A Bean Optimization-Based Cooperation Method for Target Searching by Swarm UAVs in Unknown Environments

XIAOMING ZHANG1,2 AND MOHSIN ALI1

1 School of Information and Computers, Anhui Agricultural University, Hefei 230031, China
2 Peng Cheng Laboratory, Shenzhen 518055, China

Corresponding author: Xiaoming Zhang (xmzhang@ustc.edu)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFD0301303.

ABSTRACT This paper studies the target searching problem using swarms of unmanned aerial vehicles (UAVs) in unknown environments which information is unknown to the UAVs, other than features they detect through their sensors. Effective decision and control methods are required for UAVs that consider their limitations and characteristics when confronted with target searching problems. A cooperative target searching method is proposed for swarm UAVs based on an improved bean optimization algorithm (BOA) called Robot Bean Optimization Algorithm (RBOA). Compared with conventional BOAs used for optimal computation, RBOA has two main modifications for the cooperative control of swarm robots: 1) it accounts for the free motion space of individual UAVs using a Thiessen polygon; and 2) it adds a free space search mechanism to improve the efficiency of target searching. Based on the above improvements, and by integrating a multi-phase search mechanism and scheduling control strategy, a swarm UAV collaborative search simulation platform is built for experimental purposes. The results obtained from search simulations show that the RBOA can outperform adaptive robotic particle swarm optimization (A-RPSO) in target searches in complex and unknown environments, especially with fewer evolutionary generations and smaller numbers of robots. The RBOA, which is inspired by plant population evolutionary patterns, has fast and effective search capabilities, distributed collaborative interaction, and emergent swarm intelligence. It provides new ideas and support for research into the control of swarm UAVs and swarm robots.

INDEX TERMS Swarm intelligence, cooperation, swarm UAVs, BOA, target search.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) originated in the military and are a type of aircraft that is unmanned and remotely or autonomously controlled. Swarm UAVs are a type of swarm-robot system, and research on them has become an important direction in the development of unmanned and automated technologies. By collaborating with each other, swarm UAVs demonstrate superior coordination, intelligence, and autonomy to manually-controlled systems. After decades of slow development, they have entered a period of more rapid development and their applications are increasing. In military applications, UAVs can perform tasks such as reconnaissance, surveillance, interference, and air-to-ground strike. In the civil field, UAVs can be used for target searching, agricultural plant protection, forest fire prevention, power (pipeline) inspection, and geological exploration. Through their various capabilities and coordinated actions, the efficiency of swarm UAVs can be improved. Hence, the application of UAVs has gradually developed from a single platform to a multi-platform cluster. However, without a rational and efficient decision-making method and control strategy, it is difficult for swarm UAVs to provide an advantage over single UAVs. There may be contradictions between UAVs on temporal, spatial, and task levels, and the risks of conflict and collision can make it impossible to complete established tasks. Therefore, establishing an efficient cluster management and control system for UAVs is of great practical significance for dealing with complex, dynamic, and uncertain environments, as well as to maximize UAV performance. The characteristics of distribution, self-adaptation, and robustness embodied in biological swarm
behavior are consistent with the requirements for the coordinated autonomous control of swarm UAVs. Studying the internal mechanisms involved in biological swarm behavior and adapting them to the coordinated autonomous control of swarm UAVs can improve the decision-making and planning abilities of UAVs in complex environments.

Swarm behavior is a ubiquitous natural phenomenon. In mammals, birds, fish, insects, vegetation, and even bacteria, there are complex swarm behaviors at different scales of life. A biological swarm not only forms a coordinated and orderly collective movement pattern, it also responds to external stimuli quickly and consistently, thereby showing the characteristics of distribution, self-organization, collaboration, stability, and environmental adaptability. The intrinsic mechanism and the law of action of this efficient and flexible movement mode have long been the core issues in the study of biological swarm intelligence.

Swarm intelligence comes from the intelligent behavior of cooperating biological groups and has the characteristics of being distributed, centerless, and self-organizing. Since Beni and Wang introduced this concept in their research on cellular robotic systems, swarm intelligence has attracted much research attention. Inspiration from various organisms has been used to construct a series of algorithms, such as the ant colony algorithm (ACO) [1], particle swarm optimization (PSO) [2], krill herd algorithm (KHA) [3], monarch butterfly optimization (MBO) [4], bacterial foraging optimization algorithm (BFO) [5], firefly algorithm [6], differential evolution (DE) and the bee optimization algorithm (BOA) [7]. Swarm intelligence algorithms are also a successful optimization technology and have increasingly been used to solve complex optimization problems in recent years, such as for scheduling [8], parameter estimation [9], feature selection [10], and swarm robotic systems [11].

Target searching is one of the most common tasks undertaken by swarm robotic systems. For example, in reference [12], nature-inspired algorithms (such as PSO, BFO, and the bat algorithm, BA) were used to explore unknown search areas, and PSO was found to provide the best search performance. Reference [13] used robots to search for lost targets in unknown environments. As a typical swarm robot system, swarm UAVs have also been applied to target searching. Cooperative control is very important in this application [14], and many scholars have carried out research into cooperative searching with UAVs. For example, reference [15] used UAVs to search for lost targets in unknown environments. The offline planning method constructed in reference [16] divides the task area and achieves full coverage of it. Reference [17] designed a network of UAVs to cooperatively search for multiple targets. It guides UAVs to regions with high probabilities of containing targets to search more rapidly; however, the precision of the decision results is low. Reference [18] is based on the distributed model predictive control framework. It combines Nash optimality and PSO to effectively reduce the scale and communication burden of the problem.

Although the above research has enabled swarm UAVs to cooperatively search to some extent, they still face the following challenges: (1) Lack of an effective cooperation structure. Without one, the efficiency of target searching may be reduced. (2) Crowding and collisions between multiple UAVs. (3) A lack of prior information about the environment. (4) A lack of a special guidance mechanism to guide UAVs in unknown areas. In order to solve the above problems, researchers are working to discover control strategies for swarm UAVs based on swarm intelligence.

The BOA is a kind of swarm intelligence method inspired by the evolution of natural plant distributions. By analyzing the self-adapting evolutionary law of plant populations over millions of years, this method can abstract the dynamic processes of survival and reproduction in approximately static plant populations by greatly compressing the time axis. When solving complex optimization problems, the BOA has a relatively fast optimization speed and outstanding adaptability to dynamic environments. It demonstrates amazing swarm adaptive survivability of plants and is suitable for researching swarm distributions, collaboration, and the emergence of swarm intelligence. Hence, it has the potential for use in swarm robot systems.

In this paper, a robot bean optimization algorithm (RBOA) is proposed and applied to simulate a target search task in unknown complex environments by swarm UAVs. The main characteristics of the RBOA include: (1) a free motion space, (2) an internal search mechanism, (3) a multi-phase search mechanism, and (4) a scheduling control strategy. In the simulated environments, none UAVs have information about the search area and target location, but all know that the search area is bounded. We also assume that the information that can be obtained from the environment includes the altitude of the area, which enables the UAVs to sense it and determine a locally advantageous direction. Similarly, UAVs can be equipped with sensors (for example, active laser ranging sensors) to detect spatial information. In addition, UAVs can communicate with each other.

The rest of this paper is organized as follows: Section 2 reviews the related literature on BOAs. Section 3 describes the proposed method in detail. The experimental setting is explained in Section 4, then the effectiveness of the proposed method is verified by experiments in different environments. Finally, Section 5 summarizes the paper.

II. RELATED RESEARCH INTO BOAS

At present, a large number of researchers are working to develop more efficient and practical swarm intelligence optimization algorithms. Research into swarm robot systems based on plant bionics is currently rare, mainly because it is not easy to associate the emergent mechanisms of plant swarm intelligence with the construction of swarm robots. However, ecological research shows that the evolution of a plant population over millions of years also contains swarm intelligence, which is worthy of further study and utilization.
The BOA is inspired by the natural propagation mode of seeds and the evolution of population distributions. By analyzing the self-adaptive evolutionary law of excellent plant populations over millions of years, and by greatly compressing the time axis, the processes of survival and reproduction in an approximately static plant population can be represented dynamically. The BOA optimization mechanism is presented dynamically. The BOA optimization mechanism is different to that of existing swarm intelligence optimization algorithms. Two operators—known as paternity selection and population distribution evolution—are used for optimization. The structure of the algorithm is relatively simple and easy to implement. Previous studies and experiments on the algorithm show that it has a strong global optimization ability and fast optimization speed. This suggests that the evolutionary strategy of plant population distribution is an excellent natural evolutionary strategy that is worthy of in-depth research and continued application.

A. OVERVIEW OF PLANT POPULATION DISTRIBUTION

The evolution of a plant population distribution is also known as the population distribution pattern. It is the result of long-term adaptation, self-evolution, and selection of the biological characteristics of the population by the environment [19]. Research on the evolution of plant population distributions has progressed from a qualitative descriptive stage to a quantitative calculation stage. Existing methods include the chi-square test, the variance/average ratio, the Moore test, the cluster index, the Morisita dispersion index, and so forth. Scientific research not only promotes the development of ecology but also provides a theoretical basis for bionic research. At present, the distribution patterns of plant populations in nature [20] are mainly random, uniform, and clustered distributions.

A random distribution means that there is no association between each individual plant in the population and other individuals and groups. Each individual has the same probability of appearing at a certain position and has no relationship to the locations where other individuals have already appeared. Studies have shown that a random distribution of plant individuals generally occurs when the horizontal distribution of environmental factors is within many ecological ranges and the degrees of their effects are approximately equal, or when the environment is extremely harsh. Therefore, over large spatial ranges the habitat heterogeneity is greater, while over small areas there is greater uniformity of habitat factors. Mathematical models of random distribution patterns of plant populations generally follow a Poisson distribution.

A uniform distribution means that individual plant populations are uniformly and equidistantly distributed in space. If uniformly distributed populations are sampled, the number of individuals in a sampling unit is close to the average number of individuals, and the numbers of sampled individuals that are greater or less than the average number of individuals are small. Many populations are uniformly distributed due to competition within a population for a limited living resource, and autotoxicity of individuals. Mathematical models of plant populations with uniform distribution patterns generally follow a positive binomial distribution.

A clustered distribution is also known as an aggregate distribution. This type of plant population distribution is characterized by a dense distribution that is clustered into subpopulations. However, the size of each subpopulation, the distance between them, and the densities of individuals within the subpopulations are different. Generally, each subpopulation has an internal random distribution. In nature, the evolutionary distribution of plant populations is generally clustered. The main causes of clustered distribution patterns are [20]: 1) limits in the propagation distance of seeds, 2) spatial heterogeneity in the environment, and 3) interspecific relationships. Mathematical models of clustered distribution include normal distribution, negative binomial distribution, Neyman distribution, and Polya-Eggenberger distribution [21].

B. BASIC PRINCIPLE OF THE BOA

Based on the idea of the evolution of plant populations in nature, the BOA has been proposed based on a clustered distribution. The basic settings are as follows.

In BOA, the position of an individual is represented by a real number vector, \( X = \{x_1, x_2, x_3, \ldots, x_n\} \). Here, \( n \) represents the dimension of the problem space. Among the individuals, the ones with the best fitness are called the parent individuals. The number and distribution of their offspring are determined according to the fitness value of the parent individuals. The distribution of the individuals is as follows:

\[
X_{ij}(t + 1) = \begin{cases} 
X_i(t), & \text{if } X_{ij}(t + 1) \text{ is a parent} \\
G(X_i(t)), & \text{if } X_{ij}(t + 1) \text{ is not a parent} 
\end{cases}
\]

where

1. \( X_{ij}(t + 1) \) refers to the position of individual \( j \) generated by parent \( i \) in the \( (t + 1)\text{th} \) generation;
2. \( X_i(t) \) refers to the position of parent \( i \) in the \( t\text{th} \) generation; and
3. \( G(X_i(t)) \) refers to a distribution function of a simulated plant population distribution pattern with \( X_i(t) \) as an important parameter.

For example, the distribution of the offspring of a BOA based on normal distribution is as follows:

where

1. \( N(\mu, \delta) \) refers to a normal distribution with a probability density function \( f(X) \):

\[
f(X) = 1/([\delta(2\pi)^{1/2}]\exp[-(X-\mu)^2/(2\delta^2)]) \\
-\infty < X < +\infty
\]

2. edge refers to the edge distribution point of the problem domain;
3. \( D_{\text{min}} \) refers to the shortest Euclidean distance from the point to the edge.
4. \( H(x) \) is a linear function of \( x \).
At present, research has been conducted on the preliminary design and implementation of the BOA [7], initial improvement of the algorithm, preliminary theoretical analysis of the algorithm [22], preliminary application experiments, improvement of the population distribution evolution model [23], intersection of a chaotic idea [24], typical dynamic optimization problem-solving, and simple multi-objective optimization problem-solving. This lays a preliminary foundation for application of the BOA to swarm robots. Considering the constraints of the search algorithms used for swarm robots [25], the BOA can be used for the cooperative control of the target search by swarm UAVs.

### III. BOA-BASED COOPERATION METHOD FOR TARGET SEARCHING BY SWARM UAVS

With the aim of achieving swarm UAV target searching, we designed and constructed the RBOA based on the BOA by adding free motion space, internal search mechanism, multi-phase search mechanism, and scheduling control strategy.

#### A. RBOA ALGORITHM DESIGN

As with other swarm intelligence algorithms, the BOA is mainly used to solve complex optimization problems and cannot be directly used for the cooperative control of swarm robots. Therefore, to meet the operational requirements of swarm robots, an RBOA was designed that includes swarm initialization, a free motion space strategy, an iterative scheduling strategy, distribution models, and an individual quantity allocation method. We will use swarm UAVs as an example with which to elaborate on the design of the RBOA.

1) **SWARM INITIALIZATION**

Initialization mainly involves setting the number of individuals in the UAV swarm, the number of parent UAVs, and their distance threshold. It also includes selection of a plant population distribution model and setting of its parameters. The swarm initialization expression is as follows:

\[
INI(N) = \text{rand}(N, M, d_m, G(X), R)
\]

where \(\text{rand}(\cdot)\) is a random function assigned to a position with five parameters, \(N\) is the total number of individuals, \(M\) is the number of parent UAVs, \(d_m\) is the distance threshold between parent UAVs, \(G(X)\) is the plant population distribution model, and \(R\) is the individual search range of the UAVs.

2) **SPATIAL DISTRIBUTION OF INDIVIDUAL UAVS**

The space where the swarm is located is divided into a parent UAV layer, an individual UAV layer, and a temporary dispatch layer from bottom to top. Parent UAVs are located at the parent UAV layer and individual UAVs are located at the individual UAV layer. The temporary dispatch layer is used to adjust the positions of the individual UAVs, as shown in Fig. 1.

3) **DISTRIBUTION MODEL FOR THE FIRST PHASE SEARCH**

(1) The number of UAVs can be allocated according to the fitness value of the parent UAVs. The allocation method is as follows:

\[
N_{CBi} = P_i * N,
\]

\[
P_i = \frac{f(FB_i)}{\sum_{i=1}^{M} f(FB_i)} + \alpha_i,
\]

where \(N_{CBi}\) is the number of individual UAVs deployed by parent UAV \(i\), \(P_i\) is the proportion of the allocation assigned to the parent UAV \(i\), \(f(FB_i)\) is the current fitness value of parent UAV \(i\), indicating its rank. \(FB_i\) is the current location of the parent UAV \(i\). \(\alpha_i\) is the offset of the ratio of the parent UAV \(i\) (default value is 0).

(2) **Determination of the location of individual UAVs**

The position \(X\) of an individual UAV deployed by a parent UAV is determined according to the current position of the parent and the typical preset plant population distribution model (a normal distribution model is used in this paper).

\[
X_{ij}(t + 1) = \begin{cases} 
X_i(t), & \text{if } X_{ij}(t + 1) \text{ is a parent individual} \\
N(X_i(t), H(D_{\min}[X_i(t), \text{edge}])), & \text{if } X_{ij}(t + 1) \text{ is not a parent individual}
\end{cases}
\]

\[
\text{FIGURE 1. Spatial distribution of individual UAVs.}
\]
The details are as follows:

$$g(X) = 1/\left[\delta_i(2\pi)^{1/2}\exp\left[-(X-\mu)^2/(2\delta_i^2)\right]\right],$$  \hspace{1cm} (7)

$$\mu_i = FB_{i1},$$  \hspace{1cm} (8)

$$\delta_i = d_r + [(d_{max})/(N_{CBi})] \times \alpha_{bi},$$  \hspace{1cm} (9)

where $\delta_i$ is the degree of dispersion of the distribution of individual UAV locations. $\mu_i$ is the concentrated trend position of the distribution of individual UAVs. $d_r$ is the minimum safe moving distance between UAVs. $d_{max}$ is the boundary distance of the work area. $\alpha_{bi}$ is the offset of the discrete position distribution of the UAVs.

In the case of a single parent UAV, the individual UAV is determined based on the distribution model of the parent. For multiple parent UAVs, the predetermined parent UAV spacing threshold $d_m$ is used to further expand the working range of the swarm UAVs to improve their operational effectiveness in complex environments.

4) UAV FREE MOTION SPACE AND SEARCH STRATEGY

The individual free motion space of the UAV is constructed based on the Thiessen polygon, which reduces the complexity of motion planning between the UAVs, greatly reduces the probability of collisions, and reduces the energy consumption of UAV motion planning and obstacle avoidance behavior. Then, free motion sequence points are determined by using the vertices and initial position as the base point group, and random behavior is incorporated under the premise of the global area coverage. This increases the adaptability and effectiveness of the swarm UAVs in complex working environments, as well as reducing the blindness of completely random motion and the complexity of full coverage path planning.

As shown in Fig. 2, the individual UAVs are irregularly distributed in space. The parent UAV computes the distribution and free motion space for its individual UAVs. Individual UAVs explore target information and communicate with each other in their free motion spaces. At the same time, the information is integrated and transmitted to the parent UAV. The specific steps are as follows:

(1) As shown in Fig. 3, the Thiessen polygon is generated by using each UAV as a control point, and the space is divided into a plurality of independent regions. The independent

region $R_k$ is a free motion space of the individual UAV $IN_k$:

$$R_k = \{x \in R_k | d(x, IN_k) < d(x, IN_j), j = \{1, 2, \ldots, N\}\},$$  \hspace{1cm} (10)

where $d(x, y)$ is the distance between point $x$ and the control point $y$, and $k \neq j$.

According to the characteristics of the Thiessen polygon, it can be ensured that there is only one UAV in each area, and the distance from the position in the area to the internal control point is shorter than the distance to the external control point.

(2) The vertices of the free motion space of each individual UAV are sorted in order, and a sequence of vertices is generated ($P_1, P_2, P_3, \ldots, P_N$):

(3) A line segment of the control point $IN_1(x_0, y_0)$ to each vertex is constructed.

Herein, the line segment to the vertex $P_1(x_1, y_1)$ is expressed as:

$$(x-x_1)/(y-y_1) = (x_0-x_1)/(y_0-y_1)$$  \hspace{1cm} (11)

(4) Random points are generated on each line segment as trajectory points for the UAVs moving in the free motion space. Then, the trajectory points $P_{11}, P_{12}, P_{13}, \ldots, P_{1m}$ are generated.

According to the characteristics of the Thiessen polygon, each polygon is convex. This ensures that the trajectory of the UAV’s movement is within its free motion space. Taking vertex $P_1$ as an example, its random point $P_{11}(x_{11}, y_{11})$ is generated as follows:

$$\left(\begin{array}{c} x_{11} \\ y_{11} \end{array}\right) = \gamma \times \left(\begin{array}{c} x_0 \\ y_0 \end{array}\right) + (1-\gamma) \times \left(\begin{array}{c} x_1 \\ y_1 \end{array}\right),$$  \hspace{1cm} (12)

$$\gamma = rand(1, time), 0 < \gamma < 1,$$  \hspace{1cm} (13)

where the random parameter $\gamma$ is a random number generated according to the current time.

(5) As shown in Fig. 4, starting from vertex $P_1$, the target trajectory points $P_{11}, P_{12}, \ldots, P_{1m}$ are sequentially connected to generate a target motion trajectory of the individual UAV in the free motion space. The target motion trajectory is expressed as follows:

$$IN_1 \rightarrow P_{11} \rightarrow P_{12} \rightarrow \ldots \rightarrow P_{1m},$$

where $IN_1$ is the current location of the individual UAV $IN_1$.
In the same way, the motion trajectories of the individual UAVs $IN_2$, $IN_3$, ..., $IN_n$ are sequentially generated.

(6) The individual UAV performs the exploratory task in its free motion space according to the motion trajectory, and updates the optimal fitness value and corresponding position information, which are sent to its parent UAV.

5) SCHEDULING STRATEGY FOR UAVS

Swarm UAVs always include a large number of individuals and there are generally no obstacles in the vertical direction. By adding a temporary scheduling layer and performing a vertical ascent action, it is convenient to dispatch individual UAVs to the temporary scheduling layer. Then, according to its fitness value, each UAV descends to the individual UAV layer for horizontal position scheduling. This strategy can greatly reduce excessive obstacle avoidance and complex path planning behaviors within the same horizontal plane of position scheduling and improve execution efficiency, as well as reduce energy consumption and accident rates.

The number of parent UAVs is much smaller than the number of individual UAVs. Arranging an independent layer avoids excessive obstacle avoidance and complex path planning behavior in location scheduling, which can improve execution efficiency and reduce energy consumption and accident rates. Because the parent UAVs are at a lower level, they can conveniently carry out target confirmation and subsequent operations that improve the effectiveness and accuracy of the operation.

As shown in Fig. 1, the parent UAV performs horizontal position scheduling with the individual UAVs that have the highest fitness values, and is dispatched to the area with the highest target confidence. The specific steps are as follows:

(1) The parent UAV moves to the parent UAV layer and goes to the area with the highest target confidence.

(2) Based on the new location information of the parent UAV, a new location sequence for individual UAVs satisfying the distribution parameters is generated according to the preset plant population distribution model.

(3) Verification of the position sequence points is undertaken. The sequence points with distances less than or equal to the safety distance between UAVs are deleted, and new sequence points are regenerated until the position sequence points all meet the requirements.

(4) The individual UAVs ascend vertically to the temporary scheduling layer.

(5) The individual UAVs are sorted in order of descending fitness value. Then, they are respectively scheduled from the temporary scheduling layer to the individual UAV layer and move to the coordinate position of sequence point $PC_i$, $i = 1, 2, ..., n$.

(6) Positional scheduling continues until the last individual UAV, $n$, moves to $PC_n$.

6) SEARCH STRATEGY IN THE TARGET AREA FOR THE SECOND PHASE

As shown in Fig. 5, the parent UAV uses a negative binomial distribution, a kind of plant population distribution model, to determine the path trajectories required for the detailed search within the suspected target area. The specific steps are as follows:

(1) The track point sequence of the parent UAV is determined according to the real continuous negative binomial (NB) distribution model. Its specific description is as follows:

$$nb(X) = \Gamma (r + x)/[\Gamma (x) \Gamma (x + 1)],$$

$$\Gamma (x) = \int_0^{+\infty} t^{x-1} \exp (-t) dt,$$

where $\Gamma (x)$ is a gamma function on the real number field and $r$ is the positional coordinate of the current parent UAV.

(2) A sequence of points is randomly generated from the generated track points and connected in sequence, forming a detailed search route for the parent UAV. The parent UAV moves along the search route to achieve detailed search operations within the target area.

B. SIMULATION OF SWARM UAVS BASED ON PLANT POPULATION DISTRIBUTION EVOLUTION

Based on the RBOA, we now construct a swarm UAV simulation based on plant population distribution evolution. The following steps are included in RBOA:

- Initial Position
- Track Points

---

**FIGURE 4.** The motion trajectories of the individual UAV $IN_1$.

**FIGURE 5.** Search strategy in the target area.
Step 1. The swarm UAVs are initialized. This includes setting the numbers of individual and parent UAVs used to perform the task, the distance threshold of the parent UAV, and the plant population distribution model and its parameters. Then, the individual UAVs are randomly distributed in the task space.

Step 2. Using each individual UAV as a control point, a Thiessen polygon is generated to divide the space into several independent regions. There is only one individual UAV in each area.

Step 3. A search path for each individual UAV is generated. During the specified time, the individual UAV performs a search task in the area where it is located and determines its best fitness value before the end of the period.

Step 4. If the task requirements have been met, the task execution of the UAV ends. Otherwise, Step (5) begins.

Step 5. Based on the fitness values of the individual UAVs, the parent UAV (location) of the current swarm is selected. If the number of parent UAVs in the swarm exceeds one, the distance threshold between parent UAVs must be met.

Step 6. Determine whether the current number of iterations reaches the threshold set for the start of the second phase or not. If true, go to Step 9; otherwise, go to Step 7.

Step 7. The new positions of the descendant individual UAVs are determined based on the position of the parent UAV and the preset plant population distribution model.

Step 8. The swarm UAVs update their locations and begin to perform Step 2.

Step 9. The track point sequence of the parent UAV is generated according to the NB distribution model.

Step 10. The search route for the parent UAV is generated by connecting the track point sequence.

Step 11. The parent UAV searches along the route to achieve detailed search operations within the target area.

Step 12. Update the best location and begin to perform step 9 until the task requirements have been met.

The pseudo code for the RBOA (First Phase) is as follows:

Algorithm 1 Pseudo code for the RBOA (First Phase)

1. Create and initialize swarm with \( n \) UAVs;
2. Initialize parameters;
3. Repeat
4. for each UAV \( i = 1, \ldots, n \) do
5. CalculateFitness (UA Vi)
6. end
7. Sort(Fitness);
8. ParentUAVSelect();
9. AllocateOffspring() using Eq. (6);
10. CalculateDistribution() using Eq. (7);
11. CalculateFreeMovementSpace() using Eq. (10);
12. for each UAV \( i = 1, \ldots, m \) do
13. SearchFreeMotionSpace (i);
14. end
15. UpdateIndividualLocation();
16. until stop condition is true;

satellite system (BDS) and other related equipment to determine locations within a certain area. At the same time, the communication range among the swarm UAVs can cover the entire area.

(3) Two-dimensional area and boundary determination.

In the experiments, we assume that the edge of the area to be searched is clear. The range used for the target search is determined using a two-dimensional rectangular area. Swarm UAVs can go anywhere outside of the safe distances of other UAVs in the area.

(4) Ignore the impact of severe weather and airflow on the stability of the UAVs.

B. EXPERIMENTAL DESCRIPTION

To verify the effectiveness of the proposed algorithm, we present ten two-dimensional benchmark functions to simulate different areas that need to be detected. The most straightforward assumption is to let the swarm UAVs find the lowest altitude in the area, which is the optimal value of the objective function. All experiments were simulated in Matlab.

1) EXPERIMENTAL EQUIPMENT AND SOFTWARE

Hardware configuration: Intel Core i7-5600U, 16 GB RAM. Software configuration: Matlab R2014a, Multiparametric Toolbox 3.0.

2) BENCHMARK FUNCTIONS FOR BUILDING THE ENVIRONMENT

Ten benchmark functions are selected for building the target searching environments, and their variables are in two dimensions. The reason for choosing these ten benchmark functions is that their function figures (simulated experimental environment terrains) are complex and contain many local traps. It is not easy to find the target area with few evolutionary generations and a small number of individuals for most intelligent algorithms. A list of typical selected benchmark functions is
TABLE 1. Benchmark functions.

| Function Name | Mathematical Representation | Initial Range | Best Function Value |
|---------------|----------------------------|---------------|---------------------|
| Sphere        | $F_1(x) = \sum_{i=1}^{n} x_i^2$ | [-100,100] | 0 |
| Schwefel 2.22 | $F_2(x) = \prod_{i=1}^{n} |x_i| + \sum_{i=1}^{n} |x_i|$ | [-10,10] | 0 |
| Schwefel 1.2  | $F_3(x) = \prod_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$ | [-100,100] | 0 |
| Schwefel 2.21 | $F_4(x) = \max_i(|x_i|, 1 \leq i \leq n)$ | [-100,100] | 0 |
| Rosenbrock    | $F_5(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1))^2$ | [-30,30] | 0 |
| Shaffer’s F6  | $F_6 = 0.5 + \left(\frac{\sin(x_1^2 + x_2^2)^{1/2}}{1.0 + 0.01(x_1^2 + x_2^2)}\right)$ | [-600,600] | 0 |
| Quartic       | $F_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}(0, 1)$ | [-1.28,1.28] | 0 |
| Schwefel 2.26 | $F_8(x) = -\sum_{i=1}^{n} (x_i \sin(|x_i|^{1/2}))$ | [-500,500] | -12,569.483 |
| Rastrigin     | $F_9(x) = \sum_{i=1}^{n} (x_i^2 - 10 \cos(2\pi x_i) + 10)^2$ | [-5.12, 5.12] | 0 |
| Ackley        | $F_{10}(x) = -20 \exp\left(-0.2 \left[\left(1/n \sum_{i=1}^{n} x_i^2\right)^{1/2}\right]\right) - \exp\left((1/n) \sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$ | [-32,32] | 0 |

provided in Table 1. Their figures can be seen in pages 27-32 of reference [26]. The functions are marked F1 to F10 from top to bottom.

3) COMPARISON ALGORITHM AND PARAMETER SETTINGS
The PSO and its extended version are widely used for robot target searching. A comparison of the many intelligent algorithms was made in [27], which concluded that RDPSO was best in almost all experiments. In [25], Dadgar et al. proposed A-RPSO and compared it with the RPSO and RDPSO algorithms, concluding that A-RPSO performed better. Therefore, we use the A-RPSO algorithm in the comparative experiments. In the algorithm comparison, RBOA-X denotes that the population size of RBOA is X, while A-RPSO-X means that the population size of A-RPSO is X. The parameter settings of the algorithms are shown in Table 2. The stop condition in the experiments is that the number of evolutionary generations is more than maxgen.

C. EXPERIMENTAL RESULTS AND ANALYSIS

1) EXPERIMENTAL RESULTS
For the experiment results of functions F1 to F10, Table 3 lists the standard deviations (Stdev), optimal results (Min), worst results (Max), and average results (Average) for RBOA-10, RBOA-30, ARPSO-10, and ARPSO-30 after 50 runs.

A comparison of the average convergence curves of the algorithms is shown in Fig. 6, wherein the X coordinate indicates the number of generations and the Y coordinate indicates the average fitness value. Each curve represents the average convergence of 50 experiments for every environmental condition using every algorithm. The Y coordinates are shown on a logarithmic scale to increase the clarity of the curves.

Taking the benchmark function F9 as an example, one experiment was randomly selected to show the changes in UAV distribution while running RBOA-30 and ARPSO-30, as shown in Figs. 7 and 8, respectively. In the two sets of figures, the first sub-figure represents the random initialization of the UAVs. In Fig. 7, the 8th sub-figure shows the distribution of individual UAVs (at the end of the first phase). In particular, in order to clearly show the path points of the local fine-scale search with RBOA-30 (second phase), the last two sub-figures are enlarged.

2) EXPERIMENTAL ANALYSIS
It can be seen from the results of the target search experiments that, with the number of generations and population size held constant:

TABLE 2. Parameter settings of the algorithms.

| RBOA                           | ARPSO                        |
|--------------------------------|------------------------------|
| (1) Maximum number of evolutionary generations: $\text{maxgen} = 10$.          | (1) Maximum Number of evolutionary generations: $\text{maxgen} = 10$.          |
| (2) Ratio of the generations arranged to normal distribution to the negative binomial distribution: $nph = 0.7$. | (2) Inertial weight: $\text{win} = 1$. |
| (3) Mean value of the normal distribution is the position of the current parent UAV. | (3) Acceleration coefficient: $c_1 = 2$. |
| (4) Variance of the normal distribution: $\text{variance} = (\text{popmax} - \text{popmin}) \times 0.03$. | (4) Upper bound: $U = 2$. |
| (5) Probability of success of the negative binomial distribution: $\text{popn} = 0.5$. | (5) Lower bound: $L = 0.5$. |
| (6) Number of UAVs: $\text{popsize} = 10$ or 30. | (6) Number of UAVs: $\text{popsize} = 10$ or 30. |
| (7) Number of experimental repetitions: $\text{niter} = 50$. | (7) Number of experimental repetitions: $\text{niter} = 50$. |
| (8) Number of random traversals inside the free motion area: $\text{timer} = 2$. | |
FIGURE 6. Average convergence curves of the algorithms.
In terms of the average results, it can be seen from the table of experimental results that the RBOA-10 and RBOA-30 algorithms both obtained relatively optimal values. It can also be seen from the average convergence curves that in experiments involving the same number of individuals, the RBOA algorithm had faster optimization speed,
especially in cases with many beginning generations, which has obvious advantages.

(2) In terms of the standard deviations of the final results for each experiment, RBOA-10 achieved smaller standard deviations in functions F1, F3, F4, F5, F7, and F8, while A-RPSO-10 achieved smaller standard deviations in functions F2, F6, F9, and F10. RBOA-30 achieved smaller standard deviations in functions F1, F2, F3, F4, F5, F6, F7, and F10, while A-RPSO-30 achieved smaller standard deviations in functions F8 and F9.

(3) In terms of the worst values of the final experimental results, RBOA-10 obtained better values in functions F1, F3, F4, F5, F7, and F8, while A-RPSO-10 achieved better values in functions F2, F6, F9, and F10. RBOA-30 achieved better values in functions F1, F2, F4, F5, F6, F7, F8, F9, and F10, while A-RPSO-30 achieved a better value in function F3.

(4) Functions tic and toc in MATLAB are used to record the execution time of each search experiment. According to the statistics of experimental data, the average search time of A-RPSO-10, A-RPSO-30, RBOA-10 and RBOA-30 is 1.4665, 1.8716, 0.2123 and 0.4253 respectively. Compared with A-RPSO, RBOA has better algorithm efficiency.

It follows an analysis of the algorithm experiments and results:

(1) With the same population size, based on the free motion space partitioning and free exploration mechanism, RBOA is better able to explore space than A-RPSO. Taking the free-motion space polygon region with five vertices as an example, an individual can add five random search locations distributed in the space. Therefore, search performance of the RBOA algorithm is faster than that of the A-RPSO, especially in terms of evolution results with multiple beginning generations.

(2) The local exploration mechanism of RBOA in the later generations allows RBOA to carefully explore the discovered target area, which can further optimize the search results. However, the premise of this mechanism is that the algorithm has found the target area, and that the parameters of
the negative binomial distribution are set properly. Therefore, in some experiments, this mechanism did not work as expected. In some experiments, the virtual UAV became trapped in a local region, resulting in worse results and larger standard deviations than those obtained with A-RPSO.

(3) One of the reasons why RBOA can cause trapping in a local optimal region in few generations lies in removing the random individual mechanism. Thus, the global search performance of the algorithm is further reduced. The main reason for the removal of this mechanism is because, considering an experiment using real UAVs, random individuals are highly likely to conflict, which is not acceptable.

V. CONCLUSION

In this paper, an RBOA designed to solve a target searching problem is proposed. The limitations of UAVs and the constraints of the search algorithm were considered. The RBOA has three advantages, which result in a higher level of performance compared with other approaches.

First, based on the Thiessen polygon, an individual free motion space is constructed, which increases the efficiency of the search and reduces the probability of conflict between individuals. Second, the random search mechanism in free motion space increases the intensity of regional searching. Finally, the multi-phase search mechanism increases the optimization intensity and reduces the swarm consumption.

To verify the performance of the proposed algorithm, several experiments were performed in a simulation. Experimental results show that the performance of the RBOA is better than A-RPSO, which is more significant compared with other approaches in a complex environment and when the numbers of UAVs and generations are small. This study focused on a single target searching problem. For multi-target searching problems, most algorithms will divide a swarm into several sub-swarms. Since our approach performs well with small populations and has a sub-swarm search mechanism, we expect our approach to perform well in multi-target scenarios. An extension of the RBOA for multi-target searching is a topic of our future research. After the conditions related with experimental equipment are met, we will also validate our approach using real UAVs and unmanned underwater vehicles.

REFERENCES

[1] M. Dorigo, “Ant colony optimization: Overview and recent advances,” in Handbook Metaheuristics (Operations Research and Management Science), vol. 272, M. Gendreau and J. Y. Potvin, Eds. Cham, Switzerland: Springer, 2019, pp. 311–351.
[2] Y. Zhang, D.-W. Gong, and J. Cheng, “Multi-objective particle swarm optimization approach for cost-based feature selection in classification,” IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 14, no. 1, pp. 64–75, Jan. 2017.
[3] G.-G. Wang, S. Deb, A. H. Gandomi, and A. H. Alavi, “Opposition-based krill herd algorithm with cauchy mutation and position clamping,” Neurocomputing, vol. 177, pp. 147–157, Feb. 2016.
[4] G. G. Wang, S. Deb, and Z. Cui, “Monarch butterfly optimization,” Neural Comput. Appl., vol. 31, no. 7, pp. 1995–2014, 2019.
[5] B. Hernandez-Ocana, O. Chavez-Bosquez, J. Hernández-Torruco, J. Canul-Reich, and P. Pozos-Parra, “Bacterial foraging optimization algorithm for menu planning,” IEEE Access, vol. 6, pp. 8619–8629, 2018.
[6] I. Fister, M. Perc, S. M. Kamal, and I. Fister, “A review of chaos-based firefly algorithms: Perspectives and research challenges,” Appl. Math. Comput., vol. 252, pp. 155–165, Feb. 2015.
[7] X. Zhang, B. Sun, T. Mei, and R. Wang, “A novel evolutionary algorithm inspired by beans dispersal,” Int. J. Comput. Intell. Syst., vol. 6, no. 1, pp. 79–86, Feb. 2013.
[8] H. Lu, Q. Zhou, Z. Fei, and R. Zhou, “Scheduling based on interruption analysis and PSO for strictly periodic and preemptive partitions in integrated modular avionics,” IEEE Access, vol. 6, pp. 13523–13540, 2018.
[9] M. A. Rodriguez-Guerrero, A. Y. Jaen-Cuellar, R. D. Carranza-Lopez-Padilla, R. A. Osorio-Rios, G. Herrera-Ruiz, and R. D. J. Romero-Troncoso, “Hybrid approach based on GA and PSO for parameter estimation of a full power quality disturbance parameterized model,” IEEE Trans Ind. Informat., vol. 14, no. 3, pp. 1016–1028, Mar. 2018.
[10] Y. Zhang, S. Cheng, Y. Shi, D.-W. Gong, and X. Zhao, “Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm,” Expert Syst. Appl., vol. 137, pp. 46–58, Dec. 2019.
[11] A. O. de Sá, N. Nedjah, and L. de Macedo Mourelle, “Distributed efficient localization in swarm robotic systems using swarm intelligence algorithms,” Neurocomputing, vol. 172, pp. 322–336, Jan. 2016.
[12] S. Sharma, A. Shukla, and R. Tiwari, “Multi robot area exploration using nature inspired algorithm,” Biologically Inspired Cognit. Archit., vol. 18, pp. 80–94, Oct. 2016.

[13] E.-M. Wong, F. Bourgault, and T. Furukawa, “Multi-vehicle Bayesian search for multiple lost targets,” in Proc. IEEE Int. Conf. Robot. Autom., Barcelona, Spain, Apr. 2005, pp. 3169–3174.

[14] B. Wang, “Cooperative control of multi-agent systems with limited information mode,” Ph.D. dissertation, Dept. Automat., Shanghai Jiaotong Univ., Shanghai, China, 2011.

[15] T. Furukawa, F. Bourgault, B. Lavis, and H. F. Durrant-Whyte, “Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), Orlando, FL, USA, May 2006, pp. 2521–2526.

[16] S. Yu, “Decomposition and coverage of multi-UAV cooperative search area,” J. Beijing Univ. Aeronaut. Astronaut., vol. 41, no. 1, pp. 167–173, 2015.

[17] A. Khan, E. Yanmaz, and B. Rinner, “Information exchange and decision making in micro aerial vehicle networks for cooperative search,” IEEE Trans. Control Netw. Syst., vol. 2, no. 4, pp. 335–347, Dec. 2015.

[18] J. Y. Du, “Game theory based multi-UAV cooperative searching model and fast solution approach,” J. Shanghai Jiao Tong Univ., vol. 47, no. 4, pp. 667–678, 2013.

[19] G. Y. Lan, “Brief introduction of spatial methods to distribution patterns of population,” J. Northwest Forestry Univ., vol. 18, no. 2, pp. 17–21, 2003.

[20] H. T. Li, “Introduction to studies of the pattern of plant population,” Chin. Bull. Botany, vol. 12, no. 2, pp. 19–26, 1995.

[21] S. M. Fei, “Quantitative features of populations of Pinus tabulaeformis and P. armandii regenerated following water damage at Qinling Mountain,” Chin. J. Plant Ecol., vol. 32, no. 1, pp. 95–105, 2008.

[22] X. Zhang, H. Wang, B. Sun, W. Li, and R. Wang, “The Markov model of bean optimization algorithm and its convergence analysis,” Int. J. Comput. Intell. Syst., vol. 6, no. 4, pp. 609–615, Aug. 2013.

[23] X. M. Zhang, “A novel swarm optimisation algorithm based on a mixed-distribution model,” Appl. Sci., vol. 8, no. 4, pp. 1–14, 2018.

[24] X. Zhang and T. Feng, “Chaotic bean optimization algorithm,” Soft. Comput., vol. 22, no. 1, pp. 67–77, Aug. 2016.

[25] M. Dadgar, S. Jafari, and A. Hamzeh, “A PSO-based multi-robot cooperation method for target searching in unknown environments,” Neurocomputing, vol. 177, pp. 62–74, Feb. 2016.

[26] X. M. Zhang, “Research on a novel swarm intelligence algorithm inspired by beans dispersal,” Ph.D. dissertation, Dept. Automat., Univ. Sci. Technol. China, Hefei, China, 2011.

[27] M. S. Couceiro, P. A. Vargas, R. P. Rocha, and N. M. F. Ferreira, “Benchmark of swarm robotics distributed techniques in a search task,” Robot. Auto. Syst., vol. 62, no. 2, pp. 200–213, Feb. 2014.

**XIAOMING ZHANG** received the B.S. degree in mathematics from the Shandong University of Technology, China, in 2002, the M.S. degree in control science and engineering from the Graduate University of Chinese Academy of Sciences, China, in 2006, and the Ph.D. degree in control science and engineering from the University of Science and Technology of China, Hefei, China, in 2011.

From 2006 to 2016, he was an Associate Research Fellow with the Institute of Intelligent Machines, Chinese Academy of Sciences, China. Since 2016, he has been with Anhui Agricultural University (AAU), Hefei, where he is currently an Associate Professor with the School of Information and Computer Science. His research interests include evolutionary computation, swarm intelligence, and cooperating robots.

**MOHSIN ALI** received the B.E. degree in computer science from the University of Engineering and Technology, Lahore, Pakistan, in 2014. He is currently pursuing the master’s degree with the School of Information and Computer Science, Anhui Agricultural University, Hefei, China.

From 2014 to 2015, he was a Web Developer with Genius Enterprises, Gujranwala, Pakistan. From 2015 to 2016, he was a Python Developer in Automotion, Lahore, Pakistan. His research interests include evolutionary computation and swarm intelligence. In 2016, he received a scholarship from the Chinese Scholarship Council for his master’s degree in China.