Heliyon 9 (2023) e14784

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

A parallel particle swarm optimization and enhanced sparrow search algorithm for unmanned aerial vehicle path planning

Ziwei Wang [a,b], Guangkai Sun [a,b,*], Kangpeng Zhou [a,b], Lianqing Zhu [a,b]

[a] Beijing Engineering Research Center of Optoelectronic Information and Instruments, Beijing Information Science and Technology University, Beijing 100192, People's Republic of China

[b] Bionic and Intelligent Equipment Lab, Beijing Information Science and Technology University, Beijing 100192, People's Republic of China

ARTICLE INFO

Keywords:
Enhanced sparrow search algorithm
Particle swarm optimization
Path planning
Unmanned aerial vehicle

ABSTRACT

Unmanned Aerial Vehicle (UAV) path planning is to plan an optimal path for its flight in a specific environment. But it cannot get satisfactory results using ordinary algorithms. To solve this problem, a hybrid algorithm is proposed named as PESSA, where particle swarm optimization (PSO) and an enhanced sparrow search algorithm (ESSA) work in parallel. In the ESSA, the random jump of the producer's position is strengthened to guarantee the global search ability. Each scrounger keeps learning from the vintage experience of the producers. For the best-positioned sparrow, when it perceives the threat, the difference between the best individual and the worst individual will be imposed to speed up the search process. The elite reverse search strategy was added to yields the optimum diversity. In this paper, the performance of the PESSA algorithm is verified by 10 basic functions, and it can find the optimal value on the 7 test functions. Compared with the other 12 algorithms, PESSA's average value always ranks first. Finally, the proposed PESSA is applied in 4 different scenarios including two groups of 2D environments and two groups of 3D environments. In 2D environments, the average optimization results can reach 0.0165 and 0.0521 in two cases respectively. In 3D environments, the average optimization results can reach 0.6635 and 0.5349 in two cases respectively. The results show that the PESSA algorithm can acquire more feasible and effective route than compared algorithms.

1. Introduction

Unmanned Aerial Vehicle (UAV) has traditionally been used in military operations for a number of years, it's an unmanned aircraft operated by radio remote and programmed control device. Recently, UAV has attracted great interest for their potential applications in civilian areas such as mountain rescue [1], power inspection [2] and aerial photography. However, there are a number of technical challenges when using UAV for civilian domains. Particularly, UAV path planning refers to the process of finding the best flight path for an UAV to travel from an initial location to a desired destination while taking into account specific constraint conditions. This is a complex optimization problem, and various methods have been proposed to solve it. The classical approaches include A* algorithm [3], Dijkstra algorithm [4], Artificial Potential Field [5] and the Voronoi diagram searching method [6], etc. Recently, the heuristic approaches such as meta-heuristics just like differential evolution algorithm (DE) [7], time stamp segmentation algorithm (TSS) [8],

* Corresponding author. Beijing Engineering Research Center of Optoelectronic Information and Instruments, Beijing Information Science and Technology University, Beijing 100192, People's Republic of China.
E-mail address: guangkai.sun@buaa.edu.cn (G. Sun).

https://doi.org/10.1016/j.heliyon.2023.e14784
Received 9 July 2022; Received in revised form 10 March 2023; Accepted 16 March 2023
Available online 11 April 2023

particle swarm optimization (PSO) [9] and sparrow search algorithm (SSA) [10], etc have been widely used in path planning.

On the classical approaches, Zhao et al. presented a RETRBG algorithm, which is a trajectory replanning method based on guided paths, it can improve the speed and stability of path planning to a certain extent [11]. Men et al. introduced the CPP method based on a Region Optimal Decomposition (ROD), which achieved good results when applied to the path planning of UAV in a port environment [12]. Mangeruga et al. proposed a new diving planning method based on the original underwater pathfinding algorithm, which considered safety constraints and maximizes the number of points of interest visited [13]. The 3D UAV path planning framework was proposed by Koch et al. [14], the framework takes into account the semantic attributes of the environment and user-specified airspace constraints. They designed detailed and completed small-scale 3D reconstruction. The Voronoi diagram divides the map into several small pieces that contain only one threatening convex polygon. Epstein's k-best algorithm is used by many scholars to find the best path [15]. However, due to the complex motion constraints of drones, it is somewhat difficult to use in practice [16]. The performance constraints of the UAV also need to be considered when designing the flight path of the UAV. To be more suitable for the actual situation, the movement constraints of the UAV were considered by Zhu et al. In order to realize the path planning of the UAV, they proposed a sparse A* search (SAS) algorithm [17]. Because all kinds of path constraints are considered in the planning process, the SAS algorithm can prune the search space effectively. To improve the durability of USV, a feasible path planning method for USV based on ocean current data considering energy consumption was proposed [18], this method combines three commonly used path planning methods and energy consumption function so that USVs can avoid obstacles with minimal energy. Zammit et al. implemented a heterogeneous segmentation rate optical A*, a variable velocity assured standard RRT, an unsecured RRT, and a heterogeneous multiplexed RRT (MRRT), with comparable efficacy [19]. To solve the problem of efficiency and flyable path planning, Han et al. [20] established the GO-APP algorithm and GO-LBPP algorithm. Xu et al. [21] proposed a ring self-organizing mapping (RSOM) algorithm, which can perform flight path planning for multiple UAVs based on forest fire hazard maps.

On the heuristic approaches, in order to perfectly find the optimal solution of the UAV's global path planning problem, Zhang et al. proposed an improved algorithm based on the DE algorithm to obtain the feasible path [22]. A multi-UAV path planning model based on TSS algorithm was put forward by Zhang et al. [23], which simplified the processing of multi-UAV coordination costs by using the common time base. After that, they proposed a new social Pigeon inspired optimization (SCPIO) algorithm as the optimal search solver for the TSS model. Yu et al. proposed an adaptive selective variation constrained differential evolution algorithm for UAV path planning in disaster scenarios [24]. Liu et al. presented a three-dimensional path planning algorithm based on an adaptive sensitivity decision operator combined with PSO technique [9]. This algorithm which constructed an adaptive sensitivity decision region and overcame the shortcomings of local optimization and slow convergence. They combined the BAS algorithm with three non-trivial mechanisms, including quick local search, ant colony algorithm initial path generation, and search information orientation. Phung et al. proposed an SPSO algorithm based on spherical vectors [25], the algorithm can plan a safe and feasible path in a complex environment with multiple threats. Dewangan et al. used the GWO algorithm to solve the path planning problem of 3-D UAV [26], which provided the direction for the subsequent readers. Huang et al. proposed an algorithm based on improved quantum particle swarm optimization (QPSO) [27], which can plan a UAV path more quickly and accurately. Inspired by the PSO algorithm, an algorithm called delay-tolerant network was proposed by Sanchez-Garcia et al. [28]. Inspired by the FPA algorithm, a flower pollination algorithm based on neighborhood global learning was proposed by Chen et al. [29]. However, the simulation is only carried out on the two-dimensional plane, without three-dimensional verification. Machmudah et al. [30] studied the problem of using bending mechanism to optimize flight trajectory of fixed wing UAV at fixed altitude. They used PSO, GA and GWO algorithms to generate variable velocity trajectory.

Generally speaking, the classical approaches and heuristic approaches can be used to solve two-dimensional path planning in some simple environments. However, when it comes to three-dimensional path planning in complex environment, it obviously has some disadvantages such as slow calculation speed, low accuracy and falling into local optimum. So, it is necessary to study the path planning algorithm.

The structure of this paper is as follows. The related work is introduced in section 2. Extracting DEM map data, simulating obstacles, and modeling the UAV path planning environment are the contents of section 3. In section 4, a new algorithm PESSA is proposed, and PSO algorithm and SSA algorithm are briefly introduced. In section 5, the performance of PESSA is verified by 10 basic functions and compared with the other 12 algorithms. In section 6, the objective function is established considering the UAV's own constraints and environmental constraints, the new algorithm is applied to the environment model are proposed in Section 3. The performance of the new algorithm is comprehensively verified through different environments, and the effectiveness of the new algorithm is proved by comparing with other algorithms. The last part is the summary of the thesis.

2. Related work

In recent years, researchers have turned their attention to the technology of algorithm hybridization, which is the combination of two or more metaheuristic optimization algorithms [[31,32]]. In this technology, a good variety of algorithms can show more effective optimization when dealing with practical problems [33]. When solving path planning problem, particle swarm optimization algorithm tends to converge prematurely and fall into local optimal. To solve these problems, Han et al. [34] combined fitness adaptive differential evolution algorithm (FiADE) with particle swarm optimization algorithm (PSO) and improved it. Das et al. [35] proposed a new hybrid algorithm, which is a new algorithm that combines the improved particle swarm optimization algorithm and the improved gravitational search algorithm, which can plan the optimal trajectory suitable for the operation of multiple robots in the clustering environment. He et al. proposed a new algorithm called HSGWO-MSOS for UAV path planning [36], the HSGWO-MSOS algorithm is combined the simplified grey wolf optimization algorithm (SGWO) with the improved symbiotic biological search algorithm (MSOS).

Wang et al. proposed an improved algorithm GEDGWO [37], which combined the basic GWO and GED strategy, and applied it to solve multi-UAV and multi-urban tracking path planning problem of multi-UAV and multi-target. A new algorithm that mixes the sparrow search algorithm and the bionic neural network was used by Liu et al. to plan the UAV path [38], SSA algorithm is mainly used for global planning, and BINN is used for local planning.

In the new algorithms of the last few years, SSA can provides superior results compared to other search precision algorithms. However, SSA like most other intelligent optimization algorithms, easily falls into premature and local optimal algorithms and cannot find the optimal path in 3D UAV path planning. In order to improve the performance of SSA algorithm in UAV path planning, we proposed a new method named ESSA (enhanced sparrow search algorithm) and PSO algorithm is a classical evolutionary algorithm, it is popular because of its simple calculation process, so this paper also uses this algorithm for hybrid operation and proposed a new hybrid method named PESSA (parallel particle swarm optimization and enhanced sparrow search algorithm). PESSA algorithm is run in parallel with PSO algorithm and ESSA algorithm, and the better one is the final result. The ESSA algorithm is an enhanced version of the SSA algorithm. We have mainly made the following adjustments. First, for producers, we change the random jump of their position to a random move, which increases the stability of the search range and the ability of global search. Second, we only retain their ability to follow producers and plunder food for scroungers. Finally, for the sparrows who perceive the danger, when it is in the current optimal position, we let the sparrows randomly jump to a random position between the optimal global position and the worst position and make them walk randomly to get close to other sparrows. In PESSA, we also introduce the elite reverse search strategy to improve the ability of the algorithm to jump out of the local optimum and yield the optimum diversity.

The main work of this paper is summarized as follows.

- This paper presents an improved algorithm ESSA based on SSA algorithm (Enhanced sparrow search algorithm).
- This paper proposes a parallel ESSA algorithm and an improved PSO algorithm PESSA, and carries out elite reverse search based on it, which improves the ability of the algorithm to jump out of local optimization.
- The PESSA algorithm is applied to UAV path planning problem and compared with two basic algorithms and better experimental results are obtained.

The path planning problem is illustrated in Fig. 1. It mainly includes two parts: route planning model and route planning algorithm. The route planning model includes internal and external constraints. The internal constraints contain performance constraints of UAV such as flight altitude, maximum and minimum turning angle and so on. The external constraints include terrain and threats such as geographical range, terrain conditions, threat locations, threat levels, and so on. The objective function of this paper is through the performance constraints of UAV and external flight environment constraints, which consider all the necessary factors that may affect the path quality and their impact on the path performance. Finally, the algorithm combines the objective function to get the way-points, and connects them with smooth curves to form the flight path of the UAV.

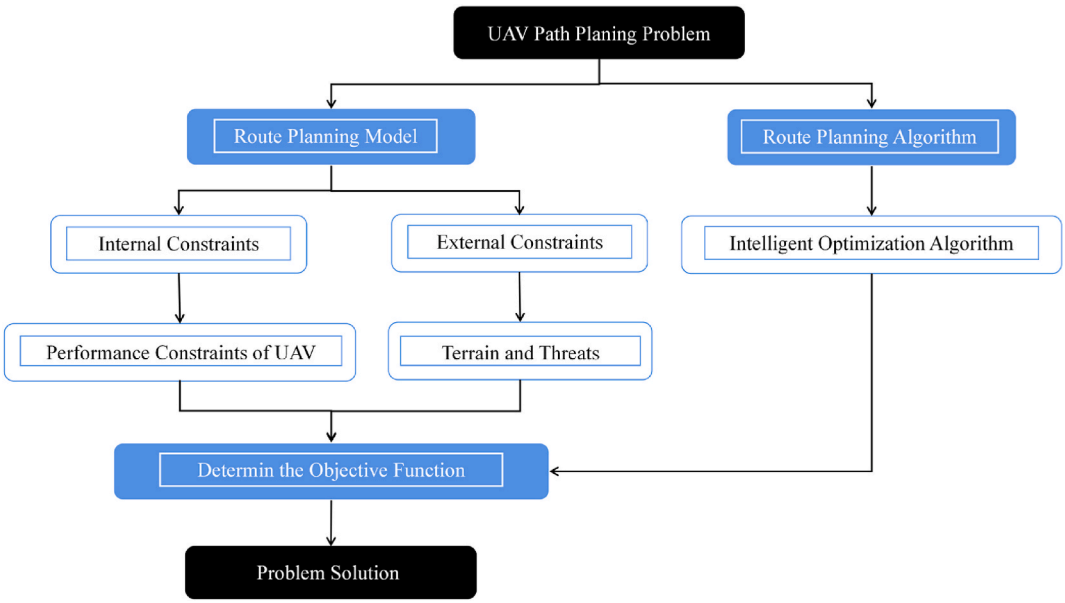

Fig. 1. UAV path planning problem.

3. Mathematical model in UAV path planning

3.1. Environment model

In this section, DEM data of the actual scene are extracted, the corresponding inspection terrain environment is established, the environment is combined with regular grid model. The obstacle is simulated by a double layer cylinder and the three-dimensional environment diagram is established. Finally, parameterization technology is used to describe the path and form a complete environmental model.

To initiate path planning, it is essential to translate the physical environment into a comprehensible format that aligns with the path planning algorithm [39]. This format is intricately tied to a search algorithm, as certain algorithms exhibit optimal performance when paired with a particular environment representation. Literature [40] outlines the performance of different representations used by different algorithms. In general, the modeling of topographic map needs to obtain the relevant Digital Elevation Model (DEM) information first. DEM is a digital simulation of terrain represented digitally, it is a branch of Digital Terrain Model (DTM). DTM includes a large amount of data such as slope, aspect and slope change rate, which contains a huge amount of information data and requires very complex information processing. DEM only contains a set of ordered values corresponding to the elevation, which has the advantages of small memory and easy reading, and is suitable for the airborne map information of small UAV.

There are various forms of data organization and expression of DEM model, three models are mainly used in engineering application. The first model is the contour model, a contour line is actually a simple arc with elevation values and corresponding coordinate points, but it can only represent part of the elevation values, generally not used to represent terrain models directly. The second model is the TIN (Triangulated Irregular Network) model composed of all original data. This data model has high accuracy, but its data volume is huge, and this form of expression is only adopted when the measurement scene is large. The third model is the regular network model, which has multiple regular grid units, and the regular grid model has a relatively small amount of data and can be compressed, so it has become one of the most widely used DEM models.

In this paper, a two-dimensional regular grid model with square shape is used to decompose the ground, The size of the two-dimensional grid is 100 × 100, X and Y are the horizontal and vertical coordinates, in which each element of matrix represents the elevation of terrain, and the sample of partial elevation data is shown in Fig. 2. Combined with the regular grid model, the corresponding patrol terrain environment is established in Matlab, as shown in Fig. 3.

In this paper, a double-layer cylinder is used for obstacles, the inner cylinder represents the absolute danger zone and the cylinder model is defined as the threat region with coordinate vector $danger\ zones_i\ [x_i, y_i, d_i, h_i, l_i]$. The detailed schematic diagram of the threat area is shown in Fig. 4, and its definition is saved in a separate matrix, as shown in Formula (1):

$$danger\ zones = \begin{bmatrix} x_1, y_1, h_1, d_1, l_1 \\ x_2, y_2, h_2, d_2, l_2 \\ \cdots \\ x_n, y_n, h_n, d_n, l_n \end{bmatrix} \tag{1}$$

O	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0. 405	0. 72	0. 945	1. 08	1. 125	1. 08	0. 945	0. 72	0. 405
3	0	0. 72	1. 28	1. 68	1. 92	2	1. 92	1. 68	1. 28	0. 72
4	0	0. 945	1. 68	2. 205	2. 52	2. 625	2. 52	2. 205	1. 68	0. 945
5	0	1. 08	1. 92	2. 52	2. 88	3	2. 88	2. 52	1. 92	1. 08
6	0	1. 125	2	2. 625	3	3. 125	3	2. 625	2	1. 125
7	0	1. 08	1. 92	2. 52	2. 88	3	2. 88	2. 52	1. 92	1. 08
8	0	0. 945	1. 68	2. 205	2. 52	2. 625	2. 52	2. 205	1. 68	0. 945
9	0	0. 72	1. 28	1. 68	1. 92	2	1. 92	1. 68	1. 28	0. 72
10	0	0. 405	0. 72	0. 945	1. 08	1. 125	1. 08	0. 945	0. 72	0. 405
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0

X Y

Fig. 2. Partial elevation data.

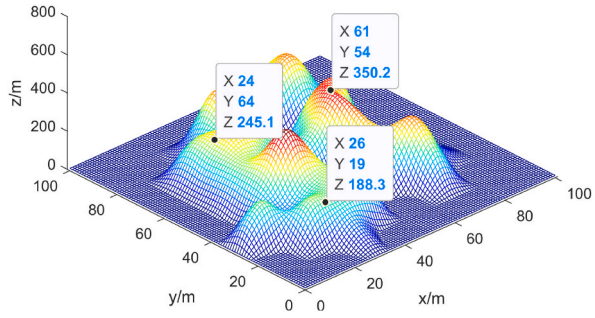

Fig. 3. Flight environment.

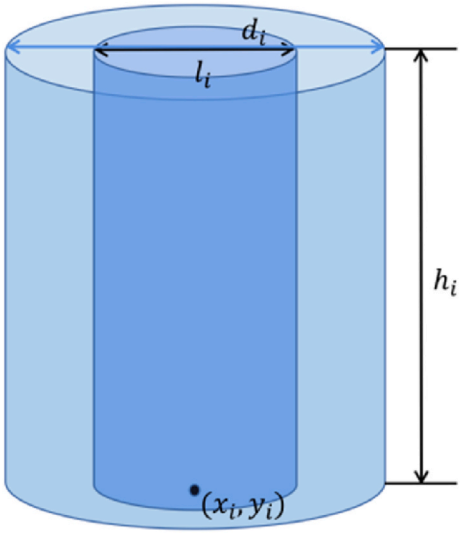

Fig. 4. A detailed diagram of the threat area.

where (x_i, y_i) is the center of the underside of the cylinder, h_i is the height of the cylinder, d_i and l_i represent the diameter of the outer cylinder and the inner cylinder respectively.

We usually use the path parameterization technology to describe the path in path planning. There are two methods to represent the path [8]. The first method is rotating coordinate system, which can reduce the dimension of the problem, but this method also has obvious disadvantages: because it can only fly along the rotating X axis, it cannot effectively guide UAV through obstacles. The second method is the B-Spline which is flexible and can maps UAV paths with arbitrary and smooth shapes, it is widely adopted in path representation [41]. In addition, the method can be realized with only a few parameters, which saves the calculation time. B-spline curve is introduced as follows:

Assume $n+2$ control points with coordinates (xc_0, yc_0, zc_0), …, (xc_k, yc_k, zc_k), …, $(xc_{n+1}, yc_{n+1}, zc_{n+1})$. These points can form curves using the B-spline method. The discrete serial points with coordinates (xp_t, yp_t, zp_t) can be generated as:

$$\begin{cases} xp_t = \sum_{j=0}^{n+1} xc_j \bullet B_{j,k}(t) \\ yp_t = \sum_{j=0}^{n+1} yc_j \bullet B_{j,k}(t) \\ zp_t = \sum_{j=0}^{n+1} zc_j \bullet B_{j,k}(t) \end{cases} \tag{2}$$

the curve's blending function, denoted as $B_{j,k}(t)$, k is determined by the curve's sequence, which is correlated to the smoothness of the curve. These blending functions are defined in a recursive manner with respect to a predetermined set of Kn values and can be expressed as:

$$B_{j,1}(t)=\begin{cases} 1 \quad if\ Kn(j)\le t\le Kn(j+1) \\ 1 \quad if\ Kn(j)\le t\le Kn(j+1)\ and\ t=n-k+3 \\ 0 \quad otherwise \end{cases} \tag{3}$$

$$B_{j,k}(t)=\frac{(t-Kn(j))\times B_{j,k-1}(t)}{Kn(j+k-1)-Kn(j)}+\frac{Kn(i+k-1)\times B_{i+1,k-1}(t)}{Kn(j+k)-Kn(j+1)} \tag{4}$$

$$\begin{cases} Kn(j)=0 \quad if\ j<k \\ Kn(j)=j-k+1 \quad if\ k\le j\le n \\ Kn(j)=n-k+2 \quad if\ n<j \end{cases} \tag{5}$$

where t varies from 0 to $n-k+3$ in constant steps, which offers a series of discrete points [42]. A complete model as shown in Fig. 5. It can be formed by splicing Figs. 3 and 4 together, combining with B-spline notation.

3.2. Objective function

In assessing the viability of the proposed algorithm, the objective function must take into account the following factors.

(1) Effects of path length

Due to the limited fuel carried by UAV, its flight distance should be as short as possible. The B-Spline curve allows for the path to be depicted as a collection of distinct points, including: $(p_0,p_1,\ldots,p_k,\ldots,p_{n+1})$. The coordinates of p_k are (xp_k, yp_k, zp_k) which is a discrete point representing the current position of the UAV. The starting point of the path is represented as p_0, while the endpoint is denoted as p_{n+1}. In general, a shorter distance between these two points requires less time. The distance can be determined using the Euclidean operator, which follows:

$$f_{length}=1-\left(\frac{L_{xp_0xp_{n+1}}}{L_{true}}\right) \tag{6}$$

here, $L_{xp_0xp_{n+1}}$ represents the length of the straight line that joins the starting point xp_0 with the endpoint xp_{n+1}. On the other hand, L_{true} denotes the true length of the trajectory, which can be computed using the following method:

$$L_{true}=\sum_{k=0}^{n}\sqrt{(xp_{k+1}-xp_k)^2+(yp_{k+1}-yp_k)^2+(zp_{k+1}-zp_k)^2} \tag{7}$$

the variable k in the equation represents the index of the discrete points, with values ranging from 0 to n.

(2) Effects of flight altitude

In order to prevent enemy radar detection, flying at low altitude is safer than flying at high altitude. However, in order to reduce the risk of hitting mountains during low flying, the flight altitude needs to be kept within a certain criterion. We have established specific minimum and maximum flight altitudes within our model. When the UAV flies in this range, it is the best. When it exceeds this range, it needs to be punished.

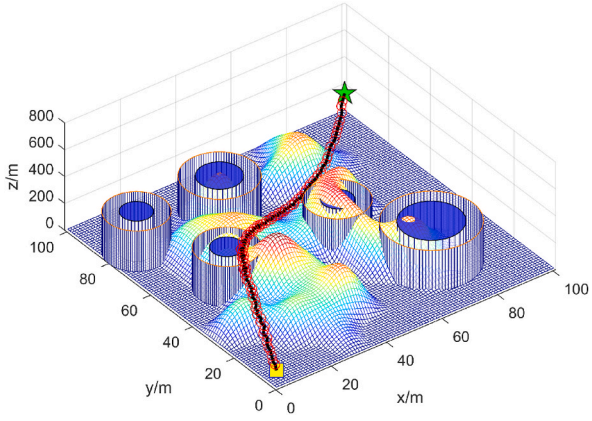

Fig. 5. The environment and route demonstration model.

$$f_{alti}=\begin{cases}\dfrac{\sum_{i=1}^{n}\left(1-\dfrac{h_{min}}{zp_i-f(xp_i,yp_i)}\right)}{n} & zp_i>f(xp_i,yp_i)+H_{max}\\ 0 & f(xp_i,yp_i)+H_{min}<zp_i<f(xp_i,yp_i)+H_{max}\\ \dfrac{\sum_{i=1}^{n}\left(\dfrac{h_{min}}{zp_i-f(xp_i,yp_i)}-1\right)}{n} & f(xp_i,yp_i)\le zp_i\le f(xp_i,yp_i)+H_{min}\end{cases} \tag{8}$$

in this equation, the current position of the UAV is represented by (xp_i, yp_i, zp_i). The function $f(xp_i, yp_i)$ calculates the height of the terrain. The permitted minimum flying height of the UAV is represented by H_{min}.

(3) Effects of danger zones

In this paper, the threat location is known, we have defined a 100% danger zone in the threat zone, in which the UAV is bound to suffer damage, so the UAV cannot fly to the 100% threat zone. The threat area in this paper is composed of two-layer cylinder models of different sizes and positions, *danger zones*$_i$ $[x_i, y_i, d_i, h_i, l_i]$, where (x_i, y_i) is the center on the *XOY* plane, d_i is the diameter of cylinder bottom, h_i is the covered height, l_i is the diameter of the no-fly zones. A detailed diagram of the threat area is shown in Fig. 3. The term associated with the violation of the danger zones for an UAV with coordinate $[xp_t, yp_t, zp_t]$ is defined as follows:

$$f_{danger}=\begin{cases}0 & if\ zp_t>h_i\\ 0 & if\ zp_t<h_i, dd\ge d_i/2\\ 1-\dfrac{dd-l_i/2}{(d_i-l_i)/2} & if\ zp_t<h_i, l_i\Big/2\le dd<d_i\Big/2\\ 1 & if\ zp_t<h_i, dd<l_i/2\end{cases} \tag{9}$$

where $dd=\sqrt{(xp_t-x_i)^2+(yp_t-y_i)^2}$ is the distance between the projection point of the current UAV position on the XY plane with the threat center.

(4) Effects of ground collisions

In order to ensure the normal flight of UAV, our planned route cannot appear in the mountain. The term associated with ground collisions is defined as follows:

$$f_{ground}=\begin{cases}0 & zp_k>h(xp_k,yp_k)\\ 1 & otherwise\end{cases} \tag{10}$$

where zp_k represents the position of the axis corresponding to the index point of UAV, and $h(xp_k, yp_k)$ is the terrain height corresponding to point (xp_k, yp_k).

(5) Effects of angle

The finally condition is that the angle of the UAV is limited. It should not exceed a predetermined maximum angle θ_{max}. The turning angle can be calculated by these discrete points as follows:

$$f_{angle}=\begin{cases}0 & if\ \theta<\theta_{max}\\ \dfrac{\theta-\theta_{max}}{180} & otherwise\end{cases} \tag{11}$$

where $\theta=cos^{-1}\left(\frac{\overrightarrow{p_tp_{t+1}}\ \overrightarrow{p_tp_{t-1}}}{||p_tp_{t+1}||\ ||p_tp_{t-1}||}\right)$, and p_t, p_{t+1} are two discrete and continuous points along the path, and the range of t is between 0 and n+1.

Based on the above discussion, this paper assumes that the weight values of each part are the same, and the values of the five constraint functions are all between 0 and 1, so there is no need to add additional weight. The UAV path planning model can be depicted as follows:

$$f_{obj}=f_{length}+f_{alti}+f_{danger}+f_{ground}+f_{angle} \tag{12}$$

4. Proposed method for global path planning

4.1. Basic algorithm

4.1.1. The SSA algorithm

SSA [10] is a swarm optimization approach inspired by the behavior of sparrows. The algorithm divides the sparrow population into two groups: producers and scroungers. The producers are responsible for finding food and guiding the displacement direction of the entire population. The better the fitness value, the more food the producers can get. The scroungers, on the other hand, move around the producers and try to improve the solutions found by the producers. In each iteration of the algorithm, the location of the producer updates are calculated by a specific formula, which is not mentioned in the given paragraph. Overall, the SSA algorithm aims to find the optimal solution for an optimization problem by mimicking the group wisdom, foraging, and anti-predation behaviors of sparrows. The location of the producer updates are as follows:

$$X_{ij}^{t+1} = \begin{cases} X_{ij}^{t} \bullet \exp\left(\dfrac{-i}{\alpha \bullet iter_{max}}\right) & if\ R_2 < ST \\ X_{ij}^{t} + Q \bullet L & if\ R_2 > ST \end{cases} \tag{13}$$

the equation given involves multiple variables and constants. The variable i takes values from 1 to N, where N represents the swarm size. The variable t represents the current iteration. The variable j takes values from 1 to d, where d represents the number of dimensions. X_{ij}^{t} denotes the value of the j-th dimension of the i-th sparrow at iteration t. The constant $iter_{max}$ represents the maximum number of iterations. The constant α is a random number between 0 and 1. The variable Q is a random number that follows a normal distribution. The matrix L is a $1 \times d$ matrix with all elements equal to 1. The variables R_2 and ST represent the alarm value and the safety threshold respectively. The value of R_2 lies between 0 and 1, while the value of ST lies between 0.5 and 1.0.

As for the scroungers, the producers are spied on by some scroungers, and upon detecting a superior food source, promptly desert their current position to engage in a fierce competition for the new resource. The formula governing the update of the scroungers' position is expressed as follows:

$$X_{ij}^{t+1} = \begin{cases} Q \bullet \exp\left(\dfrac{X_{worst}^{t} - X_{ij}^{t}}{i^2}\right) & if\ i > n/2 \\ X_{p}^{t+1} + \left|X_{ij}^{t} - X_{p}^{t+1}\right| \bullet A^{+} \bullet L & otherwise \end{cases} \tag{14}$$

where X_p was defined as the optimal position occupied of producers. X_{worst} is the current global worst location. $A^{+} = AT(AA^{T})^{-1}$, and here A represents a matrix of $1 \times d$ for which each element inside is randomly assigned 1 or -1.

When danger is detected, a swift relocation to a secure area is observed among sparrows situated at the periphery of the group. This behavior is motivated by their desire to obtain a superior vantage point. On the other hand, sparrows located at the center of the group tend to display a more erratic behavior, characterized by random wandering, with the aim of approaching nearby sparrows. The mathematical model governing this behavior can be expressed as follows:

$$X_{ij}^{t+1} = \begin{cases} X_{best}^{t} + \beta \bullet \left|X_{ij}^{t} - X_{best}^{t}\right| & if\ f_i > f_g \\ X_{ij}^{t} + K \bullet \left(\dfrac{\left|X_{ij}^{t} - X_{worst}^{t}\right|}{(f_i - f_w) + \varepsilon}\right) & if\ f_i = f_g \end{cases} \tag{15}$$

the given equation represents a mathematical formula for the SSA used in optimization problems. In this equation, X_{best}^{t} represents the current global optimal location, while X_{worst} represents the current global worst location. The parameter β is a step size control parameter, which is a normal distribution of random numbers with a mean value of 0 and variance of 1. K is a random number that lies in the range of [-1, 1]. The values f_i and f_w represent the fitness values of the present sparrow and the worst fitness value, respectively, while f_g represents the current global best fitness value. Additionally, ε is the smallest constant. By using this equation, the SSA aims to find the optimal solution for an optimization problem.

4.1.2. The PSO algorithm

Particle Swarm Optimization (PSO) [43] is a kind of swarm intelligence optimization algorithm. In this algorithm, each particle represents a possible solution to a problem and is identified by a position vector X_{id} and a velocity vector V_{id}. The swarm size is denoted by N, and the dimension of the particle is denoted by d. The update formulas for the velocity and position of each particle are based on its personal best experience and the population best experience. The update formulas governing these processes are expressed as follows:

$$V_{id}^{t+1} = \omega V_{id}^{t} + c_1 r_1 \left(p_i^{t} - X_{id}^{t}\right) + c_2 r_2 \left(X_{best}^{t} - X_{id}^{t}\right) \tag{16}$$

$$X_{id}^{t+1} = X_{id}^{t} + V_{id}^{t+1} \tag{17}$$

where t represents the current iteration number, $t = 1, 2, \ldots, iter_{max}$, $iter_{max}$ represents the total iteration number, where p_i denotes the personal historical best position of particle i, X_{best} denotes the population historical best position, ω is the inertia weight, c_1 and c_2 are the acceleration coefficients, r_1 and r_2 are two uniformly distributed random numbers.

4.2. A hybrid algorithm of particle swarm optimization and enhanced sparrow algorithm

4.2.1. Enhanced sparrow search algorithm (ESSA)

The producers in the SSA algorithm, the better the fitness value, the more food they can get, this group of particles plays a crucial role in locating food resources and determining the overall displacement direction of the population. As a result of their efforts, the producers are able to locate food resources on a much larger scale. But in the basic SSA algorithm, when $R_2 > ST$, the producer will randomly move to the current position according to the normal distribution $R_2 < ST$, the producer multiplies a function $e^{(-i/(\alpha \bullet iter_{max}))}$. It is not difficult to see that the value range of this function is between [0,1], at this time, the producer's position will gradually jump to 0, which is not a good strategy for global exploration. Therefore, we change the producer's position update strategy into a step-by-step moving process, adding a standard normal distribution random number Q. The finally location of the producer is updated as below:

$$X_{id}^{t+1} = \begin{cases} X_{id}^{t} \bullet (1+Q) & if\ R_2 < ST \\ X_{id}^{t} + Q & if\ R_2 > ST \end{cases} \tag{18}$$

where t indicates the current iteration, $t = 1, 2, \ldots, N_{iter}$, N_{iter} represents the total iteration number. X_{id}^{t} represents the value of the d th dimension of the i th sparrow at iteration t. R_2 ($R_2 \in [0, 1]$) and the variable ST, which lies within the range of [0.5, 1.0], represents a set of random numbers that are used to determine the alarm value and safety threshold. Additionally, the variable Q is drawn from a normal distribution.

As for the scroungers, they follow the producer to find food, while a group of scroungers monitor the producers, and when the producer spots something, they will show up to grab the food. But in the original SSA algorithm, when the follower is greater than $n/2$, the value of the scroungers are the product of a standard normal distribution random number and an exponential function based on the natural logarithm. When the population converges, its value will gradually converge to 0, which is obviously different from the strategy mentioned. Therefore, this step will be removed in our new algorithm, only the scroungers that may constantly monitor the producers and compete for food is retained. The position update formula for the scrounger is described as follows:

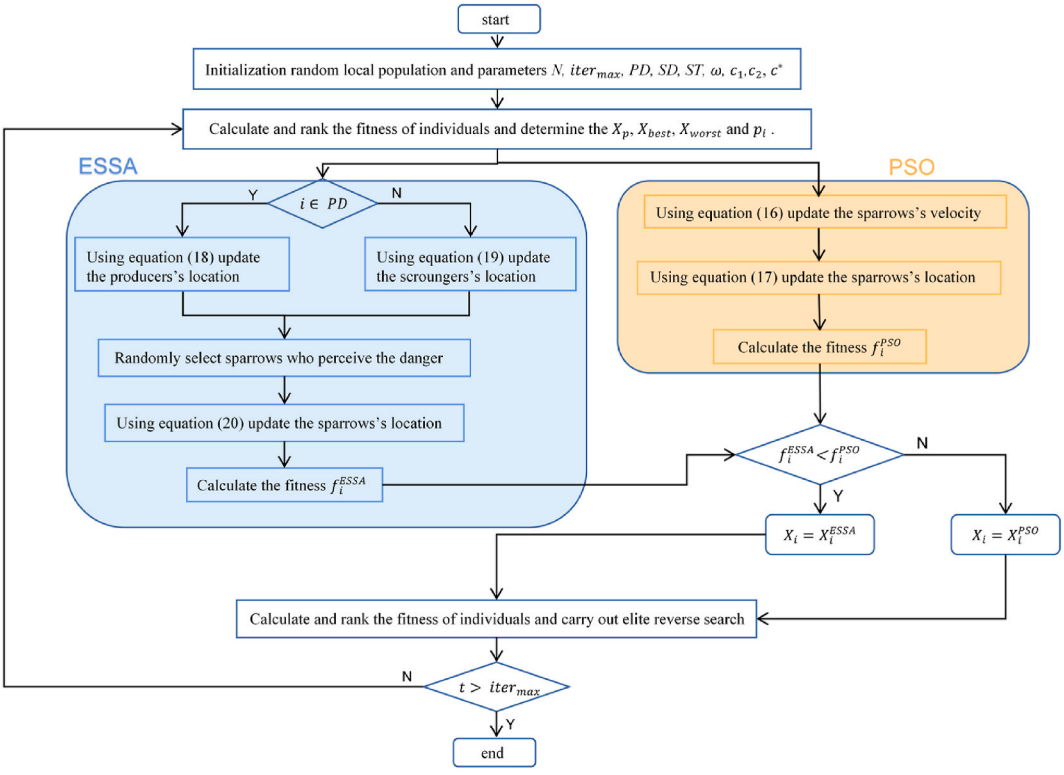

Fig. 6. The flowchart of the proposed PESSA algorithm.

$$X_{id}^{t+1} = X_p^{t+1} + \left| X_{id}^t - X_p^{t+1} \right| \bullet A^+ \tag{19}$$

where X_p is the optimal position occupied by the producer. A represents a matrix of $1 \times d$ for which each element inside is randomly assigned 1 or−1, and $A^+ = AT(AA^T)^{-1}$.

In SSA, when danger appeared, sparrows at the edge of the group moved quickly to the safety area, while sparrows in the middle of the group moved randomly to other sparrows. In the update formula of SSA algorithm, when the early-warning sparrows are at the edge of the group, they will jump to the current optimal position, that is, they will move to the safe area, in line with the rules. However, when the early-warning sparrow is in the current optimal position, it will escape to a position near itself, which is inconsistent with the rules. Therefore, when the early-warning sparrow is in the current optimal position, we let it randomly jump to a random position between the optimal position and the worst position. The mathematical model can be expressed as follows:

$$X_{id}^{t+1} = \begin{cases} c \bullet X_{best}^t + \beta \bullet \left| X_{id}^t - X_{best}^t \right| & if\ f_i > f_g \\ c \bullet X_{id}^t + \beta \bullet \left| X_{worst}^t - X_{best}^t \right| & if\ f_i = f_g \end{cases} \tag{20}$$

where $c(t) = \frac{c^* t}{iter_{max}} + c^*$ are adaptive coefficient that will increase with the number of iterations and c^* is a constant. The remaining parameters have the same meaning as the SSA algorithm.

4.2.2. A hybrid algorithm of particle swarm optimization and enhanced sparrow algorithm (PESSA)

Despite several improvements, the SSA algorithm still has certain limitations, such as single search method, easy to fall into local optimum. Therefore, we add particle swarm optimization algorithm to ESSA algorithm for parallel operation. After each iteration, choose the best result. Finally, in order to increase the global search ability and the ability to jump out of the local optimum, we reverse search some elite sparrows to improve the global search ability of the algorithm. Fig. 6 is the flow chart of PESSA algorithm in this paper. The steps to implement the algorithm are as follows.

Step 1. Initialization

We set the size of population (N), the maximum number of iterations ($iter_{max}$) and some necessary parameters such as the number of producers (PD), scroungers (N-PD) and the sparrows who perceive the danger (SD) and the safety threshold (ST), and the inertia weight ω of PSO algorithm and c_1, c_2 in this step. At the same time, the population's search area is limited.

Step 2. Executing ESSA algorithm

Calculate and rank the fitness of individuals and determine the X_p, X_{best} and X_{worst}. Then using equation (18) update the producers 's location, using equation (19) update the scroungers 's location. Last, randomly select sparrows who perceive the danger that using

Table 1
Information of benchmark functions.

No.	Name	Definition	Search range	f_{min}
F1	Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	[-100, 100]	0
F2	Schwefel 2.22	$f_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	[-10, 10]	0
F3	Quadric	$f_3(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	[-100, 100]	0
F4	Rosenbrock	$f_4(x) = \sum_{i=1}^{D-1} (100(x_i^2) - x_{i+1})^2 + (x_i - 1)^2$	[-100, 100]	0
F5	Rastrigin	$f_5(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	[-5, 5]	0
F6	Ackley	$f_6(x) = -20\exp\left(-0.2\sqrt{\frac{\sum_{i=2}^{D} x_i^2}{D}}\right) - \exp\left(\sum_{i=1}^{D} \frac{\cos(2\pi x_i)}{D}\right) + 20 + e$	[-32, 32]	0
F7	Griewank	$f_7(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]	0
F8	Generalized Penalized Function 1	$f_8(x) = \frac{\pi}{D}(10\sin(\pi y_1))^2 + \sum_{i=1}^{D-1}(y_i - 1)^2 [1 + (10\sin(\pi y_{i+1}))^2] + (y_D - 1)^2 + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i < a \\ 0, -a \ll x_i \ll a \\ k(-x_i - a)^m, x_i < -a \end{cases}$	[-50, 50]	0
F9	Generalized Penalized Function 2	$f_9(x) = \sum_{i=1}^{D} 0.1(\sin(3\pi x_1)^2) + \sum_{i=1}^{D-1}(x_i - 1)^2[1 + (\sin(3\pi x_{i+1}))^2] + (x_D - 1)^2(1 + \sin(2\pi x_D))^2 + \sum_{i=1}^{D} u(x_i, 5100, 4)$	[-50, 50]	0
F10	Schwefel	$f_{10}(x) = \sum_{i=1}^{D} -x_i \sin(\sqrt{\lvert x_i \rvert})$	[-500, 500]	$-418.9829 \times D$

equation (20) update the sparrows 's location.

Step 3. Executing PSO algorithm

First determine the X_{best} and p_i and then using equation (16) update the sparrows 's velocity, Finally, using equation (17) update the sparrows 's location.

Step 4. Keep good results and conduct elite reverse search

According to the results of the second and third steps, we choose the better sparrow position as the final position and sort them. Then the elite sparrow is searched backward to increase the ability of the algorithm to jump out of the local optimum and yield the optimum diversity. The inverse solution $X_{id}^{t\,*}$ is defined as:

$$X_{id}^{t\,*} = k\left(a_d^t + b_d^t - X_{id}^t\right) \tag{21}$$

where $a_d^t = \min\left(X_{1d}^t, X_{2d}^t, \ldots, X_{nd}^t\right)$, $b_d^t = \max\left(X_{1d}^t, X_{2d}^t, \ldots, X_{nd}^t\right)$ and k is a random number between 0 and 1.

Step 5. Perform Step 2 until the end of the iteration

Stop running when the stop condition is reached, otherwise keep running.

5. Experimental analysis

In order to verify the performance of the PESSA algorithm, this paper tested on 10 basic functions and compared them with 12 algorithms. The 10 basic functions include 4 unimodal functions and 6 multimodal functions. The basic test functions f1-f4 are unimodal functions and f5-f10 are multimodal functions, which are the same as the benchmarks used in Ref. [10]. This paper sets the dimension is D = 30. All benchmark test functions are shown in Table 1.

We set two experiments to evaluate the performance of the PESSA algorithm. Firstly, we compared PESSA algorithm with six basic algorithms including SSA [10], PSO [45], GWO [44], SCA [45], MVO [46] and MFO [47] on ten basic functions. Secondly, the PESSA algorithm is compared with other six improved algorithms based on SSA or PSO including CSSA-SCN [48], CASSA [49], SCA-CSSA [50], HCLPSO [51], GPAM-PSO [52] and A-PSO [53] in ten basic functions. In these experiments, the population size was set to 100 and the maximum number of iterations (termination criteria) was set to 1000. All experiments were run independently 30 times. The parameters of the PESSA are set as follows: the number of the producers (PD) and SD accounts for 60% and 10%, respectively, ST = 0.6 and $c^* = 0.5$. $\theta_{max} = 60^\circ$, $H_{min} = 5$ m, $H_{max} = 300$ m, $P = 10000$. The parameter settings of other algorithms are consistent with the original. The relevant data of the two groups of experiments are recorded in Table 2 and Table 4 respectively, such as mean (mean), standard deviation (SD) and optimal value (Best).

Table 2
Results of SSA, PSO, GWO, SCA, MVO, MFO and PESSA on ten benchmark functions.

Function		SSA	PSO	GWO	SCA	MVO	MFO	PESSA
F1	Mean	8.6046e-292	3.1165e-02	3.6836e-85	1.5628e-04	9.4687e-02	3.3333e+02	**0**
	Std	**0**	1.8825e-02	8.3971e-85	6.3107e-04	2.8433e-02	1.8257e+03	**0**
	Best	**0**	5.4914e-03	2.6359e-87	9.0934e-10	4.2472e-02	8.726e-07	**0**
F2	Mean	7.3143e-195	1.7342e-01	2.8433e-49	2.0688e-06	1.8646e-01	3.8000e+01	**0**
	Std	**0**	9.9054e-02	3.2247e-49	5.8033e-06	5.2861e-02	2.0577e+01	**0**
	Best	**0**	5.1978e-02	1.7921e-50	9.1897e-10	1.1008e-01	9.7604e-05	**0**
F3	Mean	2.6507e-257	1.1279e+01	6.3612e-27	1.2268e+03	6.8697e+00	1.3610e+04	**0**
	Std	**0**	3.5003e+00	1.9081e-26	2.2050e+03	3.1720e+00	1.0236e+04	**0**
	Best	**0**	5.8004e+00	3.0838e-31	1.3025e+01	2.1372e+00	5.2635e+01	**0**
F4	Mean	7.9742e-05	1.7736e+02	2.6252e+01	4.4619e+02	2.6585e+03	2.0275e+05	1.1663e-08
	Std	1.9542e-05	2.9505e+02	7.0651e-01	1.3182e+03	3.6288e+03	4.0548e+05	2.9496e-08
	Best	1.8408e-06	3.1073e+01	2.4773e+01	2.7165e+01	3.4741e+01	1.9104e+01	8.6372e-12
F5	Mean	**0**	5.3919e+01	**0**	3.5888e+00	1.0418e+02	1.2740 e+02	**0**
	Std	**0**	1.0632e+01	**0**	9.0529e+00	2.7639e+01	3.9267e+01	**0**
	Best	**0**	3.3222e+01	**0**	6.8766e-08	5.7774e+01	6.0692e+01	**0**
F6	Mean	**0**	1.1153e-01	1.0836e-14	1.0879e+01	4.2859e-01	8.2406e+00	**0**
	Std	**0**	4.4456e-02	3.1501e-15	9.6805e+00	5.7713e-01	9.0694e+00	**0**
	Best	**0**	3.2898e-02	7.9936e-15	1.8611e-05	6.1827e-02	5.0558e-04	**0**
F7	Mean	**0**	1.6733e-02	2.5154e-03	1.1755e-01	3.0547e-01	9.0227e+00	**0**
	Std	**0**	1.6537e-02	8.5660e-03	2.0879e-01	7.8344e-02	3.6257e+02	**0**
	Best	**0**	9.3610e-04	**0**	3.5336e-08	1.5055e-01	8.885e-06	**0**
F8	Mean	1.9631e-12	2.8178e-02	1.3970e-02	4.6604e-01	6.9498e-01	3.1610e-01	9.5380e-27
	Std	4.153e-12	6.0454e-02	8.7598e-03	1.8040e-01	9.6499e-01	4.8726e-01	4.9372e-27
	Best	8.8473e-17	4.7899e-05	5.7246e-07	2.5426e-01	3.4467e-04	3.5904e-06	3.9283e-30
F9	Mean	7.3657e-12	5.9275e-03	1.5562e-01	2.8379e+00	1.5411e-02	1.9534e-02	1.2406e-24
	Std	1.0950e-11	4.9632e-03	1.2250e-01	3.2526e+00	9.8602e-03	7.2157e-02	1.1670e-24
	Best	8.5295e-15	7.2987e-04	5.5958e-06	1.7384e+00	4.6413e-03	2.8524e-06	3.1790e-26
F10	Mean	−8.8652e+03	−6.2858e+03	−6.5905e+03	−4.1494e+03	−8.0880e+03	−8.9504e+03	−1.2569e+04
	Std	2.3591e+03	1.36803 + 03	5.5452e+02	2.4742e+02	6.8191e+02	8.8935e+02	**0**
	Best	−1.2569e+04	−8.52253 + 03	−7.5823e+03	−4.6491e+03	−9.3873e+03	−1.0929e+04	−1.2569e+04

Table 3
Statistical results of ten different benchmark functions.

	SSA	PSO	GWO	SCA	MVO	MFO	PESSA
Mean	3	0	1	0	0	0	10
Std	6	0	1	0	0	0	10
Best	6	0	2	0	0	0	10

Table 4
Results of CSSA-SCN, CASSA, ISSA, HCLPSO, GPAM-PSO, A-PSO and PESSA on ten benchmark functions.

Function		CSSA-SCN	CASSA	SCA-CSSA	HCLPSO	GPAM-PSO	A-PSO	PESSA
F1	Mean	**0**	4.13e-04	3.83e−04	6.96e-24	1.29e-19	6.29e−284	**0**
	Std	**0**	1.91e-03	1.10e−02	1.49e-23	6.75e-19	**0**	**0**
	Best	**0**	**0**	**0**	1.01e-25	1.07e-25	**0**	**0**
F2	Mean	**0**	\	5.03e−03	2.40e-13	1.14e-08	2.18e−190	**0**
	Std	**0**	\	1.26e−01	5.62e-13	3.70e-08	**0**	**0**
	Best	**0**	\	**0**	1.77e-14	4.31e-12	**0**	**0**
F3	Mean	**0**	1.40e-01	4.56e−02	9.60e-01	1.79e-17	4.40e−279	**0**
	Std	**0**	3.08e+02	4.55e−01	5.96e-01	8.31e-17	**0**	**0**
	Best	**0**	**0**	**0**	1.57e-01	1.45e-25	**0**	**0**
F4	Mean	5.30e−05	5.42e-02	1.45e−02	1.79e+01	2.90e-05	\	1.16e-08
	Std	9.30e-05	4.17e-01	2.84e−01	2.37e-01	6.35e-05		2.94e-08
	Best	1.43e-07	2.65e-09	7.80e−08	2.37e-01	4.17e-08		8.63e-12
F5	Mean	**0**	4.35e-02	4.12e−02	3.54e-08	7.41e-01	**0**	**0**
	Std	**0**	9.70e-01	1.10e+00	9.99e-08	5.29e+00	**0**	**0**
	Best	**0**	**0**	**0**	**0**	**0**	**0**	**0**
F6	Mean	8.88e−16	4.70e-03	8.88e−16	7.12e-12	5.21e-10	8.88e−16	**0**
	Std	8.88e−16	8.92e-02	8.88e−16	8.99e-12	1.90e-09	8.88e−16	**0**
	Best	8.88e−16	**0**	8.88e−16	7.71e-13	4.26e-14	8.88e−16	**0**
F7	Mean	**0**	2.37e-05	1.91e−04	2.58e-12	**0**	**0**	**0**
	Std	**0**	4.61e-04	5.36e−03	1.28e-11	**0**	**0**	**0**
	Best	**0**	**0**	**0**	**0**	**0**	**0**	**0**
F8	Mean	3.12e−09	2.04e-06	3.13e−04	4.62e-23	2.03e-03	1.00e−08	9.53e-27
	Std	8.52e−09	2.28e-05	1.47e−02	1.08e-22	1.45e-02	4.50e−09	4.93e-27
	Best	6.46e−15	3.38e-19	4.73e−32	6.86e-25	1.02e-10	3.28e−09	3.92e-30
F9	Mean	5.53e−08	\	3.39e−04	3.02e-21	2.37e-03	2.41e−05	1.24e-24
	Std	1.55e−07	\	8.64e−03	9.26e-21	4.56e-03	2.80e−05	1.16e-24
	Best	1.37e−14	\	1.35e−32	2.17e-23	2.52e-09	7.92e−06	3.17e-26
F10	Mean	−1.0826e+04	−8.9226e+03	\	−1.0609e+04	−1.1924e+04	\	−1.2569e+04
	Std	7.9683e+02	1.9132e+03	\	6.3310e+02	1.2842e+02	\	**0**
	Best	−1.2569e+04	−1.2525e+04		−1.2569e+04	−1.2569e+04	\	−1.2569e+04

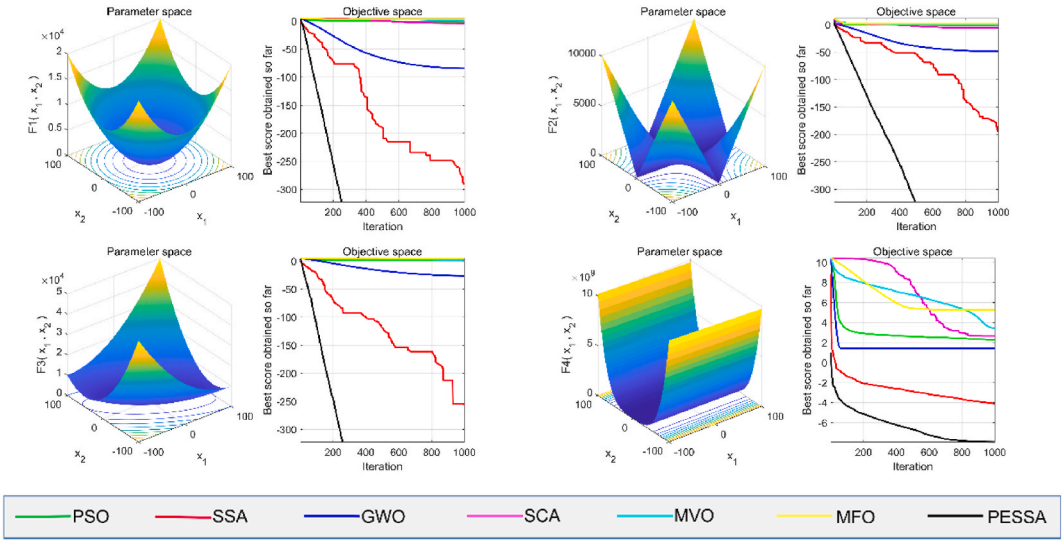

Fig. 7. Comparison of the optimization curves of seven different algorithms with four unimodal benchmark functions.

5.1. Comparison of PESSA with six basic intelligent optimization algorithms

In this part, we verify the effectiveness of the PESSA algorithm through 10 benchmark functions (Table 1), and compare it with 6 algorithms to illustrate its superiority. Figs. 7 and 8 show the convergence process of four unimodal benchmark functions and six multimodal benchmark functions respectively. The minimum (Best), average (Mean) and variance (Std) of these seven algorithms are shown in Table 2.

We can see from Table 2, in the average results, PESSA algorithm gains seven theoretical optimal values out of ten benchmark functions except F4, F8 and F9. SSA algorithm gets three theoretical optimal values in F5, F6 and F7. GWO algorithm gets one theoretical optimal value in F5. Other algorithms do not converge to the optimal value in these ten benchmark functions. In the standard values, PESSA algorithm obtains seven theoretical optimal values on ten benchmark functions except F4, F8 and F9. SSA gets six theoretical optimal values in F1-3 and F5-7. GWO algorithm gets one theoretical optimal value in F5. Other algorithms do not converge to the optimal value in these ten benchmark functions. In the best values, PESSA algorithm obtains seven theoretical optimal values on ten benchmark functions except F4, F8 and F9. SSA algorithm gets seven theoretical optimal values except F4, F8 and F9. GWO algorithm gets two theoretical optimal values in F5 and F7. Other algorithms do not converge to the optimal value in these ten benchmark functions.

The convergence speed of the algorithm can be seen from Figs. 7 and 8. In unimodal benchmark functions, PESSA algorithm has the fastest convergence speed in seven algorithms. Among them, F1 and F3 can converge to the optimum in about 200 generations, F2 in about 500 generations, although PESSA algorithm does not converge to the optimal value in F4, the convergence result and convergence speed are the best. In multimodal benchmark functions, PESSA algorithm has the fastest convergence speed in seven algorithms. Among them, the convergence speed of the F5–F7 is very fast, and it can converge to the optimal value in the 20 generations, F10 in about 50 generations, although PESSA algorithm does not converge to the optimal value in F8 and F9, the convergence result and convergence speed are the best. The algorithm proposed in this paper can jump out of the local optimal solution, which is more obvious in the multimodal benchmark functions, especially in the F8–F10 functions. While other algorithms fall into the local optimal solution in the process of searching for the optimal solution, the algorithm proposed in this paper can jump out of the local optimal solution. For example, on the F8 function, the PSO algorithm and GWO algorithm fell into the local optimal solution early and stopped converging after about 50 generations. SSA algorithm converges slowly in about 100 generations. Although some of SCA algorithms perform well in the middle, the final result is not so ideal. MVO algorithm and MFO algorithm converge slowly at 250 and 400 generations respectively, while PESSA algorithm temporarily falls into the local optimal solution at 50 to 250 generations, but quickly searches for the optimal solution after 250 generations, so PESSA algorithm can jumps out of the local optimal value. Functions F9 and F10 can also reflect the ability of the new algorithm to jump out of the local optimal solution.

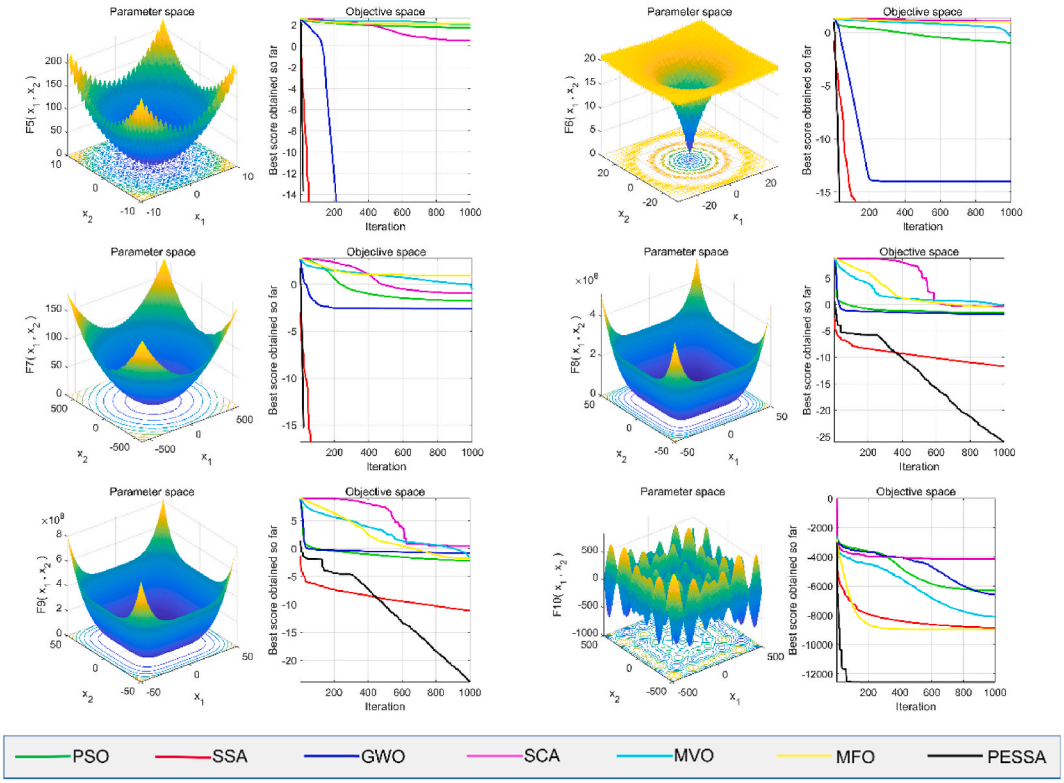

Fig. 8. Comparison of the optimization curves of seven different algorithms with six multimodal benchmark functions.

To prove the superiority of the PESSA algorithm effectively, Table 3 gives statistical results of seven algorithms ranking first among ten benchmark functions to further illustrate the effectiveness of PESSA. It can be clearly seen from Table 3 that the proposed PESSA algorithm is the best in average value, standard value and optimal value, it has the smallest values on 10 functions. SSA algorithm obtains the smallest mean values, standard deviations and best values on 3, 6, 6 functions, respectively. GWO algorithm obtains the smallest mean values, standard deviations and best values on 1, 1, 2 functions, respectively. The best results represent the optimization ability of algorithm, average results and standard deviations of optimization results reflect the stability and robustness of the algorithm. It can be concluded that our new algorithm has strong competitiveness in optimization ability, stability and robustness.

5.2. Comparison of PESSA with six related improved algorithms

In this section, the proposed PESSA algorithm is compared with six improved algorithms based on SSA or PSO algorithm, which are mainly CSA-SCN, CASSA, SCA-CSSA, HCLPSO, GPAM-PSO and A-PSO. The comparison object is the basic test function, and the test results prove the superiority of PESSA algorithm. The experimental results of ten benchmark functions are given as shown in Table 4.

We can see from Table 4, in the average results, PESSA algorithm gains seven theoretical optimal values out of ten benchmark functions except F4, F8 and F9, but it can get a better result (the deviation is less than e−10) on these two functions F8 and F9. CSSA-SCN algorithm gets five theoretical optimal values on F1–F3, F5 and F7, and can get a better result on F6. A-PSO gets two theoretical optimal values in F5 and F7, GPAM-PSO algorithm gets one theoretical optimal value on F7, it can get a better result on these three functions F1, F3 and F6. A-PSO algorithm gets two theoretical optimal values on F5 and F7, and can get a better result on these four functions F1, F2, F3 and F6. Other algorithms do not converge to the optimal value in these ten benchmark functions. SCA-CSSA algorithm can get a better result on F6. HCLPSO algorithm can get a better result on these six functions F1, F2, F6, F7, F8 and F9. In the standard values, PESSA algorithm obtains seven theoretical optimal values on ten benchmark functions except F4, F8 and F9, it can get a better result on these two functions F8 and F9. CSSA-SCN algorithm gets five theoretical optimal values on F1–F3, F5 and F7, and it can get a better result on F6. GPAM-PSO algorithm gets one theoretical optimal value on F7, and it can get a better result on these two functions F1 and F3. A-PSO algorithm gets five theoretical optimal values on F1–F3, F5 and F7, and can get a better result on F6. Other algorithms do not converge to the optimal value in these ten benchmark functions. SCA-CSSA algorithm can get a better result on F6. HCLPSO algorithm can get a better result on these six functions F1, F2, F6, F7, F8 and F9. In the best values, PESSA algorithm obtains seven theoretical optimal values on ten benchmark functions except F4, F8 and F9, it can get a better result on these three functions F4, F8 and F9. CSSA-SCN algorithm gets six theoretical optimal values on F1–F3, F5, F7 and F10, and can find solutions which are very close to the optimal values on these three functions F6, F8 and F9. CASSA algorithm gets five theoretical optimal value on F1, F3, F5–F7, and can find solutions which are very close to the optimal values on F8. SCA-CSSA algorithm gets five theoretical optimal value on F1–F3, F5 and F7, and can get a better result on these three functions F6, F8 and F9. HCLPSO algorithm gets three theoretical optimal value on F5, F7 and F10, and can get a better result on these five functions F1, F2, F6, F8 and F9. GPAM-PSO algorithm gets three theoretical optimal value on F5, F7 and F10, and can find solutions which are very close to the optimal values on these five functions F1–F3, F6 and F8. A-PSO algorithm gets five theoretical optimal value on F1–F3, F5 and F7, and can get a better result on F6.

To prove the superiority of the PESSA effectively, Table 5 gives statistical results of seven improved algorithms ranking first among ten benchmark functions to further illustrate the effectiveness of PESSA algorithm. It can be clearly seen from Table 5 that the proposed PESSA algorithm is the best in average value and standard value, it has the smallest values on 10 functions and it has the smallest values on 8 functions in best value. CSSA-SCA algorithm obtains the smallest mean values, standard deviations and best values on 5, 5, 6 functions, respectively. CASSA algorithm obtains the best values on 5 functions. SCA-CSSA algorithm obtains the best values on 7 functions. HCLPSO obtains the best values on 2 functions. GPAM-PSO algorithm obtains the smallest mean values, standard deviations and best values on 1, 1, 2 functions, respectively. A-PSO algorithm obtains the smallest mean values, standard deviations and best values on 2, 5, 5 functions, respectively. Therefore, it can be concluded that PESSA outperforms six improved algorithms based on SSA or PSO algorithm in adopted benchmark functions.

By compared 10 basic functions with 12 algorithms, the superiority of the new algorithm is demonstrated. However, it is obvious that the complexity of PESSA algorithm is increased compared with SSA algorithm and PSO algorithm, and the corresponding computation time is also increased. This paper does not compare the specific value of time increase in detail, but the problem of sacrificing time to improve the accuracy is the deficiency of this algorithm.

6. Application of new algorithm in 3D path planning

In this section, in order to verify the effectiveness of the new algorithm in path planning, we design two different simulation environments, which are the 2D environment model and the 3D environment model. In the 2D environment model, we make the flight planning space's size is 100 m by 100 m. The start point is set to (0 m, 0 m). The target point is set to (100 × 100) m^2. In the 3D environment model, we set the flight planning space's size is (100 × 100 × 400) m^3, just like the model of Fig. 2. The start point is set to (0 m, 0 m, 0 m). The target point is set to (100 m, 100 m, 200 m). The maximum iteration number set as 300 and the population size of the three algorithms (SSA, PSO, PESSA) are set as 100. Other parameters are consistent with section 4. The results are averaged 30 independent runs.

In the 2D environment model, the threats are denoted by 10 circles and the related information is shown in Table 6. The comparative results with the SSA, PSO and PESSA algorithms are given. The comparison results of 2D experiments are shown in Table 7

Table 5
Statistical results of ten different benchmark functions.

	CSSA-SCN	CASSA	SCA-CSSA	HCLPSO	GPAM-PSO	A-PSO	PESSA
Mean	5	0	0	0	1	2	10
Std	5	0	0	0	1	5	10
Best	6	5	7	2	2	5	8

Table 6
The information of threat areas in the 2-D planning field.

Case number	Threat center	Threat radius	Case number	Threat center	Threat radius
1	(30, 10)	8	2	(25, 5)	15
	(25, 60)	10		(38, 42)	15
	(60, 20)	10		(55, 10)	15
	(50, 70)	10		(25, 80)	20
	(90, 30)	8		(90, 55)	10
	(45, 35)	8		(80, 20)	20
	(75, 35)	5		(45, 70)	15
	(75, 80)	8		(75, 80)	15
	(90, 60)	10		(90, 10)	10
	(10, 10)	5		(15, 40)	15

Table 7
Comparison results of the 2 cases in 2-D.

		SSA	PSO	PESSA
Case 1	Mean	0.1471	0.1066	0.0165
	Std	0.0993	0.0837	0.0038
	Best	0.0487	0.0373	0.0112
Case 2	Mean	0.3278	0.3016	0.0521
	Std	0.2837	0.1727	0.0048
	Best	0.1382	0.0837	0.0382

and the convergence curves of three algorithms in Figs. 9–12. In the 3D environment model, the threats are denoted by 10 cylinders like Fig. 3 and the related information is shown in Table 8. The comparative results with the SSA and PSO algorithms are also given. The 3D experiment comparison results are listed in Table 9 and the convergence curves of three algorithms in Figs. 13–15.

6.1. The results of the PESSA algorithm compared with the PSO and SSA algorithms on 2D experiment

In this part, we designed two flight environments altogether. For the first case, it can be seen from Fig. 9 that the experimental results of the three algorithms (SSA, PSO and PESSA) are different in the field of 2D planning. We can see that the planning results of PSO and PESSA algorithms can meet the requirements of path planning, while the path planned by SSA algorithm is not a smooth path, and a small part of it is circuitous. Obviously, the results of PESSA algorithm perform well compared with the other two algorithms, which further proves the superiority of the proposed algorithm. This property is more obvious in convergent curves that are illustrated

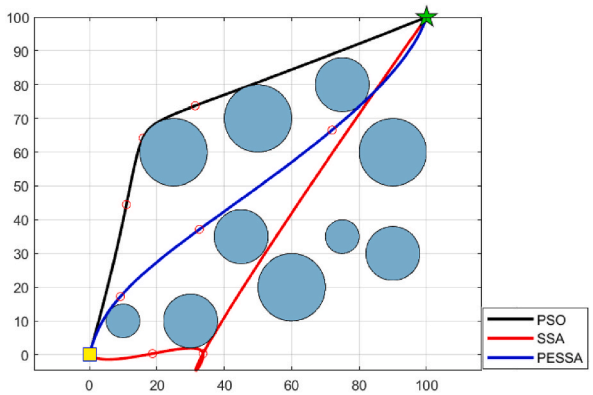

Fig. 9. The comparative path planning results of 3 algorithms in case 1, 2D.

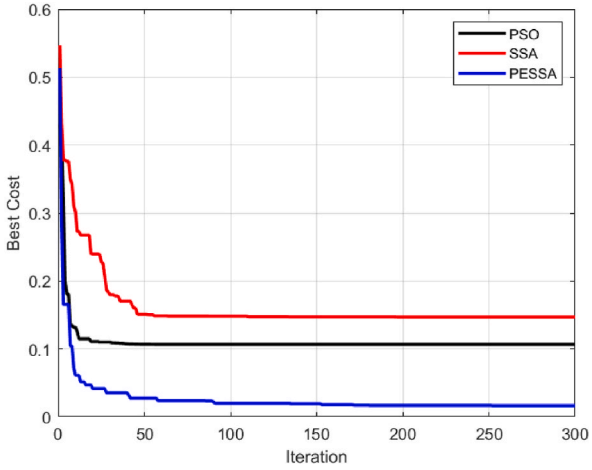

Fig. 10. The convergence curves of 3 algorithms in Case 1, 2D.

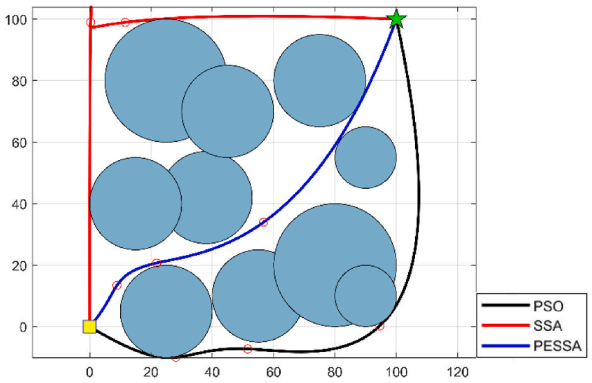

Fig. 11. The comparative path planning results of 3 algorithms in case 2, 2D.

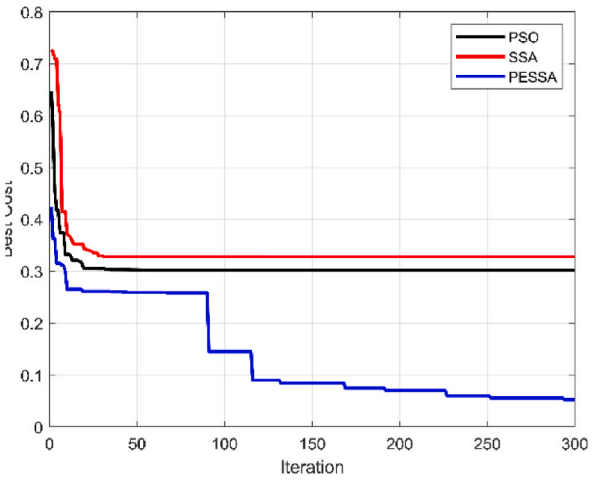

Fig. 12. The convergence curves of 3 algorithms in Case 2, 2D.

in Fig. 10, it can be seen that in the first 50 generations, the results of PESSA algorithm have far exceeded those of the other two algorithms, so the new algorithm has rapid convergence. The statistical results are illustrated in Table 7. The optimal value of the SSA is 0.0487, PSO is 0.0373 and PESSA is 0.0112. Obviously, our new algorithm is the best. The mean value of the SSA is 0.1471, PSO is

Table 8
The information of threat areas in the 3D planning field.

Case number	$[x_i, y_i]$	d_i	h_i	l_i	Case number	$[x_i, y_i]$	d_i	h_i	l_i
1	(30, 10)	16	300	10	2	(25, 5)	15	300	20
	(25, 60)	20	300	10		(38, 42)	15	300	20
	(60, 20)	20	250	14		(55, 10)	15	250	20
	(50, 90)	20	300	10		(25, 80)	20	300	30
	(90, 10)	16	300	10		(90, 55)	10	300	10
	(60, 50)	16	200	10		(80, 20)	20	200	30
	(75, 35)	10	250	6		(45, 70)	15	250	20
	(75, 85)	16	250	8		(75, 80)	15	300	10
	(90, 60)	20	250	10		(90, 10)	10	250	10
	(10, 92)	16	200	10		(15, 40)	15	200	20

Table 9
Comparison results of the 2 cases in 3-D.

		SSA	PSO	PESSA
Case 1	Mean	0.9123	0.7573	0.6635
	Std	0.0938	0.0837	0.0736
	Best	0.7293	0.6283	0.5329
Case 2	Mean	0.6642	0.9057	0.5349
	Std	0.0970	0.1495	0.0770
	Best	0.5922	0.6352	0.4792

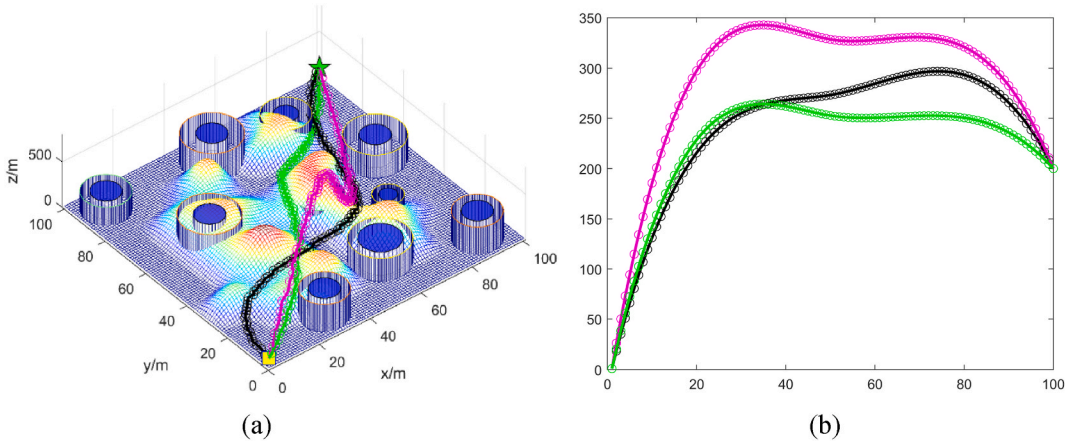

Fig. 13. The comparative path planning results of 3 algorithms in case 1, 3D. (a) Three-dimensional view. (b) Height curve from X to Z axis.

0.1066 and PESSA is 0.0165, the standard deviation of the SSA is 0.0993, PSO is 0.0837 and PESSA is 0.0038. The standard deviation of the PESSA is lowest, which proves that PESSA can search for the optimal path stably. After of all, the new algorithm performs best in all aspects.

In the second case, we used a more complex environment to prove the validity of the new algorithm. The comparative experimental results are shown in Fig. 11. The routes planned by PSO and PESSA algorithms meet the path planning requirements, while the routes planned by SSA do not meet the requirements due to fluctuations. We can clearly see that the path planned by the PESSA is better than SSA and PSO algorithm. The convergence curves of the three algorithms in Case 2 are shown in Fig. 12. It can be seen from Fig. 12 that the new algorithm can jump out of local optimum and find a better path at 100 generations. As shown in Table 7, the optimal value, average value and standard deviation of PESSA algorithm are the smallest compared with the other two algorithms, indicating that the new algorithm has superior performance.

To sum up, it can be seen from the above experimental results and analysis that PESSA algorithm can search a satisfactory path.

6.2. The results of the PESSA algorithm compared with the PSO and SSA algorithms on 3D experiment

In this part, we designed two flight environments altogether. For the first case, it is clearly show in Fig. 13 that there are significant differences in the experimental results of the three algorithms. Fig. 13(a) shows a three-dimensional strabismus view, where the route planned by the algorithm can be clearly seen. Fig. 13(b) is the height curve mapped from the X-axis to the Z-axis, from which the

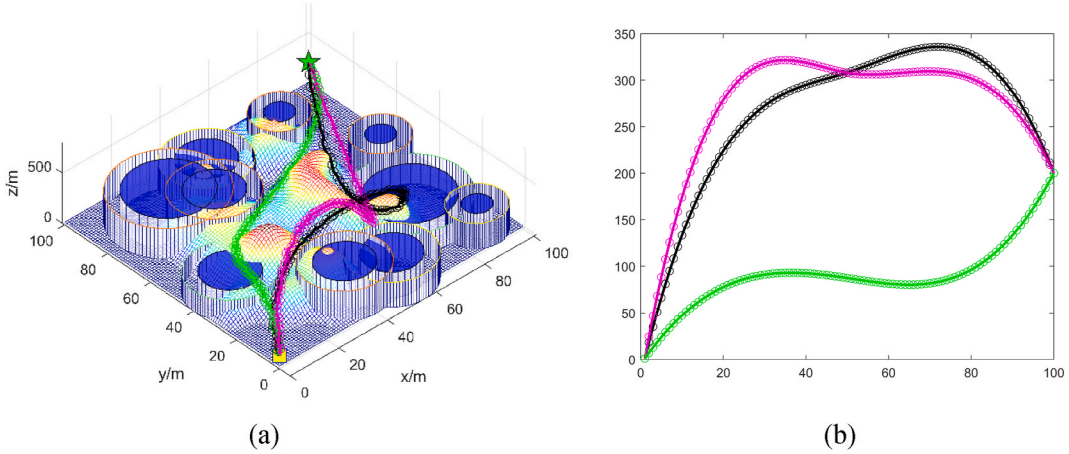

Fig. 14. The comparative path planning results of 3 algorithms in case 2, 3D. (a) Three-dimensional view. (b) Height curve from X to Z axis.

overall flight heights of the three routes can be seen. We can see that the path planning of PSO and PESSA algorithms can meet the requirements of path planning, while the path planning of SSA is oscillating. Among them, the path planned by PESSA algorithm is better than that planned by PSO. Fig. 14 shows the track planned by the three algorithms in the topographic map of case2, and Fig. 14 (a) represents the three-dimensional strabismus angle, and Fig. 14(b) represents the mapping from X-axis to Z-axis, reflecting the altitude of the flight path and its upper and lower frequencies, but it is obvious that the path planned by PESSA algorithm is the shortest, and the other two paths produce hovering state in the middle. To sum up, the algorithm PESSA proposed in this paper can well plan the flight path that meets the requirements. It can be seen from the map that the height of the flight path is within the prescribed range, and it's up and down frequencies are smooth, which is a path that can be realized.

The convergence curves of the three algorithms (PESSA, PSO and SSA) in Case 1 and 2 are illustrated in Fig. 15. Through the iteration curve, it can be clearly seen that PESSA algorithm has been superior to the other two algorithms within 300 iterations. It can be seen from the two figures that when PSO algorithm and SSA algorithm fall into local optimality, PESSA algorithm can still jump out and continue to converge, which further proves the effectiveness of the new algorithm.

The statistical results with two cases are shown in Table 9. The SSA algorithm had the worst performance of all values. The performance of PSO algorithm is general. The optimal value, mean value and standard deviation of PESSA are small. At the same time, the low standard deviation proves that PESSA can search the optimal path stably. Compared with the other three algorithms, the worst value and mean value of simulation results of the proposed algorithm are smaller. Meanwhile, the standard deviation is only 0.077, which proves that PESSA can stably search the optimal path.

To sum up, it can be seen from the above experimental results that it is impossible to obtain satisfactory paths using SSA and PSO algorithms, especially in complex flight environments.

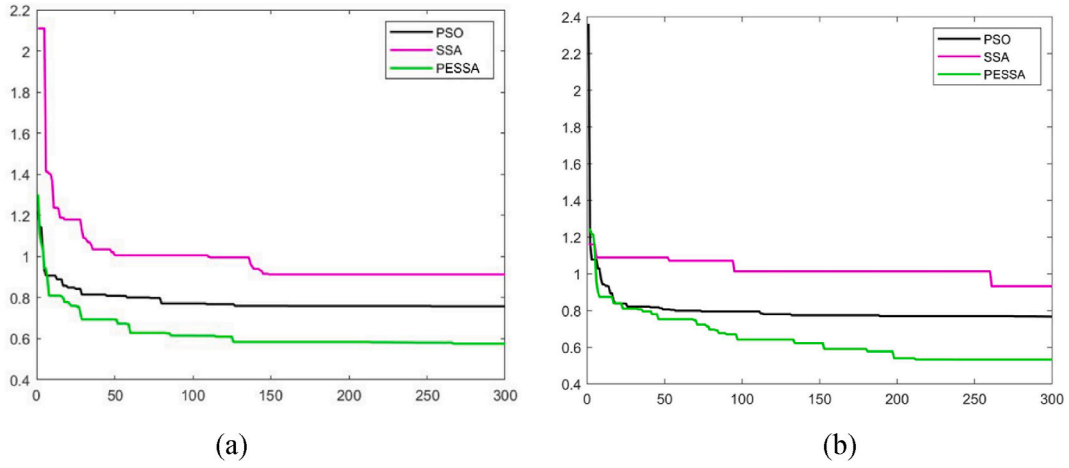

Fig. 15. The convergence curves of 3 algorithms in 3D. (a) Case 1. (b) Case 2.

7. Conclusion and prospect

In this paper, in order to solve the problem of UAV path planning in complex dangerous area, a parallel particle swarm optimization (PSO) and enhanced sparrow search algorithm (ESSA) named PESSA algorithm is presented. At first, we improve the SSA algorithm in three parts. For producers, we change the random jump of their position to random move, which increases the stability of the search range and the ability of global search. For scroungers, we only retain their ability to follow producers and plunder food. For the sparrows who perceive the danger, when it is in the current optimal position, we let the sparrows randomly jump to a random position between the global optimal position and the worst position, and make it walk randomly to get close to other sparrows. Second, PESSA algorithm is run in parallel by PSO algorithm and ESSA algorithm, and the better one is the final result. We also introduce the elite reverse search strategy to yields the optimum diversity. Meanwhile, the new algorithm is verified by 10 basic functions, compared with 6 basic functions and 6 mixed functions respectively. Finally, we apply the new algorithm to the four scenarios, compared it with the basic PSO and SSA algorithm. Finally, we can clearly see the experimental results that PESSA algorithm can successfully plan an effective and safe track, and compared with PSO and SSA algorithm, the results of PESSA algorithm are obviously more convincing.

The scope of this study is limited to 3D global path planning. In future, the combination of global planning and local planning is the research trend. Local planning will require higher efficiency of the algorithm, while its applicable algorithm needs higher speed and accuracy.

Author contribution statement

Ziwei Wang: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Guangkai Sun: Conceived and designed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data.

Kangpeng Zhou: Conceived and designed the experiments; Wrote the paper.

Lianqing Zhu: Contributed reagents, materials, analysis tools or data.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data availability statement

Data included in article/supplementary material/referenced in article.

Declaration of interest's statement

The authors declare no conflict of interest.

References

[1] M. Silvagni, A. Tonoli, E. Zenerino, M. Chiaberge, Multipurpose UAV for Search and Rescue Operations in Mountain Avalanche Events Avalanche Events, 2017, p. 5705, https://doi.org/10.1080/19475705.2016.1238852.

[2] A. Cantieri, M. Ferraz, G. Szekir, M.A. Teixeira, J. Lima, A.S. Oliveira, M.A. Wehrmeister, Cooperative UAV – UGV autonomous power pylon inspection : an investigation of cooperative outdoor, Sensors 20 (2020) 6384, https://doi.org/10.3390/s20216384.

[3] F. Duchon, A. Babinec, M. Kajan, P. Beno, M. Florek, T. Fico, L. Jurisica, Path planning with modified A star algorithm for a mobile robot, Procedia Eng. 96 (2014) 59–69, https://doi.org/10.1016/j.proeng.2014.12.098.

[4] Y. Chen, S. Shen, T. Chen, R. Yang, Path optimization study for vehicles evacuation based on Dijkstra algorithm, Procedia Eng. 71 (2014) 159–165, https://doi.org/10.1016/j.proeng.2014.04.023.

[5] F. Bounini, D. Gingras, H. Pollart, D. Gruyer, Modified artificial potential field method for online path planning applications, IEEE Intell. Veh. Symp. Proc. (2017) 180–185, https://doi.org/10.1109/IVS.2017.7995717.

[6] Y.V. Pehlivanoglu, A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV, Aero. Sci. Technol. 16 (2012) 47–55.

[7] X. Zhang, H. Duan, An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning, Appl. Soft Comput. J. 26 (2015) 270–284, https://doi.org/10.1016/j.asoc.2014.09.046.

[8] D. Zhang, H. Duan, Social-class pigeon-inspired optimization and time stamp segmentation for multi-UAV cooperative path planning, Neurocomputing 313 (2018) 229–246, https://doi.org/10.1016/j.neucom.2018.06.032.

[9] Y. Liu, X. Zhang, X. Guan, D. Delahaye, Adaptive sensitivity decision based path planning algorithm for unmanned aerial vehicle with improved particle swarm optimization, Aero. Sci. Technol. 58 (2016) 92–102, https://doi.org/10.1016/j.ast.2016.08.017.

[10] J. Xue, B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, Syst. Sci. Control Eng. 8 (2020) 22–34, https://doi.org/10.1080/21642583.2019.1708830.

[11] Y. Zhao, L. Yan, Y. Chen, J. Dai, Y. Liu, Robust and efficient trajectory replanning based on guiding path for quadrotor fast autonomous flight, Rem. Sens. 13 (2021) 1–26, https://doi.org/10.3390/rs13050972.

[12] G. Tang, C. Tang, H. Zhou, C. Claramunt, S. Men, R-dfs, A coverage path planning approach based on region optimal decomposition, Rem. Sens. 13 (2021), https://doi.org/10.3390/rs13081525.

[13] M. Mangeruga, A. Casavola, F. Pupo, F. Bruno, An underwater pathfinding algorithm for optimised planning of survey dives, Rem. Sens. 12 (2020) 1–17, https://doi.org/10.3390/rs12233974.

[14] T. Koch, M. Körner, F. Fraundorfer, Automatic and semantically-aware 3D UAV flight planning for image-based 3D reconstruction, Rem. Sens. 11 (2019) 1–29, https://doi.org/10.3390/rs11131550.
[15] D.R. Shier, On algorithms for finding the k shortest paths in a network, Networks 9 (1979) 195–214.
[16] Y. Fu, M. Ding, C. Zhou, H. Hu, Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization, IEEE Trans. Syst. Man, Cybern. Syst. 43 (2013) 1451–1465.
[17] Z. Wang, L. Liu, T. Long, C. Yu, J. Kou, Enhanced sparse A search for UAV path planning using dubins path estimation, in: Proc. 33rd Chinese Control Conf., IEEE, 2014, pp. 738–742.
[18] H. Niu, Y. Lu, A. Savvaris, A. Tsourdos, An energy-efficient path planning algorithm for unmanned surface vehicles, Ocean Eng. 161 (2018) 308–321.
[19] C. Zammit, E. V Kampen, Comparison between A* and RRT algorithms for UAV path planning, in: 2018 AIAA Guidance, Navigation, and Control Conference, 2018.
[20] F.R.L. Serrano, E. Rubio, F.A.G. Morote, M.A. Abellán, M.I.P. Córdoba, F.G. Saucedo, E.M. García, J.M.S. García, J.S. Innerarity, L.C. Lucas, International journal of applied earth observations and geoinformation, Int. J. Appl. Earth Obs. Geoinf. 113 (2022), 103014.
[21] Y. Xu, J. Li, F. Zhang, A UAV-based forest fire patrol path planning strategy, Forests 13 (2022) 1952.
[22] X. Zhang, H. Duan, An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning, Appl. Soft Comput. 26 (2015) 270–284.
[23] D. Zhang, H. Duan, Social-class pigeon-inspired optimization and time stamp segmentation for multi-UAV cooperative path planning, Neurocomputing 313 (2018) 229–246.
[24] X. Yu, C. Li, J. Zhou, A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios, Knowl. Base Syst. 204 (2020), 106209.
[25] M.D. Phung, Q.P. Ha, Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization, Appl. Soft Comput. 107 (2021), 107376.
[26] R.K. Dewangan, A. Shukla, W.W. Godfrey, Three dimensional path planning using Grey wolf optimizer for UAVs, Appl. Intell. 49 (2019) 2201–2217, https://doi.org/10.1007/s10489-018-1384-y.
[27] C. Huang, J. Fei, W. Deng, A novel route planning method of fixed-wing unmanned aerial vehicle based on improved QPSO, IEEE Access 8 (2020) 65071–65084.
[28] J. Sánchez-García, D.G. Reina, S.L. Toral, A distributed PSO-based exploration algorithm for a UAV network assisting a disaster scenario, Future Generat. Comput. Syst. 90 (2019) 129–148.
[29] Y. Chen, D. Pi, Y. Xu, Neighborhood global learning based flower pollination algorithm and its application to unmanned aerial vehicle path planning, Expert Syst. Appl. 170 (2021), 114505.
[30] A. Machmudah, M. Shanmugavel, S. Parman, T.S.A. Manan, D. Dutykh, S. Beddu, A. Rajabi, Flight trajectories optimization of fixed-wing UAV by bank-turn mechanism, Drones 6 (2022) 69.
[31] R. Thangaraj, M. Pant, A. Abraham, P. Bouvry, Particle swarm optimization: hybridization perspectives and experimental illustrations, Appl. Math. Comput. 217 (2011) 5208–5226.
[32] F.J. Rodriguez, C. García-Martinez, M. Lozano, Hybrid metaheuristics based on evolutionary algorithms and simulated annealing: taxonomy, comparison, and synergy test, IEEE Trans. Evol. Comput. 16 (2012) 787–800, https://doi.org/10.1109/TEVC.2012.2182773.
[33] D. Fodorean, L. Idoumghar, M. Brévilliers, P. Minciunescu, C. Irimia, Hybrid differential evolution algorithm employed for the optimum design of a high-speed PMSM used for EV propulsion, IEEE Trans. Ind. Electron. 64 (2017) 9824–9833.
[34] Y. Y. Han, Q. Chen, N. Pan, X. Guo, An, unmanned aerial vehicle 3D trajectory planning based on background of complex industrial product warehouse inventory Sensor. Mater. 34 (2022) 3255–3269.
[35] P.K. Das, H.S. Behera, B.K. Panigrahi, A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning, Swarm Evol. Comput. 28 (2016) 14–28.
[36] W. He, X. Qi, L. Liu, A novel hybrid particle swarm optimization for multi-UAV cooperate path planning, Appl. Intell. 51 (2021) 7350–7364.
[37] X. Wang, H. Zhao, T. Han, H. Zhou, C. Li, A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem, Appl. Soft Comput. 78 (2019) 240–260.
[38] Q. Liu, Y. Zhang, M. Li, Z. Zhang, N. Cao, J. Shang, Multi-UAV path planning based on fusion of sparrow search algorithm and improved bioinspired neural network, IEEE Access 9 (2021) 124670–124681.
[39] V. Roberge, M. Tarbouchi, G. Labonté, Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning, IEEE Trans. Ind. Inf. 9 (2012) 132–141.
[40] N. Sariff, N. Buniyamin, An overview of autonomous mobile robot path planning algorithms, in: 2006 4th Student Conf. Res. Dev., IEEE, 2006, pp. 183–188.
[41] Z. Sun, J. Wu, J. Yang, Y. Huang, C. Li, D. Li, Path planning for GEO-UAV bistatic SAR using constrained adaptive multiobjective differential evolution, IEEE Trans. Geosci. Rem. Sens. 54 (2016) 6444–6457.
[42] I.K. Nikolos, K.P. Valavanis, N.C. Tsourveloudis, A.N. Kostaras, Evolutionary algorithm based offline/online path planner for UAV navigation, IEEE Trans. Syst. Man, Cybern. Part B. 33 (2003) 898–912.
[43] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: 1998 IEEE Int. Conf. Evol. Comput. Proceedings. IEEE World Congr. Comput. Intell. (Cat. No. 98TH8360), IEEE, 1998, pp. 69–73.
[44] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Software 69 (2014) 46–61.
[45] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, Knowl. Base Syst. 96 (2016) 120–133.
[46] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2016) 495–513.
[47] S. Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, Knowl. Base Syst. 89 (2015) 228–249.
[48] C. Zhang, S. Ding, A stochastic configuration network based on chaotic sparrow search algorithm, Knowl. Base Syst. 220 (2021), 106924.
[49] G. Liu, C. Shu, Z. Liang, B. Peng, L. Cheng, A modified sparrow search algorithm with application in 3d route planning for UAV, Sensors 21 (2021) 1224.
[50] J. Zhang, K. Xia, Z. He, Z. Yin, S. Wang, Semi-supervised ensemble classifier with improved sparrow search algorithm and its application in pulmonary nodule detection, Math. Probl Eng. 2021 (2021) 1–18.
[51] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (2015) 11–24.
[52] Q. Cui, Q. Li, G. Li, Z. Li, X. Han, H.P. Lee, Y. Liang, B. Wang, J. Jiang, C. Wu, Globally-optimal prediction-based adaptive mutation particle swarm optimization, Inf. Sci. (2017) 186–217, https://doi.org/10.1016/j.ins.2017.07.038, 418–419.
[53] K. Chen, F. Zhou, Y. Wang, L. Yin, An ameliorated particle swarm optimizer for solving numerical optimization problems, Appl. Soft Comput. 73 (2018) 482–496.