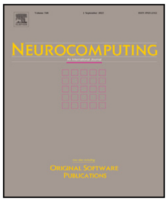
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



A hybrid optimization algorithm for multi-agent dynamic planning with guaranteed convergence in probability

Ye Zhang [a,b], Yutong Zhu [b], Haoyu Li [a,b,*], Jingyu Wang [b]

[a] *Research and Development Institute of Northwestern Polytechnical University in Shenzhen, Shen Zhen, 518057, China*
[b] *School of Astronautics, Northwestern Polytechnical University, Xi'an, 710072, China*

ARTICLE INFO

Communicated by H. Zhang

Keywords:
Trajectory planning
Optimization algorithm
Differential evolution
Convergence in probability

ABSTRACT

The paper aims to solve the problem of multi-agent path planning in complex environment using optimization algorithm. To address the issue of local optimum and premature convergence, a new method is proposed based on the whale optimization algorithm, combining the chaotic initialization, the reverse search and the differential evolution methods. It is theoretically proved that this algorithm is globally convergent in probability. When applied to path planning problems, the proposed optimization algorithm can effectively find a globally optimal and smoother path. Through simulation experiments with multi-UAVs, it is demonstrated that the proposed algorithm has better performance than the state-of-the-art methods in environment with both static and dynamic obstacles, reflecting the global convergence and robustness of the proposed algorithm.

1. Introduction

With the development of autonomy, unmanned aerial vehicles (UAVs), especially quadrotors, have been more and more involved in applications that cannot be easily accomplished by human beings [1,2], such as search and rescue, reconnaissance, resource exploration, forest fire prevention and military missions [3–5]. As one of the fundamental issues of achieving autonomy in UAVs, the subject of path planning seeks to establish an optimal and viable path in complex environments under certain safety constraints [6]. Compared with mobile robots, the path planning of aerial robots is more challenging due to its modeling in three-dimensional space and the risk of loss-of-control after collisions.

The robotics community has conducted substantial study on optimization algorithms to solve the issue of path planning and smoothness for UAVs in complex 3D environments [7]. Among them, meta-heuristic optimization algorithms are showing attractive features in many engineering applications [8–10]. Most heuristic algorithms are inspired by natural phenomena or physics laws in the nature. Considering the complexity of the search space and the stochastic nature of the optimization process, recent heuristic algorithms are proposed to improve the balance between exploration and exploitation, while preventing local optimum and low computational efficiency [11–13]. Santana et al. [14] developed a trajectory planner for a multi-robotic system based on Generic Algorithm (GA) through an approach that combines solutions for team orienteering and multiple backpack problems. Some studies reveal that the GA presents some limitations in its application to path planning—such as a lack of guarantee to obtain optimal solutions, difficulty in tuning parameters such as mutation rate and population size, and low speed of convergence [15]. Fan and Akhter [16] presented a Particle Swarm Optimization (PSO) for UAV trajectory planning using a time-varying inertia weight approach, obtaining good results. Song et al. [17] developed a new strategy to plan a smooth path for mobile robots through an improved PSO algorithm in combination with a continuous high degree Bezier curve. There are also studies on other artificial intelligence approaches, as in the work by Liu et al. [18], who designed a particle swarm optimization-trained fuzzy neural network algorithm to solve the path planning problem. An algorithm inspired by social-class pigeons was presented for solving multi-UAV path planning problem by [19], and a time stamp segmentation path planning model was developed to simplify the handling of multi-UAV coordination cost.

As one of the nature-inspired heuristic methods, the Whale Optimization Algorithm (WOA) was proposed and developed in recent years [20,21], which mimics the hunting behavior of humpback whales. Compared with other heuristic algorithms, WOA is found to have good optimization capabilities [22,23]. Due to its ease of implementation and small number of parameters [24,25], WOA has been successfully applied to various engineering optimization problems [26,27], such as parameter estimation of photovoltaic cells [28], radial distribution networks for optimal capacitors [29], wind speed prediction [30], spam profile detection [31] and so on. By utilizing WOA, premature convergence and local optima can be avoided, thus improving the convergence speed and computational accuracy [32].

[*] Corresponding author.
E-mail addresses: zhang_ye@nwpu.edu.cn (Y. Zhang), yutong.zhu@mail.nwpu.edu.cn (Y. Zhu), haoyuli@nwpu.edu.cn (H. Li).

https://doi.org/10.1016/j.neucom.2024.127764
Received 5 September 2023; Received in revised form 28 January 2024; Accepted 23 April 2024
Available online 27 April 2024
0925-2312/© 2024 Published by Elsevier B.V.

However, by exploring the optimization mechanism, it is found that the search agent of WOA can only approach the optimal solution in the second half of the iterative process. Besides, current methods [33] lack theoretical proof in global convergence so that its optimality cannot be fully guaranteed in a global scale. To address these shortcomings and to achieve good performance in dealing with the problem of global optimization, an improved algorithm based on WOA is proposed in this paper. The algorithm, which is abbreviated as **R**everse **S**earch **C**haos **D**ifferential-evolution **WOA** (RSCDWOA), is based on the reverse search (RS) and the differential evolution (DE) methods. Considering the problem of multi-agents path planning with unknown obstacles in 3D scenes [7], RSCDWOA is applied to prevent the whole path falling into local optimum. Compared with previous work on path planning problems, the main work and contributions of this paper are summarized as follows:

(i) It is theoretically proved that the RSCDWOA, algorithm is globally convergent with the addition of perturbation vectors, which demonstrates the robustness of the algorithm. It is the first time a convergence proof is given for the whale optimization algorithm and its extensions.

(ii) In simulation, the algorithm is combined with the classical Artificial Potential Field (APF) algorithm. In path planning, RSCDWOA Artificial Potential Field (RSCDWOA-APF) is shown to achieve global optimal and smooth trajectory. It is also compared with other algorithms to illustrate the superiority of the proposed RSCDWOA.

The rest of the paper is shown below. Section 2 introduces the problem to be solved in this paper based on a basic WOA method. In Section 3, the RSCDWOA algorithm is proposed. The global convergence of the proposed algorithm at different stages are mathematically proved in Section 4. In Section 5, several simulations are conducted to verify the global convergence of the new algorithm, and the superior performance of the proposed algorithm is demonstrated. The conclusion of this paper is summarized in Section 6.

2. Problem description and preliminaries

2.1. Problem description

As shown in Fig. 1, this paper considers the problem of multi-UAVs reaching a target in a complex environment containing dynamic obstacles. The path planning for multi-agent systems involves minimizing the movement of agents through intricate environments, ensuring they reach their destination step by step while avoiding collisions with both static and dynamic obstacles, as well as other robots [34]. This process aims to determine the optimal path with the least cost for each agent, and the mathematical calculation can be employed to determine the subsequent position of UAVs

$$\begin{aligned} x_i(t+1) &= x_i(t) + v_i \cos\theta_i \\ y_i(t+1) &= y_i(t) + v_i \sin\theta_i \end{aligned} \tag{1}$$

where $(x_i(t+1), y_i(t+1))$ denotes the subsequent position coordinates of UAV i, while $(x_i(t), y_i(t))$ represents its current position coordinates. The velocity and radial position of UAV i are denoted by v_i and θ_i, respectively. The optimization process relies on these kinematic characteristics of UAVs.

The following requirements should be satisfied. First, the planned paths should be globally optimal and avoid falling into a local optimum. Secondly, the maximum curvature of the path should be minimized considering the requirements of practical flight. Thirdly, the planned path should show some robustness in face of stochastically dynamic obstacles. Considering the flight characteristics of UAV and the safety in the flight process, the evaluation function of a flight path can be defined by calculating several costs regarding the aforementioned requirements [35].

Despite the strong optimization capability of WOA compared to other optimization algorithms, WOA ultimately creates only near-optimal solutions, which reduces the diversity of solutions and halts its exploration [36]. Therefore, the aim of the paper is to propose a new optimization algorithm that converges globally in probability to find optimal paths for multi-UAVs that satisfies the above requirements. To achieve this, the following assumptions are imposed.

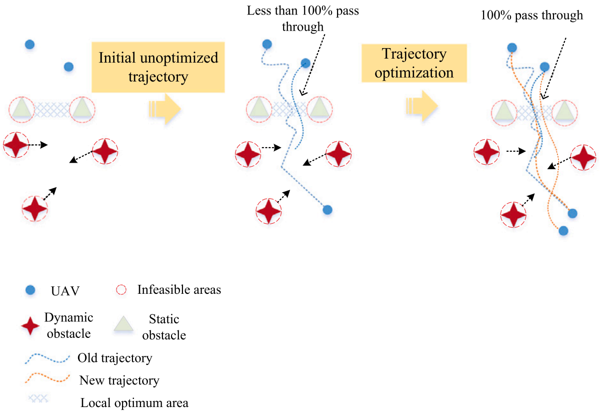


Fig. 1. Schematic diagram of path planning and optimization in complex environment.

Assumption 1. The mutual collisions between UAVs are not considered.

Assumption 2. The continuous solution space of infinite states maps to a finite discrete set.

Assumption 3. All UAV models are considered as particles.

2.2. The Whale Optimization Algorithm (WOA)

The WOA method is a bionic algorithm that simulates whales' hunting behaviors. In WOA, the hunting process is divided into two steps: encircling the victim and bubble-net attacking. The position of the encircled prey is modified based on the current optimal solution. In the initial phase, each search agent is randomly put in the search space, and the closest search agent to the target prey is picked as the best candidate solution. The remaining search agents then modify their positions by approaching the top candidate solution in succeeding cycles, as demonstrated in the following equations:

$$D_i(t) = |C \cdot X_*(t) - X_i(t)| \tag{2}$$

$$X_i(t+1) = X_*(t) - A \cdot D_i(t) \tag{3}$$

where t indicates the current iteration, $X_i(t)$ represents the current position of the search agent $i(i = 1, 2, \dots, n)$ at time t, and the best candidate solution is $X_*(t)$. $D_i(t)$ is the search agent's distance from i and $X_*(t)$ at time t. A and C are two vectors of coefficients that are calculated with Eqs. (4)–(5).

$$A = 2 \cdot a \cdot r - a \tag{4}$$

$$C = 2 \cdot r \tag{5}$$

$$a = 2 - \frac{2t}{t_{\max}} \tag{6}$$

During iterations, a drops from 2 to 0 in a linear fashion. r is a random number generated in $(0, 1)$, and $t_{\max}$ is the max number of iterations.

Positions are updated during bubble-net attacks using Eq. (7). b is a constant that affects the shape of the spiral, and it is set to 1. l is a random number generated in $(-1, 1)$.

$$X_i(t+1) = D_i(t) \cdot e^{bl} \cdot \cos(2\pi l) + X_*(t) \tag{7}$$

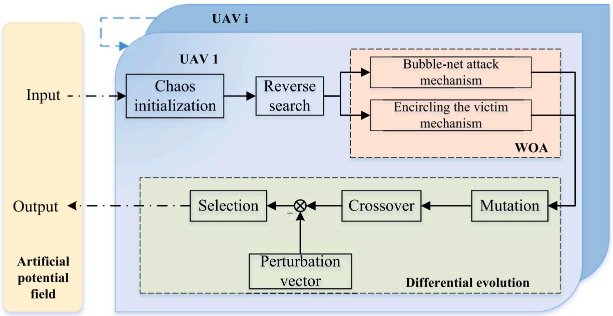


Fig. 2. Basic structure of the proposed RSCDWOA method in path planning for multi-UAVs.

Notably, two types of whale hunting are conducted simultaneously, i.e., each whale is in either the encircling or spiral phase. Consequently, a random number q is produced beforehand. We assume that there is a 50% probability to select two models to update the position of the whale during the optimization process [20]. If q is less than or equal to 0.5, the encircling mode is implemented. If not, the spiral mode is utilized.

$$X_i(t+1) = \begin{cases} X_*(t) - A \cdot D_i(t), & q < 0.5 \\ D_i(t) \cdot e^{bl} \cdot \cos(2\pi l) + X_*(t), & q \geq 0.5 \end{cases} \tag{8}$$

In the encircling mode, in addition to the location update mechanism described above, whales randomly hunt for the position of the prey, as shown in Eqs. (9)–(10). $X_{rand}(t)$ represents a randomly selected another whale from the population of whales at time t. The position update then depends on $X_{rand}(t)$ rather than $X_*(t)$.

$$X_i(t+1) = X_{rand}(t) - A \cdot D_i(t) \tag{9}$$

$$D_i(t) = |C \cdot X_{rand}(t) - X_i(t)| \tag{10}$$

The two ways of updating position in the enveloping mode are not chosen at random. When every value in A is more than 1 or less than −1, the random updating method is chosen and the population tends to adopt random exploration. When A is greater than −1 and less than 1, the optimal method for updating candidate solutions is selected.

In the WOA algorithm, a set of solutions are first initialized at random, and the positions of each search agent are updated in each iteration based on either the randomly selected search agent or the best solution so far found. Then the parameter a is reduced from 2 to 0 as the number of iterations increases in order to achieve exploration and exploitation. A random search agent is selected when $|A| > 1$, and the optimal option is picked when $|A| < 1$ to update the position of the search agent. The WOA can change between helical and circular motion depending on the value of q. Finally, The process is ended when the termination requirement is met.

3. An improved algorithm RSCDWOA

The original WOA converges slowly, and it is easily to fall into local optimum. Therefore, this section proposes an improved algorithm named the Reverse-search Chaos Differential-evolution WOA (RSCDWOA), which combines the chaos theory with differential evolution to increase the unpredictability of the search agent's position so that every corner of the search space can be explored. It will be later shown that the new trajectory found by RSCDWOA fully achieves the global optimum and passes smoothly through the region of local optimum (see Fig. 2).

3.1. The chaos theory

Chaos is a dynamic and deterministic system that is extremely sensitive to its initial conditions and parameters. The nature of chaos is random and unpredictable, while it also exhibits some regularity. It has been demonstrated that the chaos theory can aid meta-heuristics such as PSO in determining the ideal value of global coverage [37]. As shown in the previous subsection, the values of coefficients A and C are determined by a random vector $r \in [0, 1]$, while in RSCDWOA, r is substituted by a chaotic map $w(t)$.

$$A = 2 \cdot a \cdot w(t) - a \tag{11}$$

$$C = 2 \cdot w(t) \tag{12}$$

$w(t)$ is calculated by a sequence of chaotic map. Here, we will evaluate and choose between two chaotic maps [38], Eqs. (13) and (14), which correspond to logical maps and tent maps, respectively.

$$w(t+1) = \mu \cdot w(t) \cdot [1 - w(t)], \quad w(t) \in (0, 1) \tag{13}$$

$$w(t+1) = \begin{cases} \dfrac{w(t)}{\beta}, & w(t) < \beta \\ \dfrac{1 - w(t)}{1 - \beta}, & w(t) \geq \beta \end{cases} \tag{14}$$

The parameter l has a significant impact on the spiral shape of the whale search in WOA. In RSCDWOA, l is also substituted by $w(t)$ [37], which adds some randomness to the updating of each search agent i.

$$X_i(t+1) = D_i(t) \cdot e^{bw(t)} \cdot \cos[2\pi w(t)] + X_*(t) \tag{15}$$

Although the chaos theory is used to achieve more diverse parameters and solutions, it is important to note that X_* may not be updated a specified number of times during the search process, which makes X_* fall into local optimum. Therefore, this paper combines the reverse search methods and the differential evolution algorithm to improve search efficiency and avoid local optimum.

3.2. The reverse search method

In order to effectively increase the search efficiency of whale populations, the adoption of reverse search operation is explored. Refs. [39, 40] offer the concepts of secondary reverse point and secondary reflection reverse point, which substantially increase the efficiency of reverse search methods. The basic property of reverse point is given below.

Property 1. *Let $x \in [a, b]$, where $[a, b]$ is a real number interval, then the reverse point of x is $\bar{x} = a + b - x$.*

Property 2. *Let $x \in [a, b]$, where $[a, b]$ is a real number interval, and $\bar{x}$ is its reversal point, then the quadratic reversal point is defined as $\bar{x}_q = r_d(h, \bar{x})$, where $h = (a + b)/2$ is the center position of the interval $[a, b]$, and $r_d(h, \bar{x})$ represents a random number that obeys the normal distribution in the interval $[h, \bar{x}]$.*

Property 3. *Let $x \in [a, b]$, where $[a, b]$ is a real number interval, then the second reflection reverse point of x is defined as $\bar{x}_r = (x, h)$.*

A schematic diagram for reverse search is shown in Fig. 3. For an individual $X_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ in a certain whale group, where x_{ik} ranges from $[l_i, u_i]$, the reverse point can be expressed as $\bar{x}_{ik} = l_i + u_i - x_{ik}$, then its quadratic reverse point is $\bar{x}_{ik}(q) = r_d(h, \bar{x}_{ik})$, $h = (l_i + u_i)/2$, and the reverse point of secondary reflection is $\bar{x}_{ik}(r) = r_d(x_{ik}, h)$. Therefore, the reverse search operator is used to obtain its secondary reflection reverse point:

$$\bar{X}_i(r) = \left[\bar{x}_{i1}(r), \bar{x}_{i2}(r), \ldots, \bar{x}_{im}(r)\right] \tag{16}$$

If the fitness function $f(\bar{X}_r(r))$ satisfies $f(\bar{X}_i(r)) < f(X_i)$, then the replacement operation is performed, which is $X_i = \bar{X}_i$.

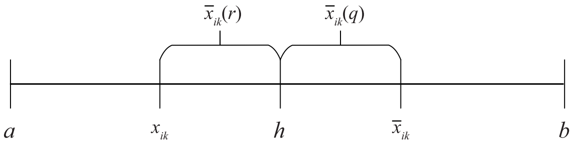


Fig. 3. A schematic diagram for reverse search.

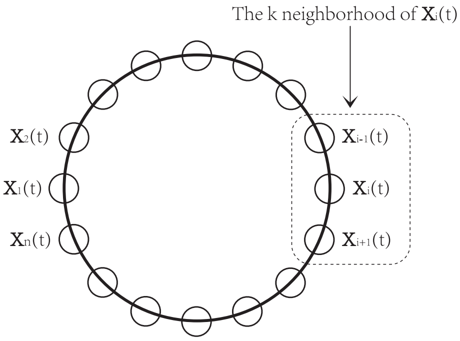


Fig. 4. The ring topology of k-neighborhood [42].

3.3. The differential evolution method

I. Mutation

The Differential evolution (DE) method utilizes both global and local search strategies [41,42]. Therefore, the final position pf each search agent is determined by the local and global search results. The mutant portion of DE is denoted by Eqs. (18)–(20). For any search agent i, its neighbors fall inside the interval $[i-k, i+k]$, where k is a nonzero integer in the range $[1, (n-1)/2]$. If k equals to 1, the neighborhood shown in Fig. 4 consists of vectors $X_{i-1}(t), X_i(t), X_{i+1}(t)$.

$X_{i,best\_l}$ is the place with the highest fitness value in close proximity to the search agent i. Two matrices, γ_1 and γ_2, are employed to increase the randomness of the search procedure, and they are both calculated as:

$$\gamma = \lambda \times rand(\Lambda, \psi) \tag{17}$$

where λ is a constant that has been set to 0.0001 in this paper. Λ represents the population size, and ψ is the dimension of each search agent. The local donor vector for each search agent i at time t, represented by $L_i(t)$, is calculated as follows:

$$L_i(t) = X_i(t) + \gamma_1 \cdot \left(X_{i,best\_l} - X_i(t)\right) + \gamma_2 \cdot \left(X_{p1}(t) - X_{p2}(t)\right) \tag{18}$$

where p_1 and p_2 are two random values chosen from $[i-k, i+k]$. $X_{i,best\_g}$ is the best solution for the current global whale population, so the global donor vector for each search agent i at time t, denoted by $G_i(t)$, is computed as:

$$\begin{aligned} G_i(t) = {} & X_{i,best\_g} + \mathcal{F} \cdot \left(X_{i,best\_g} - X_i(t)\right) \\ & + \mathcal{F} \cdot \left(X_{r1}(t) - X_{r2}(t)\right) \end{aligned} \tag{19}$$

where r_1 and r_2 are two random numbers chosen in the current whale population. $\mathcal{F}$ is a predefined scale factor. Combining $L_i(t)$ and $G_i(t)$ with Eq. (20) we have the final donor vector $V_i(t)$:

$$V_i(t) = \omega \cdot G_i(t) + (1-\omega) \cdot L_i(t) \tag{20}$$

where $\omega \in (0,1)$ is a predefined coefficient.

II. Crossover

In the crossover section, the diversity of the whale population is further improved after the donor vector is computed in the mutation section. $U_{i,j}(t)$ denotes the value of dimension j for each search agent i at time t, which is updated according to the following rules:

$$U_{i,j}(t) = \begin{cases} V_{i,j}(t), & rand(\Lambda, \psi) \geq C_{rate} \text{ or } \chi = j \\ X_{i,j}(t), & \text{otherwise} \end{cases} \tag{21}$$

where C_{rate} represents the crossover rate. Specifically, if a random number is greater than C_{rate}, $U_{i,j}(t)$ is updated by $V_{i,j}(t)$. In addition, if a random dimension χ is equal to j, $U_{i,j}(t)$ is also updated by $V_{i,j}(t)$. This is to avoid situations where the initial value of C_{rate} is too small, which would invalidate the cross component. In other cases, it is updated by $X_{i,j}(t)$.

After the crossover operator of the differential evolution process in RSCDWOA, a perturbation vector $X_{i,j}(t)$ corresponding to each trial vector $U_{i,j}(t)$ is added and the occurrence probability of each perturbation vector is denoted by α, which is a control parameter. The newly generated vector is denoted as $W_{i,j}(t)$ and φ is a random number between 0 and 1.

$$W_{i,j}(t) = \begin{cases} X_{i,j}(t), & \varphi \leq \alpha \\ U_{i,j}(t), & \text{otherwise} \end{cases} \tag{22}$$

III. Selection

During the selection process, the position with better fitness value is chosen as $X_i(t+1)$ of each search agent i.

$$X_i(t+1) = \begin{cases} W_i(t), & f\left(W_i(t)\right) \leq f\left(X_i(t)\right) \\ X_i(t), & f\left(X_i(t)\right) < f\left(W_i(t)\right) \end{cases} \tag{23}$$

Algorithm 1 presents the overall process for RSCDWOA. A flow chart in Fig. 5 shows the complete implementation process of the proposed algorithm.

Algorithm 1 RSCDWOA

```
Initialize the population of whales $X_i (i = 1, 2, \cdots, n)$
Initialize $C_{rate}$, $\mathcal{F}$, lifespan $\mathcal{S}$
$s \leftarrow 0$
step3.1: chaos theory
Calculate $w(t+1)$ with Eq. (13) or (14)
Calculate the fitness of each search agent
while $t \leq \hat{t}$ do
    for each search agent do
        Update $a$, $A$, $C$, $l$ and $q$
        if $q \leq 0.5$ then
            if $|A| < 1\&s < \mathcal{S}$ then
                Update position of current search agent with Eq. (3)
            else if $|A| \geq 1\&s < \mathcal{S}$ then
                Select a random search agent $X_{rand}$
                Update position of current search agent with Eq. (8)
            else
                step3.3: differential evolution
                Mutation
                Generate the final donor vector with Eq. (20)
                Crossover
                Generate $U_i(t)$ and $W_i(t)$ with Eq. (21) and (22)
                Selection(step3.2: reverse search)
                Update position of each search agent with Eq. (23) and
                (16)
            end if
        else if $q \geq 0.5$ then
            Update position of each search agent with Eq. (7)
        end if
        Check if any search agent goes beyond the search space and
        amend it
        Calculate the fitness value of each search agent
        Update $X_*$ if there is a better solution
    end for
    $++s$ if there is no better solution
    $++t$
end while
return
```

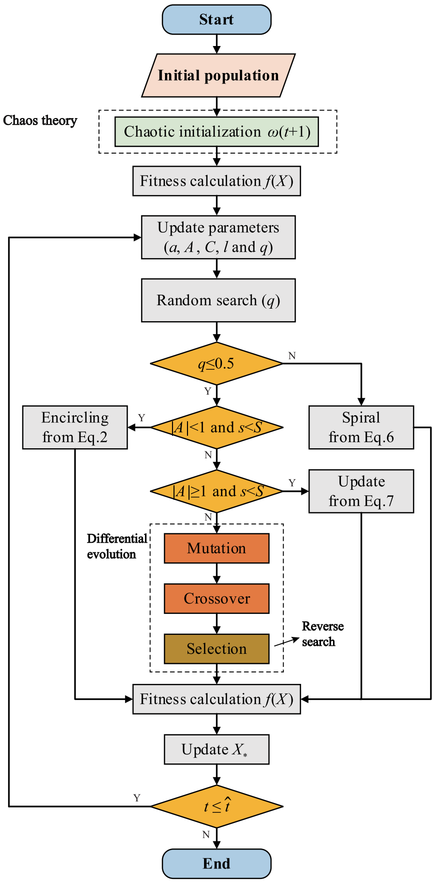


Fig. 5. The whole process of RSCDWOA.

4. Analysis of convergence

The continuous solution space of infinite states can be discretized to facilitate the theoretical analysis of the algorithm [43]. Now we define the state of individual i in at time t as $X_i(t) = X_i^t, X_i^t \in D$, where D is the state space, then the sequence $\{X_i^t, t \geq 1\}$ formed by state X_i^t is a random sequence with discrete values in the discrete space.

It is known that the current state of an individual is only related to the previous state and is independent of t. Besides, the number of populations and the number of states are finite in the discrete space, the sequence $\{X_i^t, t \geq 1\}$ is a finite homogeneous Markov chain. As the value of the optimal function is recorded at each iteration, it is also an absorbing Markov process. Let g^t be the locally optimal state of individual i at time t, B^* be the set of globally optimal solutions, and Ω be the set of locally optimal solutions. Then Ω is a subset of the state space D and every vector in Ω is equal yet not globally optimal.

Definition 1 (*Convergence in Probability [44]*). The sequence $\{X_i^t, t \geq 1\}$ converges to the global optimum according to probability when and only when:

$$\lim_{t\to\infty} p\left\{X_i^t \cap B^* \neq \phi\right\} = 1 \tag{24}$$

A random sequence $\{X_i^t, t \geq 1\}$ converges probabilistically, that is, the probability that the best individual sequence of the overall sequence $\{X_i^t, t \geq 1\}$ will converge to the global optimum as the iteration time t increases infinitely is close to 1. Non-convergence probabilistically means that the limit of the probability of convergence is less than 1, but this does not mean that the algorithm cannot converge to the global optimum. In fact, algorithms that do not converge probabilistically may have high convergence. However, probabilistically convergent algorithms are usually more robust than probabilistically non-convergent algorithms.

Due to the finite accuracy of numerical computation in computers, we can map the continuous state space D to a finite discrete set d. The state space of the stochastic process is defined as the overall space of the algorithm, and if the overall scale is m, the state space of the random sequence $\{X_i^t, t \geq 1\}$ is

$$D^m = \overbrace{d \times d \times \cdots \times d}^{m} \tag{25}$$

Let X, Y, Z, U, V be the states of D^m, $p\{X_i^{t+1} = Y | X_i^t = X\}$, indicating that at time t the state is X, at the next time the state changes to Y, and so on.

Let a_0, b_0, c_0 be the operators of the mutation, crossover and selection process in RSCDWOA. $p\{a_0(X) = Y\}$ is the probability that X mutates to Y at time t, and similarly to obtain $p\{b_0(X, Y) = Z\}$, $p\{c_0(X, Z) = V\}$. Hence,

$$\begin{aligned} p\left\{X_i^{t+1} = V | X_i^t = X\right\} = &\sum_{Y,Z\in D^m} p\left\{a_0(X) = Y\right\} \\ &\cdot p\left\{b_0(X, Y) = Z\right\} \cdot p\left\{c_0(X, Z) = V\right\} \end{aligned} \tag{26}$$

where $X, Y, Z, V \in D^m$.

Definition 2. The sequence $\{X_i^t, t \geq 1\}$ is the overall sequence of the algorithm. If for some integer $k \geq 0$, $p\{X^{t+k} = Y | X^t = X\} > 0$ indicates that state X is reachable to state Y. If state Y is also reachable to state X, X, Y is said to be reachable to each other.

Lemma 1. *$\exists X^{t_k} \in \Omega$ at time t_k, then*

$$\lim_{t\to\infty} p\left\{X_i^t = X_i^{t_k} | X_i^{t_k} \in \Omega\right\} = 1 \tag{27}$$

where $\Omega = \{X = (\overbrace{\vec{b}, \vec{b}, \ldots, \vec{b}}^{m}), \vec{b} \in D^m$ *and* $\vec{b} \in B^*\}$

Proof. When the state $X^{t_k} \in \Omega$ at time t_k, all vectors of the overall X^{t_k} are equal, so that the difference between any two vectors is 0. By the equations for the differential evolution process, we can obtain

$$\begin{aligned} &p\left\{a_0(X_i^{t_k}) = X_i^{t_k}\right\} = 1, \quad \forall X_i^{t_k} \in \Omega \\ &p\left\{b_0(X_i^{t_k}, X_i^{t_k}) = X_i^{t_k}\right\} = 1, \quad \forall X_i^{t_k} \in \Omega \\ &p\left\{c_0(X_i^{t_k}, X_i^{t_k}) = X_i^{t_k}\right\} = 1, \quad \forall X_i^{t_k} \in \Omega \end{aligned} \tag{28}$$

Thus according to formula (26), we can get

$$\begin{aligned} &p\left\{X_i^{t+1} = X | X_i^t = X \in \Omega\right\} \\ &= \sum_{Y,Z\in D^m} p\left\{a_0(X) = Y\right\} \cdot p\left\{b_0(X, Y) = Z\right\} \\ &\cdot p\left\{c_0(X, Z) = X\right\} \\ &= p\left\{a_0(X) = X\right\} \cdot p\left\{b_0(X, X) = X\right\} \\ &\cdot p\left\{c_0(X, X) = X\right\} \\ &= 1 \end{aligned} \tag{29}$$

where $X \in \Omega$ and $Y, Z \in D^m$

Therefore, for a given positive integer t_k, if the overall $X_i^{t_k} \in \Omega$, then

$$\lim_{t\to\infty} p\left\{X_i^t = X_i^{t_k} | X_i^{t_k} \in \Omega\right\} = 1 \tag{30}$$

Theorem 1. *If it contains only the bubble-net attack mechanism, RSCDWOA cannot converge in probability. That is*

$$\lim_{t\to\infty} p\left\{X_i^t \cap B^* \neq \phi\right\} < 1 \tag{31}$$

Proof. Let X^0 be a random vector of initial individuals uniformly distributed on D. Each individual in X^0 is generated with equal probability. Clearly $p\{X^0 \in \Omega\} > 0$.

According to the bubble-net attack mechanism iterative formula, we get

$$X_i^{t+1} = g^t + \left|g^t - X_i^t\right| \cdot e^{bw(t)} \cdot \cos[2\pi w(t)] \tag{32}$$

Thus its one-step transfer probability is:

$$\begin{aligned}
& p\left\{X_i^{t+1} = g^{t+1} | X_i^t = g^t\right\} \\
& = p\left\{X_i^{t+1} = g^t + \left|g^t - X_i^t\right| \cdot e^{bw(t)} \cdot \cos[2\pi w(t)]\, \middle| X_i^t = g^t\right\} \\
& = p\left\{X_i^{t+1} = g^t | X_i^t = g^t\right\} \\
& = \begin{cases} 1, & g^t = g^{t+1} \\ 0, & g^t \neq g^{t+1} \end{cases}
\end{aligned} \tag{33}$$

From Lemma 1, when $X^0 \in \Omega$, we get:

$$\lim_{t\to\infty} p\left\{X_i^t = X^0 | X^0 \in \Omega\right\} = 1 \tag{34}$$

Then,

$$\lim_{t\to\infty} p\left\{X_i^t = g^t | g^t \in \Omega\right\} = 1 \tag{35}$$

implying that,

$$\lim_{t\to\infty} p\left\{X_i^t \cap B^* = \phi | g^t \in \Omega\right\} = 1 \tag{36}$$

Hence, we can get:

$$\begin{aligned}
& \lim_{t\to\infty} p\left\{X_i^t \cap B^* \neq \phi\right\} \\
& = 1 - \lim_{t\to\infty} p\left\{X_i^t \cap B^* = \phi\right\} \\
& \leq 1 - \lim_{t\to\infty} p\left\{X_i^t \cap B^* = \phi, X^0 \in \Omega\right\} \\
& \leq 1 - \lim_{t\to\infty} p\left\{X_i^t \cap B^* = \phi | X^0 \in \Omega\right\} \cdot p\left\{X^0 \in \Omega\right\} \\
& = 1 - p\left\{X^0 \in \Omega\right\} \\
& < 1
\end{aligned} \tag{37}$$

Therefore,

$$\lim_{t\to\infty} p\left\{X_i^t \cap B^* \neq \phi\right\} < 1 \tag{38}$$

$$p\left\{X_i^{t+1} = g^{t+1} | X_i^t = g^t\right\} = 1 \tag{39}$$

According to Definition 1, the sequence $\{X_i^t, t \geq 1\}$ does not converge in probability, so Theorem 1 stands.

Theorem 2. *If it contains only the encircling victim mechanism, RSCDWOA can converge in probability, which is*

$$\lim_{t\to\infty} p\left\{X_i^t \cap B^* \neq \phi\right\} = 1 \tag{40}$$

Proof. According to the encircling victim mechanism iterative formula, we get

$$X_i^{t+1} = g^t - A \cdot \left|C \cdot g^t - X_i^t\right| \tag{41}$$

Thus its one-step transfer probability is

$$\begin{aligned}
& p\left\{X_i^{t+1} = g^{t+1} | X_i^t = g^t\right\} \\
& = p\left\{X_i^{t+1} = g^t - A \cdot |C \cdot g^t - X_i^t|\, | X_i^t = g^t\right\} \\
& = p\left\{X_i^{t+1} = g^t | X_i^t = g^t\right\} \\
& = \begin{cases} 1, & g^t = g^{t+1}, \quad A = 0 \text{ or } C = 1 \\ 0, & g^t \neq g^{t+1}, \quad A \neq 0 \text{ and } C \neq 1 \end{cases}
\end{aligned} \tag{42}$$

Assuming $A = 0$ or $C = 1$

$$\begin{cases} A = 2aw(t) - a = 0 \\ C = 2w(t) = 1 \end{cases} \tag{43}$$

we have

$$w(t) = \frac{1}{2} \tag{44}$$

Depending on the kind of chaotic mapping chosen, we divide the discussion into two cases, the former being logical map and the latter being tent map.

(1) when $w(t+1) = \mu \cdot w(t) \cdot [1 - w(t)]$

In this case,

$$w(t+1) = \frac{1}{4}\mu \tag{45}$$

if and only if $\mu = 2$

$$w(t+1) = \frac{1}{2} = w(t) \tag{46}$$

Therefore, $p\{X_i^{t+1} = g^{t+1} | X_i^t = g^t\} = 1$, trapped in local optimum.

(2) when $w(t+1) = \begin{cases} \dfrac{w(t)}{\beta}, & w(t) < \beta \\ \dfrac{1 - w(t)}{1 - \beta}, & w(t) \geq \beta \end{cases}$

where $0 < \beta < 1$, substitute $w(t) = 1/2$ to obtain:

$$w(t+1) = \begin{cases} \dfrac{1/2}{\beta}, & \dfrac{1}{2} < \beta \\ \dfrac{1/2}{1-\beta}, & \dfrac{1}{2} \geq \beta \end{cases} \tag{47}$$

Only when $\beta = 0$ or $\beta = 1$, we have:

$$w(t+1) = \frac{1}{2} = w(t) \tag{48}$$

and because $0 < \beta < 1$, so

$$w(t+1) \neq \frac{1}{2} \tag{49}$$

In summary, when selecting the tent map, we get $A \neq 0$ and $C \neq 1$, thus

$$p\left\{X_i^{t+1} = g^{t+1} | X_i^t = g^t\right\} = 0 \tag{50}$$

This means that if it only contains the encircling victim mechanism, RSCDWOA does not fall into the local optimum. Suppose the individual i is still not in the global optimal solution set B^* at time t, we have:

$$\begin{aligned}
& p\left\{X_i^t \cap B^* = \phi\right\} \\
& = p\left\{X_i^t \cap B^* = \phi | X_i^{t-1} \cap B^* \neq \phi\right\} \cdot p\left\{X_i^{t-1} \cap B^* \neq \phi\right\} \\
& + p\left\{X_i^t \cap B^* = \phi | X_i^{t-1} \cap B^* = \phi\right\} \cdot p\left\{X_i^{t-1} \cap B^* = \phi\right\}
\end{aligned} \tag{51}$$

Since the algorithm is a Markov process in the absorbing state, thus

$$p\left\{X_i^t \cap B^* = \phi | X_i^{t-1} \cap B^* \neq \phi\right\} = 0 \tag{52}$$

implying that Eq. (51) gives

$$\begin{aligned}
& p\left\{X_i^t \cap B^* = \phi\right\} \\
& = p\left\{X_i^t \cap B^* = \phi | X_i^{t-1} \cap B^* = \phi\right\} \cdot p\left\{X_i^{t-1} \cap B^* = \phi\right\}
\end{aligned} \tag{53}$$

According to Eq. (50), individual i will therefore enter the global optimal solution set with some probability, and thus

$$0 < p\left\{X_i^t \cap B^* \neq \phi | X_i^{t-1} \cap B^* = \phi\right\} < 1 \tag{54}$$

Eq. (53) gives

$$\begin{cases} p\left\{X_i^t \cap B^* = \phi\right\} \\ = \left[1 - p\left\{X_i^t \cap B^* \neq \phi | X_i^{t-1} \cap B^* = \phi\right\}\right] \\ \cdot p\left\{X_i^{t-1} \cap B^* = \phi\right\} \\ p\left\{X_i^{t-1} \cap B^* = \phi\right\} \\ = \left[1 - p\left\{X_i^{t-1} \cap B^* \neq \phi | X_i^{t-2} \cap B^* = \phi\right\}\right] \\ \cdot p\left\{X_i^{t-2} \cap B^* = \phi\right\} \\ p\left\{X_i^{t-2} \cap B^* = \phi\right\} \\ = \left[1 - p\left\{X_i^{t-2} \cap B^* \neq \phi | X_i^{t-3} \cap B^* = \phi\right\}\right] \\ \cdot p\left\{X_i^{t-3} \cap B^* = \phi\right\} \\ \dots \\ p\left\{X_i^1 \cap B^* = \phi\right\} \\ = \left[1 - p\left\{X_i^1 \cap B^* \neq \phi | X_i^0 \cap B^* = \phi\right\}\right] \\ \cdot p\left\{X_i^0 \cap B^* = \phi\right\} \end{cases} \tag{55}$$

Summarized as

$$p\left\{X_i^t \cap B^* = \phi\right\} \\ = \prod_{k=1}^{t} \left[1 - p\left\{X_i^k \cap B^* \neq \phi | X_i^{k-1} \cap B^* = \phi\right\}\right] \\ \cdot p\left\{X_i^0 \cap B^* = \phi\right\} \tag{56}$$

According to Eq. (54),

$$\lim_{t\to\infty} \prod_{k=1}^{t} \left[1 - p\left\{X_i^k \cap B^* \neq \phi | X_i^{k-1} \cap B^* = \phi\right\}\right] = 0 \tag{57}$$

Hence,

$$\lim_{t\to\infty} p\left\{X_i^t \cap B^* = \phi\right\} = 0 \tag{58}$$

which is equivalent to:

$$\lim_{t\to\infty} p\left\{X_i^t \cap B^* \neq \phi\right\} = 1 \tag{59}$$

According to Definition 1, Theorem 2 holds.

Now we have

$$W_{i,j}^t = \begin{cases} X_{i,j}^t, & \varphi \leq \alpha \\ U_{i,j}^t, & \text{otherwise} \end{cases} \tag{60}$$

After the crossover operator of the differential evolution process in RSCDWOA, a perturbation vector $X_{i,j}^t$ corresponding to each trial vector $U_{i,j}^t$ is added and the probability of occurrence of each perturbation vector is denoted by α, where α is a control parameter. The perturbation vector $X_{i,j}^t$ contains j sequences $\{X_i^t, t \geq 1\}$, for all $j = 1, 2, \dots, n$. The newly generated vector is denoted as $W_{i,j}^t$, the process is denoted as $\boldsymbol{M}$ and its operator is denoted as m_0. Thus, φ is a random number between 0 and 1. Due to the addition of the perturbation vector, the next process selection equation is updated as follows

$$X_i^{t+1} = \begin{cases} W_i^t, & f(W_i^t) \leq f(X_i^t) \\ X_i^t, & f(X_i^t) < f(W_i^t) \end{cases} \tag{61}$$

The process after the selection update is denoted as $\boldsymbol{N}$ and its operator is denoted as n_0. $f(X)$ denotes the minimum function value (fitness value) of the state (population) X.

Lemma 2. *After carrying out the* $\boldsymbol{M}$ *process, all states (populations) of its state space* D^m *are reachable to each other. For all states* $X, Y \in D$*, the probability of a one-step transfer from* X *to* Y *is greater than 0, thus*

$$p\left\{m_0 \cdot b_0 \cdot a_0(X) = Y\right\} > 0 \tag{62}$$

Proof. After the $\boldsymbol{M}$ process has been performed, three operators are used, namely the mutation operator a_0, the crossover operator b_0 and the $\boldsymbol{M}$ process operator m_0. Assuming that the $\boldsymbol{N}$ process has not yet been performed in D^m, for $\forall X, Y, Z \in D^m$, based on Eq. (60) we get

$$p\left\{m_0 \cdot b_0 \cdot a_0(X) = Y\right\} \\ = (1-\alpha) \cdot p\left\{b_0 \cdot a_0(X) = Y\right\} \\ + \alpha \cdot \sum_{Z \in D^m} p\left\{b_0 \cdot a_0(X) = Z\right\} \cdot p\left\{m_0(Z) = Y\right\} \tag{63}$$

where $X, Y, Z \in D^m$, and from Eq. (60), we can get

$$p\left\{m_0(Z) = Y\right\} > 0 \quad \text{for all } Z. \tag{64}$$

then

$$\alpha \cdot \sum_{Z \in D^m} p\left\{b_0 \cdot a_0(X) = Z\right\} \cdot p\left\{m_0(Z) = Y\right\} > 0 \tag{65}$$

Therefore, we can get

$$p\left\{m_0 \cdot b_0 \cdot a_0(X) = Y\right\} > 0 \tag{66}$$

Employing the $\boldsymbol{M}$ process, the probability of a one-step transfer from any state X to any state Y is greater than 0. Thus, all states in D^m are reachable to each other at this time. Next, properties of the $\boldsymbol{N}$ process operator will be proved.

Lemma 3. *Given states (populations)* $X, Y, Z \in D^m$*, where* $Z \subset X \cup Y$*, the* $\boldsymbol{N}$ *process has the following two properties:*

(i) *if* $f(Z) \neq \min\{f(X), f(Y)\}$*,* X, Y *cannot generate* Z *by the* $\boldsymbol{N}$ *process, thus* $p\{n_0(X, Y) = Z\} = 0$.

(ii) *if* $f(Z) = \min\{f(X), f(Y)\}$*,* X, Y *can generate* Z *by the* $\boldsymbol{N}$ *process, thus* $p\{n_0(X, Y) = Z\} > 0$.

Proof. From Eq. (61), this obviously holds.

Remark 1. It follows from the $\boldsymbol{M}$ process that the generated mutation vector $W_{i,j}^t$ is equal to either the perturbation vector $X_{i,j}^t$ with probability α or the trial vector $U_{i,j}^t$ with probability $1-\alpha$. When α is 0, $W_{i,j}^t$ is equal to $U_{i,j}^t$, implying that the $\boldsymbol{M}$ process has not occurred; when α takes a smaller number between 0 and 1, $U_{i,j}^t$ will be replaced by the perturbation vector with a smaller probability, and since the algorithm contains only bubble-net attack and encircling the victim mechanism is chosen with equal probability, this ensures that even if a body is caught in a local optimum in the bubble-net attack mechanism, it can quickly jump out of the local optimum in the encircling the victim mechanism by the increased perturbation vector. At the same time, the probability that $W_{i,j}^t$ equals $U_{i,j}^t$ is large, which allows the algorithm to maintain its original search capability. Therefore α is usually set to a smaller number between 0 and 1.

Remark 2. It is clear from the $\boldsymbol{N}$ process that, similar to the selection process before the update, the best individual (fitness value) is selected to the next iteration.

Remark 3. The convergence framework with respect to differential evolution can be represented by the elite retention strategy of the genetic algorithm [45,46], and since the global convergence of RSCDWOA depends on the differential evolution stage. Similar to mathematical induction, the proof of the global optimality of the algorithm in terms of probability is therefore required to show that the probability of it producing any individual for each generation of the population is greater than 0. This means that for any function and initialization, the probability of obtaining the optimum at each generation during evolution is always greater than 0. Furthermore, it is shown that the algorithm maintains the optimum solution for the next generation.

Theorem 3. *After adding the perturbation vector, RSCDWOA containing the bubble-net attack and the encircling the victim mechanism converges globally in probability, that is, the sequence* $\{X_i^t, t \geq 1\}$ *converges globally in probability.*

Proof. $\forall X, Y, Z \in D^m$, then the probability of transfer

$$\begin{aligned} & p\left\{X_i^{t+1}=Z|X_i^t=X\right\} \\ & =p\left\{n_0\cdot m_0\cdot b_0\cdot a_0(X)=Z\right\} \\ & =\sum_{Y\in D^m} p\left\{m_0\cdot b_0\cdot a_0(X)=Y\right\}\cdot p\left\{n_0(X,Y)=Z\right\} \end{aligned} \tag{67}$$

Let B_1 be a set consisting of populations in which at least one individual is optimal, That is $B_1 \in D^m$:

$$B_1=\left\{X=\left(\vec{x}_1,\vec{x}_2,\ldots,\vec{x}_m\right)\in D^m|\vec{x}_i\in B^*\right\}, \quad \exists i\in(1,2,\ldots,m) \tag{68}$$

For the sake of completeness of the proof, the transfer probabilities will then be illustrated in two cases.

(1) $X\in B_1, Z\in B_1$

In this case,

$$f(Z)=\min\left\{f(X),f(Y)\right\} \tag{69}$$

From Lemma 3, we can get

$$p\left\{n_0(X,Y)=Z\right\}>0 \tag{70}$$

According to Lemma 2, we have

$$p\left\{m_0\cdot b_0\cdot a_0(X)=Y\right\}>0, \quad \text{for all } X,Y\in D^m \tag{71}$$

Thus, based on Eq. (67)

$$p\left\{X_i^{t+1}=Z|X_i^t=X\right\}>0 \tag{72}$$

Similarly, we can obtain

$$p\left\{X_i^{t+1}=X|X_i^t=Z\right\}>0 \tag{73}$$

Therefore, the states X, Z are reachable to each other.

(2) $X\in B_1, Z\notin B_1$

In this case,

$$f(Z)\neq\min\left\{f(X),f(Y)\right\} \tag{74}$$

From Lemma 3, we can get

$$p\left\{n_0(X,Y)=Z\right\}=0 \tag{75}$$

based on Eq. (67), we can get

$$p\left\{X_i^{t+1}=Z|X_i^t=X\right\}=0 \tag{76}$$

Therefore, state X to state Z is not reachable. Similarly, when $X\notin B_1, Z\in B_1$, state Z to state X is not reachable. From the Eqs. (72) (73) (76), all elements in B_1 are reachable to each other. Thus, it follows that B_1 is a positive, irreducible, acyclic and closed set. $B_1\subset D^m$, B_1 is the set of point cuts. $D\backslash B_1$ is the set of cuts from D^m to B_1 and is an acyclic set of states. By the acyclic Markov chain property [47], it follows that for the sequence $\{X_i^t, t\geq 1\}$, from the steady process we have

$$\lim_{t\to\infty} p\left\{X_i^t=Y\right\}=\pi(Y), \quad Y\in B_1 \tag{77}$$

Thus

$$\lim_{t\to\infty} p\left\{X_i^t\in B_1\right\}=1 \tag{78}$$

Finally, we can get

$$\lim_{t\to\infty} p\left\{X_i^t\cap B^*\neq\phi\right\}=1 \tag{79}$$

Theorem 3 stands.

5. Simulation results

5.1. Basic setting

In order to verify the effectiveness and superiority of RSCDWOA, we have incorporated it in our simulations with the classical artificial potential field (APF) algorithm. The problem considered in the simulation experiment is multi-UAVs reaching a target in a environment with static and dynamic obstacles, and the requirements proposed in Section 2.1 need to be satisfied under previously mentioned assumptions. In a 100 km × 100 km × 20 km $x-y-z$ space, there are n UAVs between the starting point and the matching target. The minimum UAV speed is 50 meters per second, the minimum turning radius is 2 km, and the minimum flying altitude is 1 km. The minimal distance between UAVs for the multi-UAV path planning problem is $[50,50,20]^T$ m.

There are four UAVs and nine obstacles in the region. As shown in Table 1, their respective basic information is presented. Two scenarios were built to show that RSCDWOA solves the described problem. In the fully static scenario, which generally tends to produce locally optimal solutions between adjacent static obstacles, it is shown that RSCDWOA leads to globally optimal solutions for the trajectories. In mixed static and dynamic obstacle scenarios, it is verified that RSCDWOA improves the smoothness of the trajectory and outperforms other optimization algorithms. The basic settings of obstacle locations for the two scenarios are shown in Figs. 7(a) and 7(b), where the radius of influence is $X\in[4,18]$, $Y\in[5,14]$. In addition, a comparison is made between the proposed approach in this paper and the Ref. [48]. The implementation process of the simulation is displayed in Fig. 6, where APF is integrated with RSCDWOA in path planning.

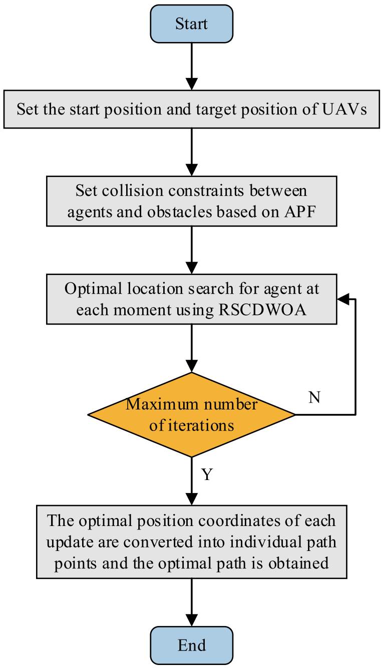

Start
Set the start position and target position of UAVs
Set collision constraints between agents and obstacles based on APF
Optimal location search for agent at each moment using RSCDWOA
Maximum number of iterations
N
Y
The optimal position coordinates of each update are converted into individual path points and the optimal path is obtained
End

Fig. 6. The combination of proposed RSCDWOA and UAV trajectory planning.

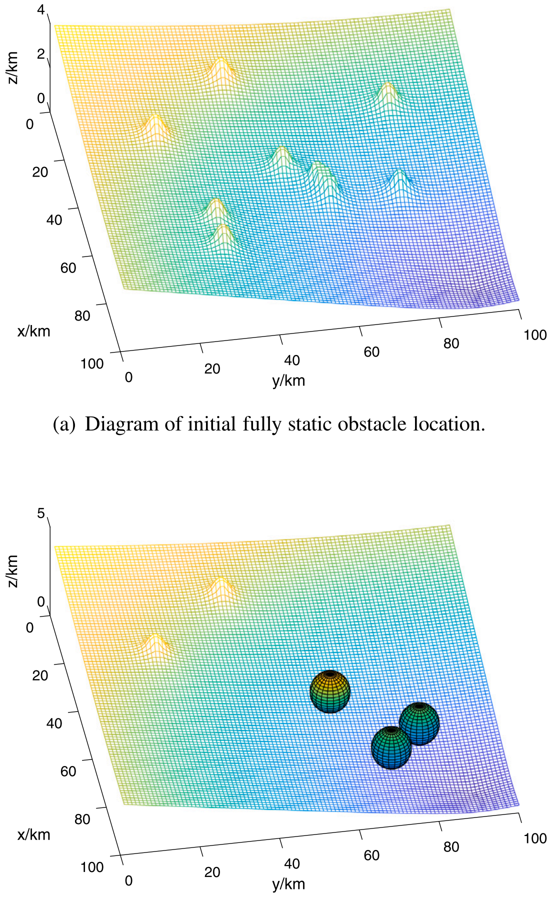
Table 1
Information of UAVs.

UAV No.	Start position/km	Target position/km
1	[20,10,5]	[90,90,0]
2	[10,20,5]	[90,90,0]
3	[15,20,5]	[90,90,0]
4	[20,20,5]	[90,90,0]

5.2. Selection of chaos map

Logistic map and tent map are the most commonly employed maps for chaotic behavior. In this paper, the chaotic value distribution and parameters of the two maps are shown and compared in Fig. 8.

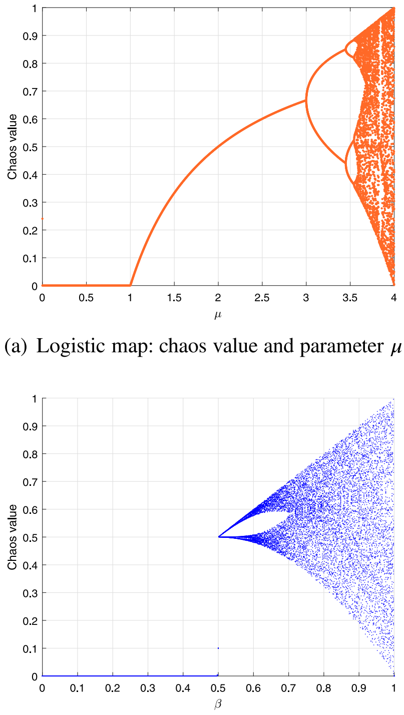
When $\mu=4$, the logistic map has a better distribution, and β begins to bifurcate at about $0.7-0.75$, therefore we choose to use $0.7-0.75$.



(a) Diagram of initial fully static obstacle location.

(b) Diagram of initial mixed static and dynamic obstacle location.

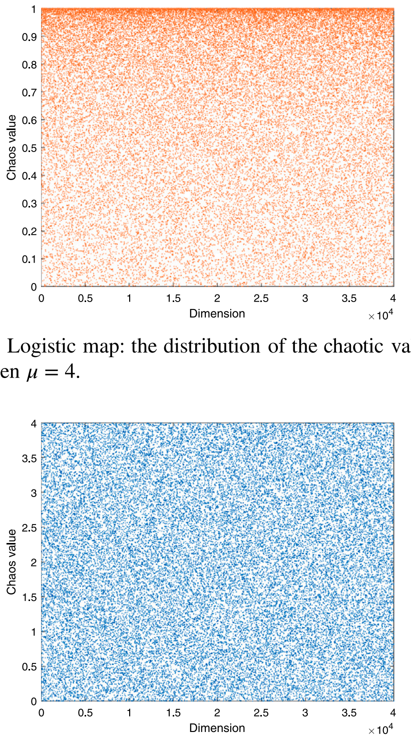
Fig. 7. Diagram of basic setting.



(a) Logistic map: chaos value and parameter μ.

(b) Tent map: chaos value and parameter β.

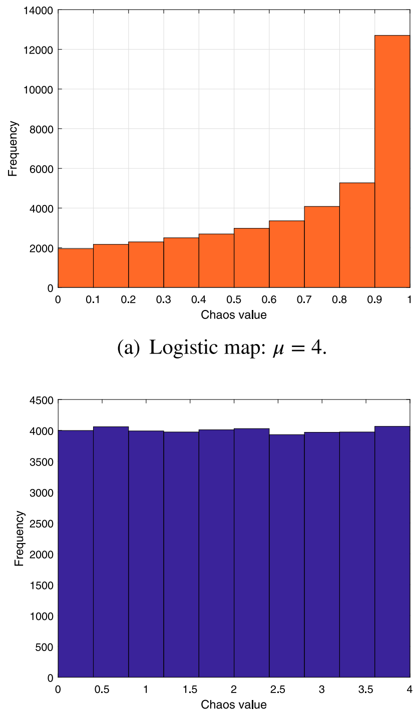
Fig. 8. The relationship between chaos value distribution and parameters.



(a) Logistic map: the distribution of the chaotic value when $\mu = 4$.

(b) Tent map: the distribution of the chaotic value when $\beta = 0.7 - 0.75$.

Fig. 9. The distribution of the chaotic value of the corresponding parameter in the dimension.



(a) Logistic map: $\mu = 4$.

(b) Tent map: $\beta = 0.7 - 0.75$.

Fig. 10. Histogram:frequency and chaos value.

The distribution of the chaotic value of the corresponding parameter in the dimension is shown in Fig. 9.

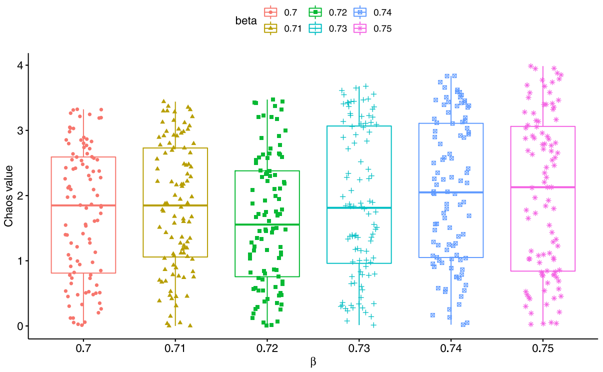


Fig. 11. Boxplot of chaos values and parameter β.

Table 2
Wilcoxon test.

β	0.70	0.71	0.72	0.73	0.74	0.75
P-value_1	0.0275	0.0043	0.0165	0.0067	0.0025	0.0193
P-value_2	0.0185	0.0092	0.0092	0.0037	0.0034	0.0279
P-value_3	0.0223	0.0012	0.0128	0.0025	0.0027	0.0124
P-value_4	0.0196	0.0121	0.0181	0.0098	0.0030	0.0243
Average	0.0220	0.0067	0.0142	0.0057	0.0029	0.0210

In order to show the selection strategy more clearly, we also draw a histogram as shown in Fig. 10 to present the law of its distribution. It is clearly shown that the logical map does not have a uniform distribution, and the frequency is greater near 1, which is consistent with the proof in Theorem 2. Next, we use the Wilcoxon test for the optimal choice of the parameter β of the logistic map between $0.7-0.75$.

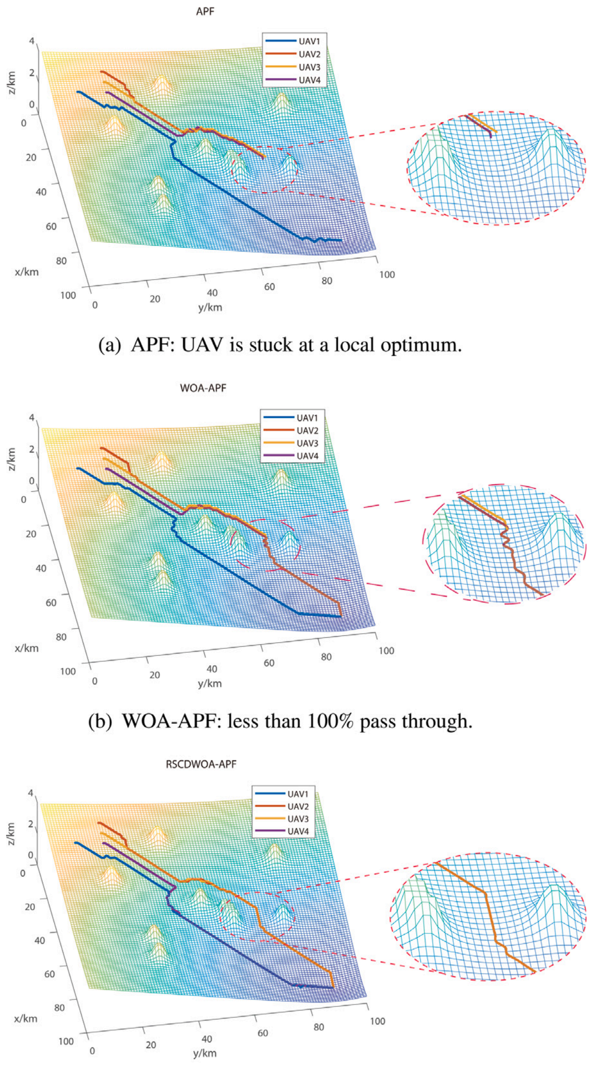
This paper selects six numbers uniformly between $0.7-0.75$ with an accuracy of 0.01. We then use hypothesis testing to determine whether there is a significant difference from a uniform distribution under this parameter. The range of the uniform distribution in this paper is $0-4$. In order to increase the reliability of the data inspection, it is necessary to perform a circular displacement, add 1 to each displacement, and divide it into 4 gears. Each parameter is inspected four times and the average value is taken. As shown in Fig. 11 and Table 2, it can be found that after the Wilcoxon test, the p-value of the six numbers are all less than 0.05, and the null hypothesis is rejected. Among them, the p-value of 0.74 is the smallest, so this paper finally selects $\beta = 0.74$.

5.3. Verification of static obstacle avoidance

When an obstacle is located between the current location and the target position, or when the areas of many obstacles form a concave polygon, even if the attraction gain of the target is increased in Refs. [48,49], it is still unable to jump to the local optimum or escape the obstacle. As depicted in Fig. 12(a), the UAV is stuck at a local optimum and its coordinates are $[39.34, 68.35, 2.6]^T$ km. Fig. 12(b) demonstrates that WOA can avoid local optima with some probability, but it does not pass 100% of the local optima and the path curvature is still high after optimization. After using RSCDWOA, the original path is smoothed and is able to pass 100% of the local optimum area. The good smoothing effect is shown in Fig. 12(c).

5.4. Verification of dynamic obstacle avoidance

Fig. 13 displays the outcomes of three dynamic obstacles from a numerical simulation used in the path planning process. The positional relationship between the UAVs and three dynamic obstacles is depicted in Figs. 13(a) and 13(b). Figs. 13(c), 13(d), and 13(e) show that the UAVs can avoid obstacles 1 and 2 sequentially as well as dynamic obstacle 3. As a result, as illustrated in Fig. 13(f), the desired location is reached at $t = 103$ s.



(a) APF: UAV is stuck at a local optimum.

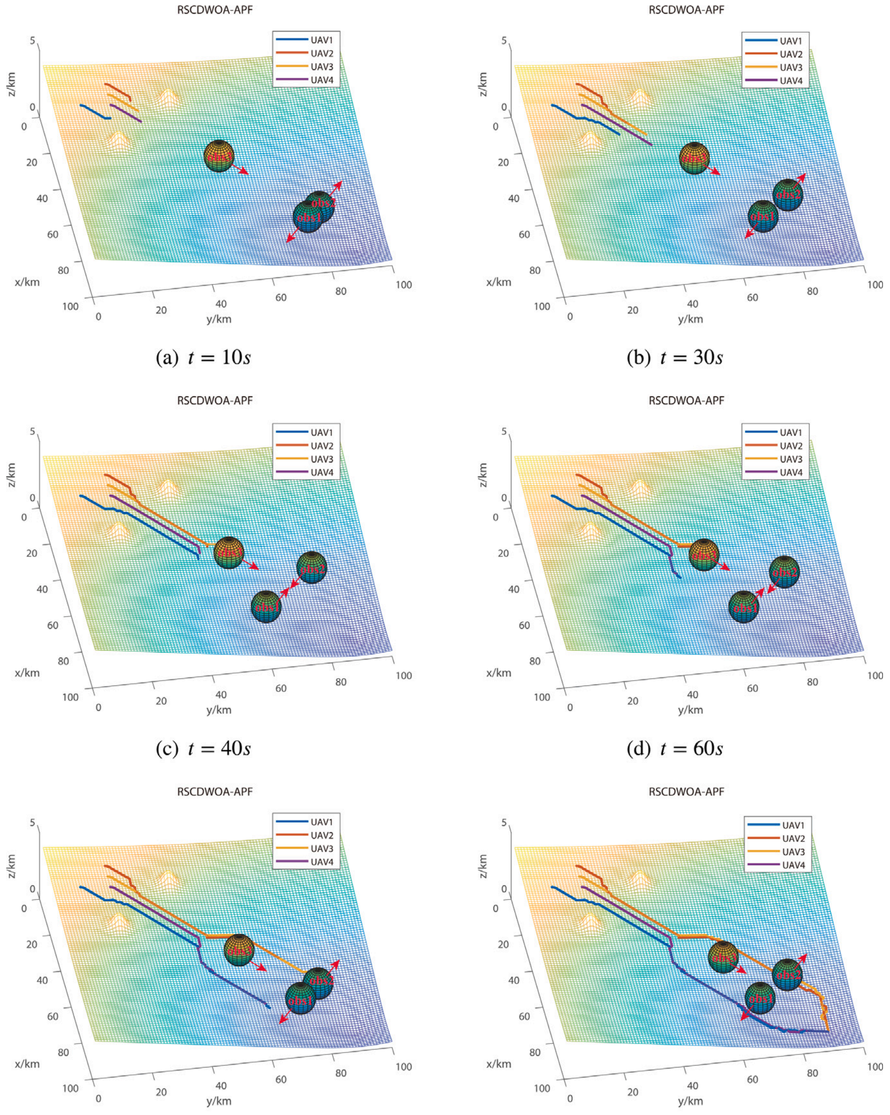
(b) WOA-APF: less than 100% pass through.

(c) RSCDWOA-APF: 100% pass through.

Fig. 12. Comparison diagram for trajectory optimization.

The left subfigure of Fig. 14(a) depicts the curvature of the full route process. Clearly, the most intense chattering happens when the path point is between 0 and 200. The right subfigure of Figs. 14(a) and 14(b) depict the path's curvature and curvature derivatives before and after smoothing equally distributed between 0 and 200 at each waypoint. After smoothing, it can be noted that the whole path's curvature does not exceed 2. This shows that the algorithm described in this paper is effective for trajectory smoothing.

As comparisons, particle swarm optimization (PSO) and genetic algorithm (GA) are used to fuse APF in Fig. 15 in this research. Fig. 16 depicts the evolution of the optimal fitness value in smooth path optimization with respect to various optimization strategies. It can be observed that, the fitness value is continuously updated during the smooth path optimization process. This demonstrates that these optimization algorithms may overcome common obstacle scenarios (such as local capture and premature convergence), thereby reaching the genuine optimal solution during the optimization process by jumping out of the substandard solution. After 40 rounds, the fitness value of every optimization algorithm convergence. It has been determined that RSCDWOA has the lowest fitness value, indicating that it has the highest adaptability.

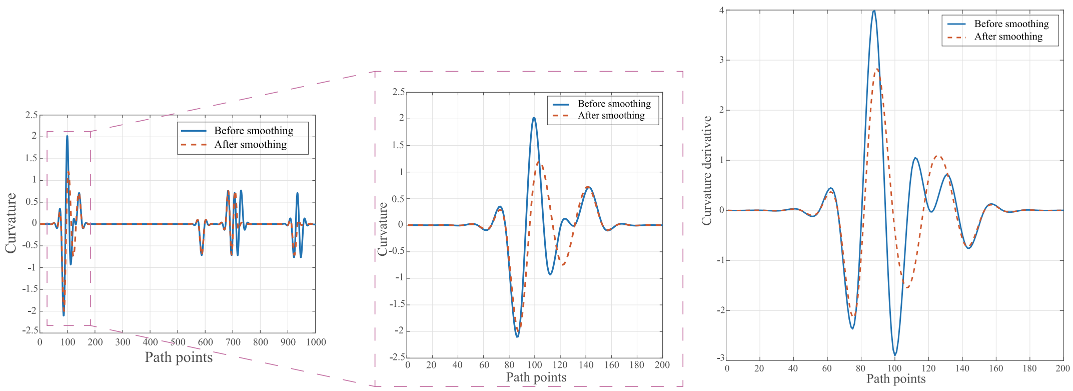


(a) $t = 10s$ (b) $t = 30s$

(c) $t = 40s$ (d) $t = 60s$

(e) $t = 80s$ (f) $t = 103s$

Fig. 13. Cases of avoiding multiple dynamic obstacles.

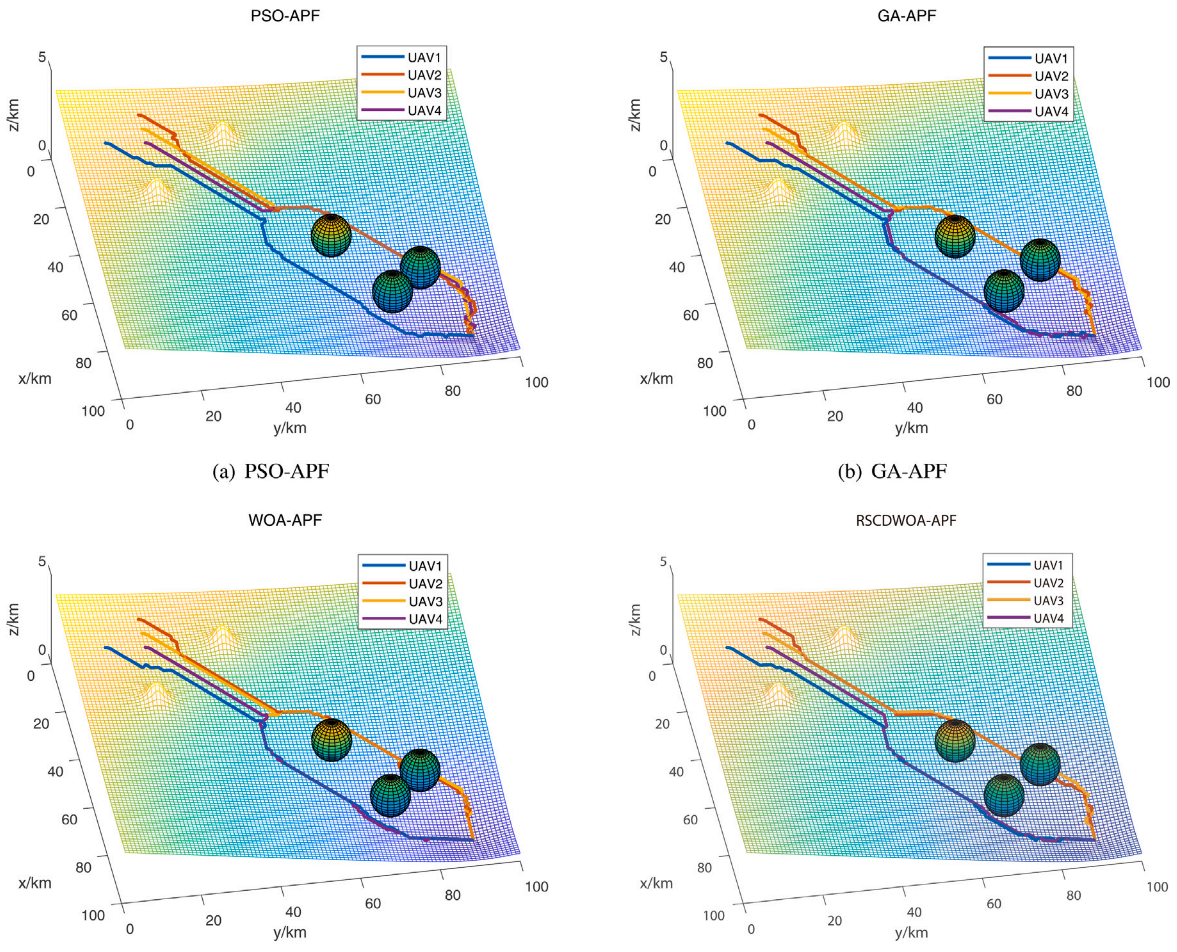


(a) The full route and curvature between 0 - 200 points.

(b) Curvature derivative between 0 - 200 points.

Fig. 14. Path curvature and curvature derivative before and after smoothing.

We compare the performance of different methods using the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE). The RMSE and MAE values of each UAV under different optimization algorithms are compared, and the better model is the one that yields



(a) PSO-APF

(b) GA-APF

(c) WOA-APF

(d) RSCDWOA-APF

Fig. 15. Dynamic obstacle avoidance based on APF fused with different optimization algorithms.

Table 3
Model evaluation indicators.

UAV No.	Optimization algorithm	MAE	RMSE
UAV1	PSO	0.1983	0.4724
	GA	0.1771	0.4449
	WOA	0.1139	0.3293
	RSCDWOA	**0.0538**	**0.1762**
UAV2	PSO	0.1952	0.4625
	GA	0.1734	0.4435
	WOA	0.1176	0.3391
	RSCDWOA	**0.0493**	**0.1724**
UAV3	PSO	0.1986	0.4765
	GA	0.1752	0.4438
	WOA	0.1093	0.3258
	RSCDWOA	**0.0576**	**0.1864**
UAV4	PSO	0.2027	0.4812
	GA	0.1725	0.4436
	WOA	0.1124	0.3279
	RSCDWOA	**0.0513**	**0.1742**

Table 4
Description of benchmark functions.

Function	Dim	Iteration	Equation
Sphere	30	100	$f_1(X) = \sum_{i=1}^{n} X_i^2$
Sumsquares	30	100	$f_2(X) = \sum_{i=1}^{n} iX_i^2$
Step	30	100	$f_3(X) = \sum_{i=1}^{n} [X_i + 0.5]^2$
Quartic	30	100	$f_4(X) = \sum_{i=1}^{n} iX_i^4 + \text{random}[0,1)$
Rosenbrock	30	100	$f_5(X) = \sum_{i=1}^{n-1} [100(X_{i+1} - X_i^2) + (X_i - 1)^2]$
Schwefel 2.22	30	100	$f_6(X) = \sum_{i=1}^{n} \lvert X_i \rvert + \prod_{i=1}^{n} \lvert X_i \rvert$
Rastrigin	30	100	$f_7(X) = \sum_{i=1}^{n} [X_i^2 - 10\cos(2\pi X_i) + 10]$
Griewank	30	100	$f_8(X) = \sum_{i=1}^{n} \frac{X_i^2}{4000} - \prod_{i=1}^{n} \cos(\frac{X_i}{\sqrt{i}}) + 1$
Ackley	30	100	$f_9(X) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} X_i^2}\right) -$ $\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi X_i)\right) + 20 + e$

the lowest RMSE and MAE values. It is shown that with RSCDWOA, the MAE and RMSE of each UAV are found to be the smallest in Table 3. Therefore, RSCDWOA has superior performance.

5.5. Statistical tests of RSCDWOA

To evaluate the performance of RSCDWOA comprehensively, a mixed static and dynamic obstacle scenario is created with different types of fitness functions, including unimodal functions ($f_1 - f_6$) and multimodal functions ($f_7 - f_9$) [33,42] in Table 4. In addition, in order to make a more comprehensive evaluation, two recently proposed optimization algorithms with the best performance WOASCALF [50] and SLAOS-WOA [51] based on the improvement of WOA are introduced. We use the nonparametric Wilcoxon rank-sum test to compare RSCDWOA with other well-known meta-heuristic algorithms on a pairwise basis. The results obtained based on the Wilcoxon rank-sum test have the following three signs '–', '∼' and '+'. Table 5 declares the results obtained from this test, the sign '–' indicates that RSCDWOA obtained worse results than other meta-heuristic algorithms. The sign '∼' indicates that the results obtained by these algorithms are the same

Table 5
Pair-wise comparison with RSCDWOA and all compared algorithms.

Function	RSCDWOA vs. PSO	RSCDWOA vs. GA	RSCDWOA vs. WOA	RSCDWOA vs. WOASCALF	RSCDWOA vs. SLAOS-WOA
f_1	+	+	+	~	~
f_2	+	+	−	−	~
f_3	~	+	~	+	−
f_4	−	+	~	~	+
f_5	~	+	~	+	+
f_6	+	~	+	~	~
f_7	+	+	+	+	~
f_8	+	+	+	~	+
f_9	+	+	+	+	+
Total of + sign	6	8	5	4	4

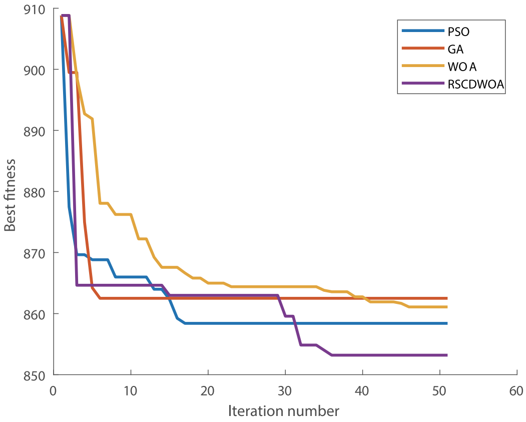


Fig. 16. Best fitness in PSO, GA, WOA and RSCDWOA.

Table 6
Comparison of results (Mean, Best, Std, p-value) for unimodal and multimodal benchmark functions.

Function	Metrics	WOA	WOASCALF	SLAOS-WOA	RSCDWOA
f_1	Mean	9.98E−01	1.67E+03	9.53E−03	**3.04E−04**
	Best	1.51E−01	9.18E+00	2.36E−03	**3.39E−05**
	Std	1.61E+00	5.27E−03	2.55E−02	**5.28E−04**
	p-value	3.02E−11	7.27E−09	9.33E−05	
f_2	Mean	5.82E−03	3.28E+00	4.38E−02	**5.26E−05**
	Best	3.35E−04	2.75E−07	3.69E−04	**4.92E−08**
	Std	7.92E−03	9.34E+00	6.53E−02	**7.46E−05**
	p-value	3.87E−08	5.03E−05	3.97E−09	
f_3	Mean	8.72E−01	2.07E+00	6.52E−03	**1.42E−03**
	Best	7.36E−02	**2.31E−04**	5.42E−04	9.13E−03
	Std	3.82E+00	8.31E+00	9.75E−03	**3.96E−03**
	p-value	3.55E−11	5.86E−08	7.31E−04	
f_4	Mean	3.17E−02	9.12E−01	2.98E−03	**2.53E−03**
	Best	7.37E−03	2.73E−01	**3.82E−05**	5.23E−05
	Std	6.36E−02	3.87E+00	6.34E−03	**4.05E−03**
	p-value	9.24E−14	3.75E−07	6.24E−11	
f_5	Mean	5.42E−01	3.51E+00	2.13E−02	**2.31E−03**
	Best	6.37E−03	2.21E−04	7.98E−04	**9.25E−05**
	Std	7.84E−01	7.52E+00	5.65E−02	**4.73E−03**
	p-value	9.20E−12	5.32E−07	2.79E−18	
f_6	Mean	3.24E−03	7.21E−01	4.54E−02	**1.56E−03**
	Best	5.76E−04	9.74E−02	8.35E−04	**9.96E−05**
	Std	2.12E−02	3.33E+00	**3.97E−03**	6.29E−02
	p-value	3.08E−04	5.62E−06	6.84E−11	
f_7	Mean	3.75E−02	6.81E−03	4.28E−02	**1.93E−04**
	Best	8.37E−04	7.36E−06	2.04E−04	**2.09E−07**
	Std	5.24E−02	9.17E−01	5.15E−02	**5.18E−04**
	p-value	4.38E−09	8.37E−11	1.12E−05	
f_8	Mean	4.32E−02	4.43E−04	2.91E−03	**7.26E−05**
	Best	8.76E−04	6.92E−06	7.85E−05	**2.21E−07**
	Std	7.73E−02	8.73E−04	8.51E−03	**9.97E−05**
	p-value	4.82E−04	8.63E−11	9.72E−21	
f_9	Mean	8.13E−02	3.58E−03	3.48E−03	**9.73E−04**
	Best	4.47E−03	8.56E−04	9.89E−05	**7.56E−05**
	Std	2.32E−01	6.86E−02	8.31E−03	**2.21E−03**
	p-value	3.29E−14	8.68E−08	6.94E−24	

Table 7
Comparison of average running time of each iteration for different algorithms.

Dimension	WOA	WOASCALF	SLAOS-WOA	RSCDWOA
20	0.0378	0.0228	**0.0215**	0.0326
30	0.0392	**0.0258**	0.0284	0.0306
50	0.1278	**0.0678**	0.0735	0.0839
100	0.4425	**0.2832**	0.2981	0.3196

and there is no difference. The sign '+' indicates that RSCDWOA gives better results than other algorithms.

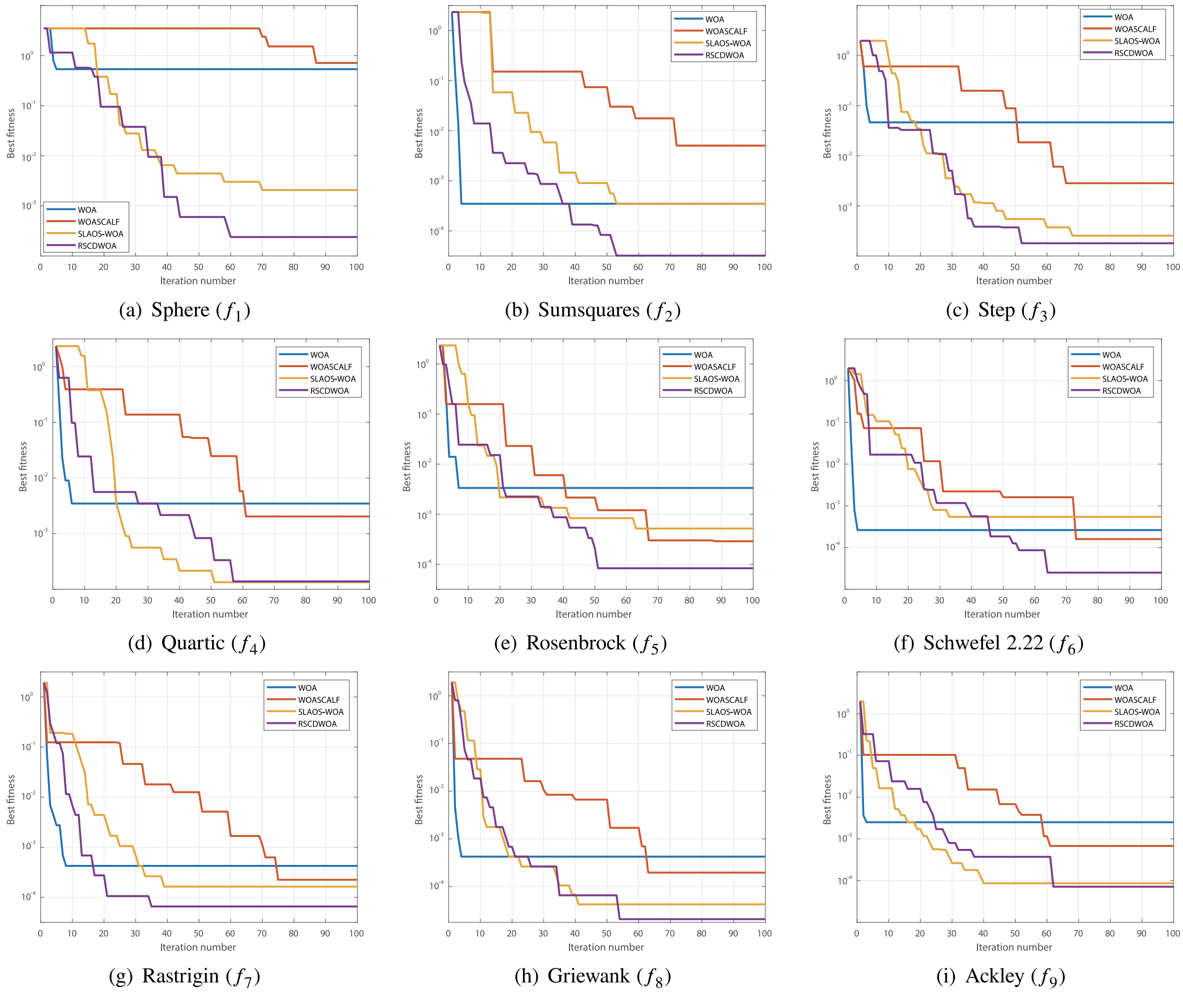
From the table we can conclude that the computational results of RSCDWOA are much better than PSO and GA. In comparison, RSCDWOA has comparable performance with WOA, WOASCALF and SLAOS-WOA. Therefore, this paper will continue to test the performance of RSCDWOA with the last three optimization algorithms in terms of other metrics. The number of iterations for each algorithm is 100 and the dimension is 30. Four evaluation metrics such as the mean value, the best one, the std and the p-value are calculated as shown in Table 6.

As can be seen from Table 6, most of the results of RSCDWOA outperformed other methods. The convergence curves of the four algorithms with respect to nine benchmark functions are shown in Fig. 17. Again, the results show that the RSCDWOA achieved better performance than other optimization algorithms.

In addition, to test the performance of the algorithms, we further listed the running time of RSCDWOA and other optimization algorithms. Table 7 shows the average running time required to perform each iteration. Each algorithm iterates 100 times in dimensions 20, 30, 50 and 100, respectively. Each experiment was repeated 10 times to obtain the average running time of each iteration as shown. As can be seen in Table 7, our method consumes a bit longer running time than other methods. The reason why the proposed method consumes more time is mainly due to the inclusion of perturbation vectors in differential evolution, which increases the computational cost while improving the robustness. Nevertheless, by introducing the reverse search strategy, large increase in computational cost is prevented. In conclusion, RSCDWOA largely improves its performance with a small increase in computational time.

6. Conclusion

In this paper, we aim to address the issue of local optimum in solving dynamic planning problems using optimization algorithms. By combining attractive characteristics of the chaotic initialization, the reverse search and the differential evolution methods, an improved algorithm based on the whale optimization algorithm is proposed. More importantly, it is demonstrated through concise theoretical proof that the proposed algorithm converges globally in probability after the addition of perturbation vectors, reflecting that the proposed algorithm is both globally convergent and robust. In the simulation experiment, the algorithm is integrated with the classical APF algorithm to solve the path planning problem of multi-UAV facing multiple static and



(a) Sphere (f_1)

(b) Sumsquares (f_2)

(c) Step (f_3)

(d) Quartic (f_4)

(e) Rosenbrock (f_5)

(f) Schwefel 2.22 (f_6)

(g) Rastrigin (f_7)

(h) Griewank (f_8)

(i) Ackley (f_9)

Fig. 17. Comparison of convergence curves of RSCDWOA and other four algorithms obtained in nine benchmark functions.

dynamic obstacles in a 3D scene. It is demonstrated through simulation results that the proposed optimization algorithm can solve the problem of local optimality in traditional APF algorithm and achieve the global optimal path planning. Meanwhile, the global convergence of the theoretical proof is verified. Through comparison with latest optimization algorithms, the proposed algorithm is shown to have better performance in multiple metrics and stronger path smoothing capability when applied to path planning problems. Future work will focus on the issue of collision avoidance among multiple agents, where the shape and dynamics of the agents will be considered and more impacting factors will be investigated.

CRediT authorship contribution statement

Ye Zhang: Writing – review & editing, Writing – original draft, Supervision, Funding acquisition, Formal analysis, Conceptualization. **Yutong Zhu:** Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Data curation, Conceptualization. **Haoyu Li:** Validation, Software, Funding acquisition, Formal analysis. **Jingyu Wang:** Validation, Supervision, Investigation, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

The work of the first author (Ye Zhang) was supported by the Guangdong Basic and Applied Basic Research Foundation under the Grant No. 2021A1515110569, and the National Natural Science Foundation of China under the Grant No. 52202502.

The work of the corresponding author (Haoyu Li) was supported by the Guangdong Basic and Applied Basic Research Foundation under the Grant No. 2021A1515110359, and the Natural Science Basic Research Program of Shaanxi under the Grant No. 2023-JC-QN-0677.

References

[1] P. Sujit, S. Saripalli, J.B. Sousa, Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless, IEEE Control Syst. Mag. 34 (1) (2014) 42–59.

[2] N.H. Motlagh, T. Taleb, O. Arouk, Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives, IEEE Internet Things J. 3 (6) (2016) 899–922.

[3] V. Roberge, M. Tarbouchi, G. Labonté, Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning, IEEE Trans. Ind. Inform. 9 (1) (2012) 132–141.

[4] X. Dong, B. Yu, Z. Shi, Y. Zhong, Time-varying formation control for unmanned aerial vehicles: Theories and applications, IEEE Trans. Control Syst. Technol. 23 (1) (2014) 340–348.

[5] S. Hayat, E. Yanmaz, R. Muzaffar, Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint, IEEE Commun. Surv. Tutor. 18 (4) (2016) 2624–2661.

[6] Y.K. Hwang, N. Ahuja, et al., A potential field approach to path planning, IEEE Trans. Robot. Autom. 8 (1) (1992) 23–32.

[7] O. Montiel, U. Orozco-Rosas, R. Sepúlveda, Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles, Expert Syst. Appl. 42 (12) (2015) 5177–5191.

[8] X.-S. Yang, Metaheuristic optimization: Nature-inspired algorithms and applications, in: Artificial Intelligence, Evolutionary Computing and Metaheuristics: in the Footsteps of Alan Turing, Springer, 2013, pp. 405–420.

[9] A. Sotoudeh-Anvari, A. Hafezalkotob, A bibliography of metaheuristics-review from 2009 to 2015, Int. J. Knowl. Intell. Eng. Syst. 22 (1) (2018) 83–95.

[10] P. Wang, Y. Zhou, Q. Luo, C. Han, Y. Niu, M. Lei, Complex-valued encoding metaheuristic optimization algorithm: A comprehensive survey, Neurocomputing 407 (2020) 313–342.

[11] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.

[12] S.Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, I. Aljarah, Grasshopper optimization algorithm for multi-objective optimization problems, Appl. Intell. 48 (2018) 805–820.

[13] X.-B. Meng, X.Z. Gao, L. Lu, Y. Liu, H. Zhang, A new bio-inspired optimisation algorithm: Bird swarm algorithm, J. Exp. Theor. Artif. Intell. 28 (4) (2016) 673–687.

[14] K.A. Santana, V.P. Pinto, D.A. Souza, Multi-robots trajectory planning using a novel GA, in: Information Technology and Systems: Proceedings of ICITS 2020, Springer, 2020, pp. 353–363.

[15] B. Patle, A. Pandey, D. Parhi, A. Jagadeesh, et al., A review: On path planning strategies for navigation of mobile robot, Defence Technol. 15 (4) (2019) 582–606.

[16] G.M. Nayeem, M. Fan, Y. Akhter, A time-varying adaptive inertia weight based modified PSO algorithm for UAV path planning, in: 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques, ICREST, IEEE, 2021, pp. 573–576.

[17] B. Song, Z. Wang, L. Zou, An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve, Appl. Soft Comput. 100 (2021) 106960.

[18] X.-h. Liu, D. Zhang, J. Zhang, T. Zhang, H. Zhu, A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm, Cluster Comput. (2021) 1–15.

[19] D. Zhang, H. Duan, Social-class pigeon-inspired optimization and time stamp segmentation for multi-UAV cooperative path planning, Neurocomputing 313 (2018) 229–246.

[20] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.

[21] M.M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, Neurocomputing 260 (2017) 302–312.

[22] F.S. Gharehchopogh, H. Gholizadeh, A comprehensive survey: Whale optimization algorithm and its applications, Swarm Evol. Comput. 48 (2019) 1–24.

[23] N. Zeng, D. Song, H. Li, Y. You, Y. Liu, F.E. Alsaadi, A competitive mechanism integrated multi-objective whale optimization algorithm with differential evolution, Neurocomputing 432 (2021) 170–182.

[24] Y. Ling, Y. Zhou, Q. Luo, Lévy flight trajectory-based whale optimization algorithm for global optimization, IEEE Access 5 (2017) 6168–6186.

[25] C. Kumar, R.S. Rao, A novel global MPP tracking of photovoltaic system based on whale optimization algorithm, Int. J. Renew. Energy Dev. 5 (3) (2016).

[26] Y. Dong, H. Zhang, C. Wang, X. Zhou, Wind power forecasting based on stacking ensemble model, decomposition and intelligent optimization algorithm, Neurocomputing 462 (2021) 169–184.

[27] H. Ding, X. Gu, Hybrid of human learning optimization algorithm and particle swarm optimization algorithm with scheduling strategies for the flexible job-shop scheduling problem, Neurocomputing 414 (2020) 313–332.

[28] D. Oliva, M. Abd El Aziz, A.E. Hassanien, Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm, Appl. Energy 200 (2017) 141–154.

[29] D.B. Prakash, C. Lakshminarayana, Optimal siting of capacitors in radial distribution network using whale optimization algorithm, Alexandria Eng. J. 56 (4) (2017) 499–509.

[30] J. Wang, P. Du, T. Niu, W. Yang, A novel hybrid system based on a new proposed algorithm—Multi-objective whale optimization algorithm for wind speed forecasting, Appl. Energy 208 (2017) 344–360.

[31] A.-Z. Ala'M, H. Faris, J. Alqatawna, M.A. Hassonah, Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts, Knowl.-Based Syst. 153 (2018) 91–104.

[32] Z. Yan, J. Zhang, J. Zeng, J. Tang, Three-dimensional path planning for autonomous underwater vehicles based on a whale optimization algorithm, Ocean Eng. 250 (2022) 111070.

[33] J. Bi, W. Gu, H. Yuan, Hybrid whale optimization algorithm with differential evolution and chaotic map operations, in: 2021 IEEE International Conference on Networking, Sensing and Control, ICNSC, Vol. 1, IEEE, 2021, pp. 1–6.

[34] J. Tan, L. Zhao, Y. Wang, Y. Zhang, L. Li, The 3D path planning based on A* algorithm and artificial potential field for the rotary-wing flying robot, in: 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC, Vol. 2, IEEE, 2016, pp. 551–556.

[35] C. Huang, X. Zhou, X. Ran, J. Wang, H. Chen, W. Deng, Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning, Eng. Appl. Artif. Intell. 121 (2023) 105942.

[36] H. Yuan, J. Bi, W. Tan, M. Zhou, B.H. Li, J. Li, TTSA: An effective scheduling approach for delay bounded tasks in hybrid clouds, IEEE Trans. Cybern. 47 (11) (2016) 3658–3668.

[37] L.-Y. Chuang, C.-H. Yang, J.-C. Li, Chaotic maps based on binary particle swarm optimization for feature selection, Appl. Soft Comput. 11 (1) (2011) 239–248.

[38] A.H. Abdullah, R. Enayatifar, M. Lee, A hybrid genetic algorithm and chaotic function model for image encryption, AEU-Int. J. Electron. Commun. 66 (10) (2012) 806–816.

[39] H.R. Tizhoosh, Opposition-based reinforcement learning, J. Adv. Comput. Intell. Intell. Inform. 10 (3) (2006).

[40] M. Ergezer, D. Simon, D. Du, Oppositional biogeography-based optimization, in: 2009 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2009, pp. 1009–1014.

[41] V. Tirronen, F. Neri, T. Karkkainen, K. Majava, T. Rossi, A memetic differential evolution in filter design for defect detection in paper production, in: Applications of Evolutionary Computing: EvoWorkshops 2007: EvoCoMnet, EvoFIN, EvoIASP, EvoINTERACTION, EvoMUSART, EvoSTOC and EvoTransLog. Proceedings, Springer, 2007, pp. 320–329.

[42] J. Luo, B. Shi, A hybrid whale optimization algorithm based on modified differential evolution for global optimization problems, Appl. Intell. 49 (2019) 1982–2000.

[43] H. Zhongbo, The theoretical researches and algorithmic design of convergent differential evolution algorithm in probability, 2014.

[44] R.M. Burton, Pointwise properties of convergence in probability, Statist. Probab. Lett. 3 (6) (1985) 315–316.

[45] J. Suzuki, A Markov chain analysis on simple genetic algorithms, IEEE Trans. Syst. Man Cybern. 25 (4) (1995) 655–659.

[46] G. Rudolph, Convergence analysis of canonical genetic algorithms, IEEE Trans. Neural Netw. 5 (1) (1994) 96–101.

[47] S.M. Ross, Stochastic Processes, John Wiley & Sons, 1995.

[48] Y.-b. Chen, G.-c. Luo, Y.-s. Mei, J.-q. Yu, X.-l. Su, UAV path planning using artificial potential field method updated by optimal control theory, Internat. J. Systems Sci. 47 (6) (2016) 1407–1420.

[49] U. Orozco-Rosas, O. Montiel, R. Sepúlveda, Mobile robot path planning using membrane evolutionary artificial potential field, Appl. Soft Comput. 77 (2019) 236–251.

[50] A. Seyyedabbasi, WOASCALF: A new hybrid whale optimization algorithm based on sine cosine algorithm and levy flight to solve global optimization problems, Adv. Eng. Softw. 173 (2022) 103272.

[51] Y. Huang, Y. Li, Z. Zhang, Q. Sun, A novel path planning approach for AUV based on improved whale optimization algorithm using segment learning and adaptive operator selection, Ocean Eng. 280 (2023) 114591.

Ye Zhang received her Ph.D degree in Aerospace engineering and automation from Delft University of Technology, the Netherlands, in 2019. She is currently an associate professor with the School of Astronautics, Northwestern Polytechnical University, Xi'an, China. His research interests include intelligent methods on autonomous decision-making, and self-recovery fault-tolerant control.

Yutong Zhu received double B.Eng. degree in Electrical and Electronic Engineering from Nanjing Normal University, Nanjing, China, in 2022 and from Northumbria University, Newcastle upon Tyne, UK, in 2022. He is currently pursuing the M.Eng. degree with the School of Astronautics, Northwestern Polytechnical University, Xi'an, China. His research interests focus on path planning, optimization algorithm, and multi-agent systems, and differential games.

Haoyu Li received his doctoral degree at the year 2019 from the Mathematics and IT Department of École Centrale de Lyon, France. Then he continued his research as a post-doctor at Northwestern Polytechnical University, Xi'an, China. Currently, he works as an associate professor at the Astronautics School of Northwestern Polytechnical University. His current research interests include stochastic signal processing, Markov jump systems, system identification, control and guidance technologies of aircraft.

Jingyu Wang received the Ph.D. degree in signal, image and automatic from the Université Paris-Est, Paris, France, in 2015. He is currently a Full Professor with the School of Astronautics and the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China.

His research interests include information processing, computer vision and intelligent perception.