complexity of each iteration is $\mathcal{O}(2N_p\left|\mathcal{K}_{local}\right|)$. Thus, the complexity of Algorithm 2 is $\mathcal{O}(TN_p\left|\mathcal{K}_{local}\right|)$. In Algorithm 1, the complexity of line 3-5 is $\mathcal{O}(N_0)$ as a whole. Consequently, the total computational complexity of the algorithm is $\mathcal{O}(TN_p\left|\mathcal{M}\right|\left|\mathcal{K}_{local}\right|+\left|\mathcal{M}\right|N_0)$.

3.2 3D Deployment

In the previous subsection, we have divided users into different groups and ensured the number of groups to be as little as possible, which is also the number of UAVs. In this subsection, the 3D position of each UAV will be optimized to improve the QoS of served users and to reduce the total interference of the system. Taking the UAV m and the group $\mathcal{L}_m$ served by UAV m for example, a minimum service region will be derived first to exclude users not belonging to $\mathcal{L}_m$ and normal transmission from UAV m to users belonging to $\mathcal{L}_m$ will be guaranteed at the same time.

To minimize the service region of UAV m, we need a minimum circle to cover all users belonging to $\mathcal{L}_m$. The problem can be formulated as followed:

$$\min_{x_m,y_m}\ \max_{l\in\mathcal{L}_m}\ s_{m,l} \tag{16}$$

where $s_{m,l}$ denotes the distance between UAV m and user l in $\mathcal{L}_m$. It has been demonstrated that (16) is a convex optimization problem and can be solved by CVX [5]. The coordinates (x_m, y_m) , which are the center of the obtained minimum circle, are also the 2D position of UAV m. The result of (16) r_m is the service radius of UAV m, which is no larger than the maximum service according to the definition, i.e., $r_m < r_{max}$. Thus, we need to adjust the flight height of each UAV next.

According to Fig.2, when the service radius of UAV m is fixed, there is always an optimal height that can maximize the channel gain of UAV m. That is to say, users belonging to $\mathcal{L}_m$ can get better QoS if UAV is deployed at the optimal height. The optimal height h_m for UAV m to maximize the channel gain can be obtained by solving the following equation:

$$\frac{\partial \bar{g}(r_m, h_m)}{\partial h_m} = 0, \tag{17}$$

where r_m is the real service radius for UAV m.

4 Simulation Results

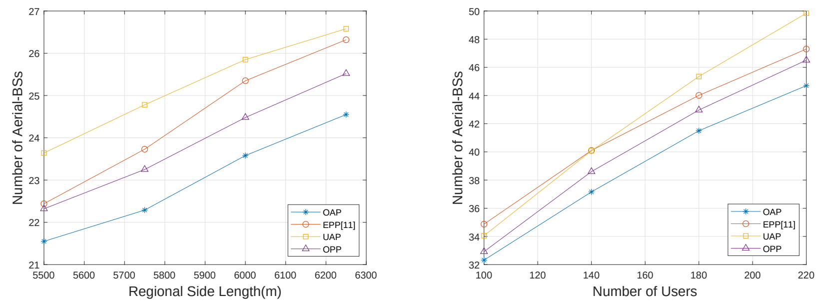
The simulation results are represented and analyzed in this section. In our simulations, users distribute randomly in a square area and the results in a sophisticated urban environment are considered. The parameters in our simulation are shown in Table 1.

Table 1. Simulation Parameters

Parameter	Value
a	11.95
b	0.14
β_0	5×10^{-5}
κ	0.01
T_{pso}, T_{abc}	1000
P_{num}, B_{num}	200
N_{max}	5

According these parameters, we can calculate that $r_{max} = 578\,m$. We choose three other algorithms to compare with the OAP algorithm proposed in this paper, which are Unordered ABC-based placement (UAP) algorithm, Ordered PSO-based Placement (OPP) algorithm and Edge-Prior placement (EPP) algorithm [11]. UAP algorithm picks k_0 randomly from uncovered users, which is different from OAP algorithm. Besides, T_{abc}, T_{pso}, P_{num} and B_{num} denote the maximum iteration times of ABC algorithm and PSO algorithm, the population of ABC algorithm and PSO algorithm, respectively.

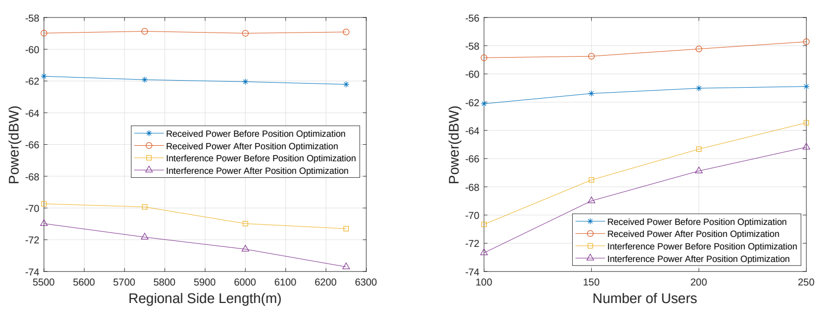
Fig.3 demonstrates the priority of minimizing the number of UAVs of our algorithm in a general way. Each point is averaged over 100 independent user distributions. In Fig.3(a), the comparison of the number of UAVs with users distributing in different areas is presented. It can be seen that no matter in which condition, OAP algorithm performs the best. Meanwhile, as the distribution area becomes larger, the density of users becomes smaller, so more UAVs are needed to serve all the ground users. In Fig.3(b), the comparison of the number of UAVs



(a) Comparison of required number of UAVs with varying regional area when using different algorithm, where the service ability is 5 and the number of users is 100.

(b) Comparison of required number of UAVs with varying numbers of users when using different algorithm, where the regional side length is 6000m and the service ability is 5.

Fig. 3. Comparison of number of required UAVs when using different algorithm



(a) Comparison of received power and interference power with varying regional area before and after position optimization, where the service ability is 8 and the number of users is 100.

(b) Comparison of received power and interference power with varying numbers of users before and after position optimization, where the regional side length is 6000m and the service ability is 5.

Fig. 4. Comparison of received power and interference power before and after position optimization

with different numbers of users is presented. OAP algorithm still performs the best and more UAVs are needed for serving more numbers of users.

Fig.4 shows the effect of our 3D deployment method on improving QoS in a general way. We take the received power and the interference power into consideration with transmitting power $P_t = 30\,dBW$. Fig.4(a) shows the change in received power and interference power before and after position optimization with different users distribution area. It can be seen that the received power always increases and the interference power always decreases obviously after position optimization. With the distribution area enlarging, the received power increases and the interference power decreases. Fig.4(b) shows the change in the received power and the interference power before and after position optimization with different number of users. It is also obvious that the received power increases and the interference decreases after position optimization.

In summary, the simulation results show that compared with other methods, our 3D deployment method performs better on reducing the number of UAVs. After optimizing the 3D position of each UAV, the QoS of wireless communication can be improved obviously.

5 Conclusion

A UAV-mounted wireless communication system was investigated and a 3D deployment method based on ABC algorithm was proposed in this paper. The algorithm could minimize the number of UAVs required to serve all ground users while ensure the QoS requirement of users. We first derived the maximum service radius of UAVs according to the QoS requirement. Then we proposed the OAP algorithm to deploy multiple UAVs, where the number of UAVs was minimized first and then the 3D positions of UAVs were optimized to improve the QoS of the system. Simulation results showed that our method has the superiority of minimizing the number of UAVs and improving the QoS compared with other algorithms.

References

1. Springer.: Handbook Of Unmanned Aerial Vehicles (2015)
2. Xiao, Z., Xia, P , Xia, X.G.: Enabling UAV cellular with millimeter-wave communication: potentials and approaches. IEEE Commun. Mag. **54**(5), 66-73 (2016).
3. Zeng, Y., Zhang, R, Lim, T.J.: Wireless Communications with Unmanned Aerial Vehicles: Opportunities and Challenges. IEEE Commun. Mag. (2016).
4. Al-Hourani A., Kandeepan S., Lardner S.: Optimal LAP Altitude for Maximum Coverage. IEEE Wireless Commun. Lett. **3**(6), 569-572 (2014).
5. Alzenad M., El-Keyi A., Yanikomeroglu H.: 3-D Placement of an Unmanned Aerial Vehicle Base Station for Maximum Coverage of Users With Different QoS Requirements. IEEE Wireless Commun. Lett. **7**(1), 38-41 (2018).
6. Alzenad M., El-Keyi A., Lagum F., et al.: 3D Placement of an Unmanned Aerial Vehicle Base Station (UAV-BS) for Energy-Efficient Maximal Coverage. IEEE Wireless Commun. Lett. **PP**(99), 1-1 (2017).

7. Lyu J. , Zeng Y. , Zhang R. , et al.: Placement Optimization of UAV-Mounted Mobile Base Stations. IEEE Commun. Lett. **21**(3), 604-607 (2017).
8. Strumberger, N., Bacanin, S. Tomic, M. Beko and M. Tuba,: Static drone placement by elephant herding optimization algorithm, 2017 25th Telecommunication Forum (TELFOR), pp. 1-4. Belgrade (2017). https://doi.org/10.1109/TELFOR.2017.8249469.
9. Wang G.G., Deb S., Coelho L.D.S.: Elephant Herding Optimization, 2015 3rd International Symposium on Computational and Business Intelligence. IEEE (2015).
10. Zhao H., Wang H., Wu W., et al.: Deployment Algorithms for UAV Airborne Networks Toward On-Demand Coverage. IEEE J. Select. Areas Commun. **36**(9), 2015-2031 (2018).
11. Qin J., Wei Z., Qiu C. and Feng Z.: Edge-Prior Placement Algorithm for UAV-Mounted Base Stations, 2019 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-6. Marrakesh, Morocco (2019). https://doi.org/10.1109/WCNC.2019.8885992.
12. Zeng Y., Wu Q. and Zhang R.: Accessing From the Sky: A Tutorial on UAV Communications for 5G and Beyond, Proc. IEEE **107**(12), 2327-2375 (2019).
13. Linderoth J.T., Savelsbergh M.W.P.: A Computational Study of Search Strategies for Mixed Integer Programming. INFORMS J. Comput. **11**(2), 173-187 (1999).
14. Karaboga, D.: An idea based on honey bee swarm for numerical optimization (2005).