Research Article

Path Planning of Unmanned Autonomous Helicopter Based on Human-Computer Hybrid Augmented Intelligence

Zengliang Han [iD],[1] Mou Chen [iD],[1,2] Tongle Zhou [iD],[1] Zhiqiang Nie,[2] and Qingxian Wu[1]

[1]*College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China*
[2]*Science and Technology on Electro-optic Control Laboratory, Luoyang Institute of Electro-Optical Equipment of Avic, Luoyang 471000, China*

Correspondence should be addressed to Mou Chen; mouchen@nuaa.edu.cn

Unmanned autonomous helicopter (UAH) path planning problem is an important component of the UAH mission planning system. The performance of the automatic path planner determines the quality of the UAH flight path. Aiming to produce a high-quality flight path, a path planning system is designed based on human-computer hybrid augmented intelligence framework for the UAH in this paper. Firstly, an improved artificial bee colony (I-ABC) algorithm is proposed based on the dynamic evaluation selection strategy and the complex optimization method. In the I-ABC algorithm, the following way of onlooker bees and the update strategy of nectar source are optimized to accelerate the convergence rate and retain the exploration ability of the population. In addition, a space clipping operation is proposed based on the attention mechanism for constructing a new spatial search area. The search time can be further reduced by the space clipping operation under the path planning result within acceptable changes. Moreover, the entire optimization process and results can be fed back to the knowledge database by the human-computer hybrid augmented intelligence framework to guide subsequent path planning issues. Finally, the simulation results confirm that a feasible and effective flight path can be quickly generated by the UAH path planning system based on human-computer hybrid augmented intelligence.

1. Introduction

In recent years, the developments in automated and unmanned flight technologies have been of high interest to many military organizations throughout the world [1, 2]. Due to their outstanding capability to work in remote or hazardous situations, the unmanned autonomous helicopters (UAHs) have been widely used for many civil and military applications [3]. Path planning is the key to ensure the successful completion of UAH missions [4]. This particular issue is classified as an NP-Hard optimization problem in the high dimension [5]. In the past few years, a variety of methods have been proposed for the path planning problem of UAHs [6, 7]. Traditional methods, such as Voronoi diagram algorithm [8], A∗ algorithm [9], artificial potential field algorithm [10], and rapidly exploring random tree algorithm [11] were proposed to process the path planning problem. However, most of these

methods exist the long search times and precociousness issues when the UAH flight path has complex constraints.

Recently, with the development and application of metaheuristic algorithms, more and more swarm intelligence algorithms have been widely applied to the complex optimization problems [12, 13]. As a kind of nature-inspired algorithms, metaheuristic algorithms were originated from imitating biological interactive behaviors or physical phenomena [14], such as the particle swarm optimization (PSO) algorithm [15], genetic algorithm (GA) [16], and differential evolution (DE) algorithm [17]. In recent years, lots of studies have been investigated on improving and modifying the existing metaheuristic algorithms, such as wavelet mutation strategy differential evolution (WMSDE) algorithm [18], multistrategy-based coevolutionary DE (MPPCEDE) algorithm [19], and multiple strategies quantum-inspired differential evolution (MSIQDE) algorithm [20].

Based on the existing research, recently, series of meta-heuristic algorithms had been used to solve the UAH path planning problem [21]. In [22], a modification of wolf pack search (WPS) algorithm was presented for path planning of the unmanned aerial vehicle (UAV). Similarly, a grey wolf optimizer (GWO) was applied for path planning problem in the battle field in [23]. In [24], an improved bat algorithm (I-BA) was proposed to process the 3D path planning problems for unmanned combat air vehicles (UCAV). In [25], the multiverse optimizer (MVO) was used for resolving the 2D path planning problem for the UAV. In [26], the discrete particle swarm optimization (PSO) technique was enhanced and used to the path planning problem for surface inspection based on UAV vision.

In 2005, a novel global optimization algorithm based on swarm intelligence called artificial bee colony (ABC) algorithm was proposed by Karaboga [27]. Compared with other intelligent algorithms, the principle of the artificial bee colony (ABC) algorithm makes the algorithm has the advantages of implementation and flexibility. With the continuous in-depth study of the ABC algorithm, it has been widely applied to resolve many engineering application and control problems, such as system identification, task assignment, and path planning. The ABC algorithm has a superior effect in solving path planning problems. However, it is easy to fall into a local optimal solution during the ABC algorithm iteration. To this end, many improved ABC algorithms have been proposed, such as Rosenbrocks rotational direction strategy [28], Boltzmann selection strategy [29], DE-ABC (differential evolution-ABC) [30], PSO-ABC (particle swarm optimization-ABC) [31], and QE-ABC (quantum evolutionary-ABC) [32]. Although those modified ABC algorithms have been widely used in various path planning problems, they were limited by the inherent limitations of the heuristics. Thus, the path planning results need to be further optimized and improved.

Due to the open-ended nature of the path planning problems, no matter how intelligent computers are, they still cannot completely replace humans. Unmanned systems still need human supervision to take advantage of them. Therefore, it is indispensable to utilize human cognitive capabilities or human-like cognitive models into optimization layer to exploit a new path planning system for the UAH [33]. In recent years, with the development of artificial intelligence, researchers have become interested in a new kind of technique called hybrid-augmented intelligence [34]. From the actual situation, "hybrid-augmented intelligence" has been widely used in the fields of industrial development decision-making, online intelligent learning, medical health care, and human-computer codriving [35–37]. The idea of human-computer integration provides a new research approach for the path planning of the UAH. In this paper, the human-computer hybrid augmented intelligence is applied to the path planning system of the UAH. Combining the human intelligent experience to compensate for the shortcomings of the heuristic algorithm, a feasible flight path can be calculated by the improved algorithm. Furthermore, the search time of the flight path can also be further reduced while ensuring that it meets all constraints.

To sum up, the ABC algorithm has great preponderance in solving the multiconstraints optimization problem. Similarly, the concept of hybrid augmented intelligence relies on the characteristics of human-computer integration, enabling the system to handle the complex and difficult problems. To overcome the defects of the traditional ABC algorithm and improve the quality of the flight path, a path planning system is proposed based on human-computer hybrid augmented intelligence framework for the UAH in this paper. The innovations in this paper are summarized as follows.

(i) A new path planning system based on human-computer hybrid augmented intelligence framework for UAH is designed to improve the rapidity and optimality of the automatic path planner. The used human-computer hybrid augmented intelligence framework combines the advantages of human and computer for planning a high-quality flight path

(ii) An I-ABC algorithm is proposed based on the dynamic evaluation selection strategy and the complex optimization method for accelerating the convergence rate and maintaining the exploration ability of the population

(iii) A space clipping method is designed based on the attention mechanism for reconstructing the spatial search area. The space clipping operation further reduces the subsequent search time of the flight path for the path planning system

(iv) The simulation studies are executed to comprehensively prove the effectiveness of path planning system based on human-computer hybrid augmented intelligence framework for UAH by various air combat environment models

The remainder of the paper is organized as follows. Section 2 demonstrates the problem statement. Section 3 investigates the framework of human-computer hybrid augmented intelligence for path planning. Section 4 investigates an improved ABC algorithm. Section 5 investigates the space clipping method based on human optimizing attention mechanism. Section 6 provides the feasibility and effectiveness of the system by simulation experiment. Finally, the concluding remarks are involved in Section 7.

2. Problem Statement

As shown in Figure 1, the path planning is the critical part of the UAH mission assignment system, and it is used to find the optimal flight path under the constraints such as weather threat, terrain threat, radars threat, and missiles threat. In this paper, we mainly study the 2D path planning of the UAH which execute the mission at the same flight altitude. We assume that the UAH maintains constant flight speed and altitude during its mission.

2.1. Modeling of UAH Path Planning. In this paper, the starting point and the terminal point are defined as S and T.
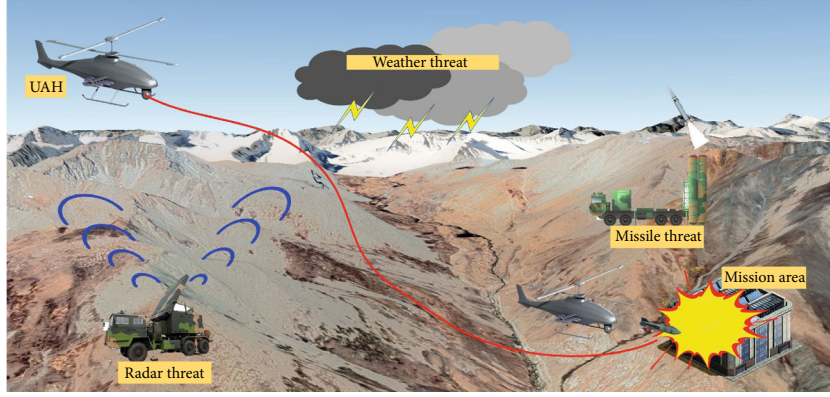
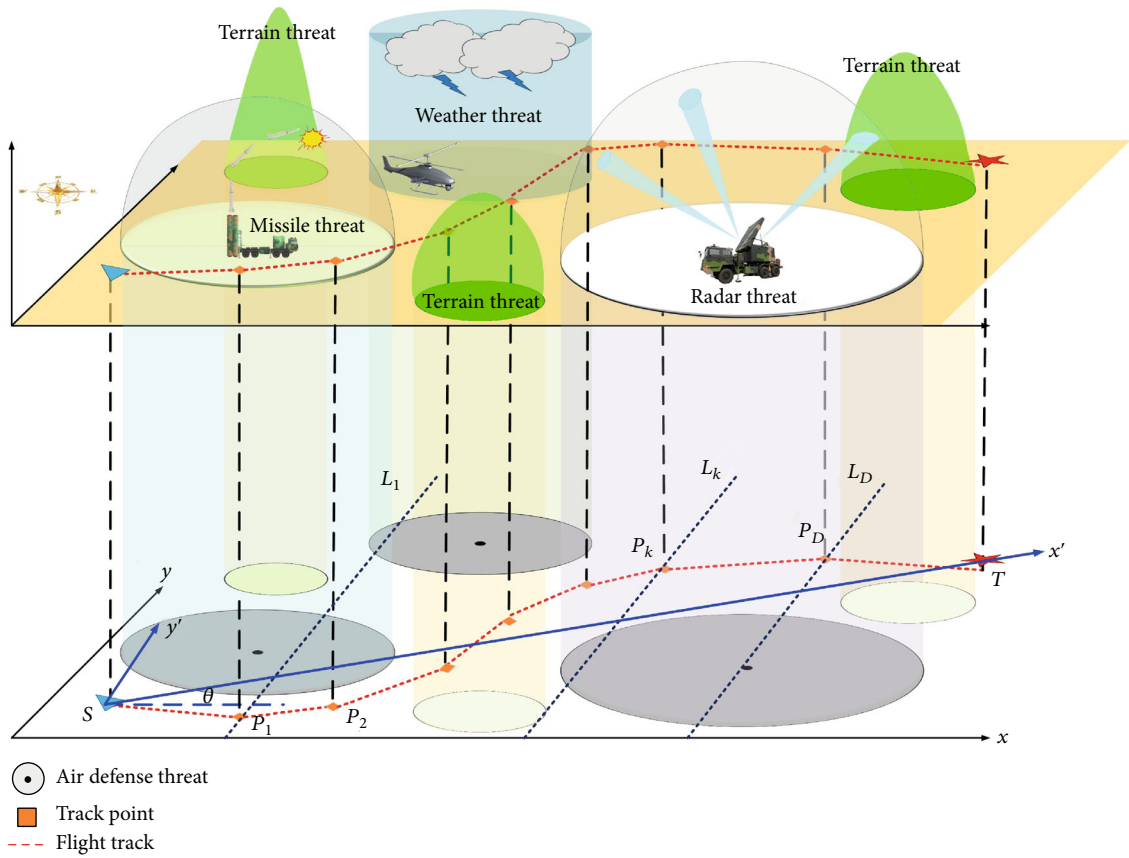FIGURE 1: Schematic of UAH path planning mission.

FIGURE 2: Schematic of 2D UAH battlefield model.

According to the method given in [38] which is shown in Figure 2, we divided ST into $(D+1)$ equal parts by D vertical lines $\{L_k, k = 1, 2, \cdots, D\}$, which intersect ST at each segment point. A flight path is formed by connecting the series of points on the new axis. To simplify the calculations, the threat regions covered in this paper are assumed to be circular projections. Therefore, the path planning problem can be converted to the route point coordinate optimization problem.

To simplify the calculation process, a new coordinate frame need to be established. In this section, the segment S

T is considered as the new x-axis [39]. Each path point on the original combat field gets transferred in the new axes as defined in (1).

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}, \tag{1}$$

where θ represents the angle between the x-axis of the inertial frame and the x' of the body-fixed frame direction, (x_s, y_s) correspond to the start point on the body-fixed frame.

The damage probability is determined by the distance between the UAH and the threat center. Assuming that the new coordinates of the threat center point are (x_i, y_i) and the threat radius are r_i, then the threat area can be expressed as [40]

$$(x - x_i)^2 + (y - y_i)^2 = r_i^2. \tag{2}$$

2.2. Cost Function and Performance Constraints. In this paper, the performance evaluation index of the UAH flight path is composed of the threat cost J_{threat}^i and the fuel cost J_{fule}^i. The total cost and the each cost are described as follows [41].

$$J = \sum_{i=1}^{D+1} \left[\tau J_{threat}^i + (1 - \tau) J_{fule}^i \right], \tag{3}$$

where τ is the weighting parameter between 0 and 1.

The flight cost from the point along L_i to the one along L_{i+1} is calculated at five points (as shown in Figure 3). If the flight path shown above falls into a threat region, the J_{threat}^i is calculated as follows.

$$J_{threat}^{L_i \to L_{i+1}} = \frac{l_i}{5} \cdot \sum_{k=1}^{N_t} \left[s_k \cdot \left(\frac{1}{d_{0.1,i}^k} + \frac{1}{d_{0.3,i}^k} + \frac{1}{d_{0.5,i}^k} + \frac{1}{d_{0.7,i}^k} + \frac{1}{d_{0.9,i}^k} \right) \right], \tag{4}$$

where N_t is the total number of threats, l_i is the length of the ith subtrack, $d_{0.1,i}^k$ stands for the distance between the 1/10 point on the path and the kth threat center, and s_k is regarded as the grade of the kth threat.

Assume that the UAH is moving at a constant speed, the J_{fule}^i is calculated as follows.

$$J_{fule}^i = \sum_{i=1}^{D} \frac{l_i}{v} \eta, \tag{5}$$

where l_i is the length of the ith subtrack, v is the speed of the UAH, and η is the fuel consumption per unit time of the UAH.

Considering the actual flight situation for the UAH, the yawing angle constraints are introduced as follows.

$$\varphi_j = \left| \arctan \left(\frac{y_{j+1} - y_j}{x_{j+1} - x_j} \right) \right| \leq \varphi^{max}, \tag{6}$$

where φ_j is the yaw angle of the jth node and φ^{max} is the maximum yawing angle.

3. Human-Computer Hybrid Augmented Intelligence Framework for the Path Planning

Introducing human intelligence to the loop of the path planning system can can tackle the fuzzy and uncertain problems
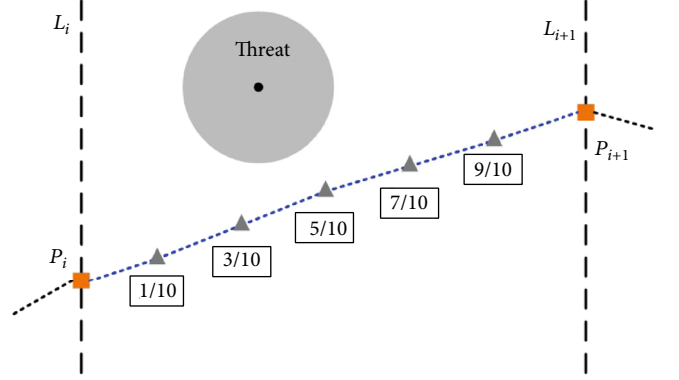

FIGURE 3: Schematic diagram of flight cost computation.

[42]. Hence, human and computer cooperate with each other to form a double-sided information exchange and control. The $'1 + 1 > 2'$ hybrid augmented intelligence can be realized by conforming human cognitive ability, computer computing, and storage capacities [43].

In this paper, a path planning system based on the human-computer hybrid augmented intelligence framework is designed. The new system integrates the artificial bee colony (ABC) algorithm and human intelligence. Through the human-computer cooperation, the intelligent process ability of the path planning system will be enhanced, and the flight path of the UAH will be efficiently improved. The framework of the path planning system based on the human-computer hybrid augmented intelligence is shown in Figure 4.

The core of the path planning system for the UAH is mainly composed of the following two parts.

3.1. Computer Processing Module. When the new combat mission is entered, the computer will calculate an original flight path by the ABC algorithm according to the relevant constraints [44]. This process will always be guided by the experience of the expert knowledge database [45]. The original flight path needs to be evaluated to determined if it becomes the final result. However, it is initially difficult for the traditional ABC algorithm to obtain a high confidence flight path for the UAH path planning system.

3.2. Human Optimizing Module. When the original flight path cannot meet the needs of the actual combat, the human-computer hybrid augmented intelligence system will decide whether the result needs human adjustment or human intervention [46]. In this paper, the human optimization module is mainly divided into the following two parts.

(i) Algorithm optimization layer

The ABC algorithm has a good effect in solving path planning problems; however, it is easy to fall into a local optimal solution during algorithm iteration. In the algorithm optimization layer, the traditional ABC algorithm is improved by the dynamic evaluation selection strategy and the complex optimization method. The improved ABC (I-ABC) algorithm raises the optimisation efficiency, avoids
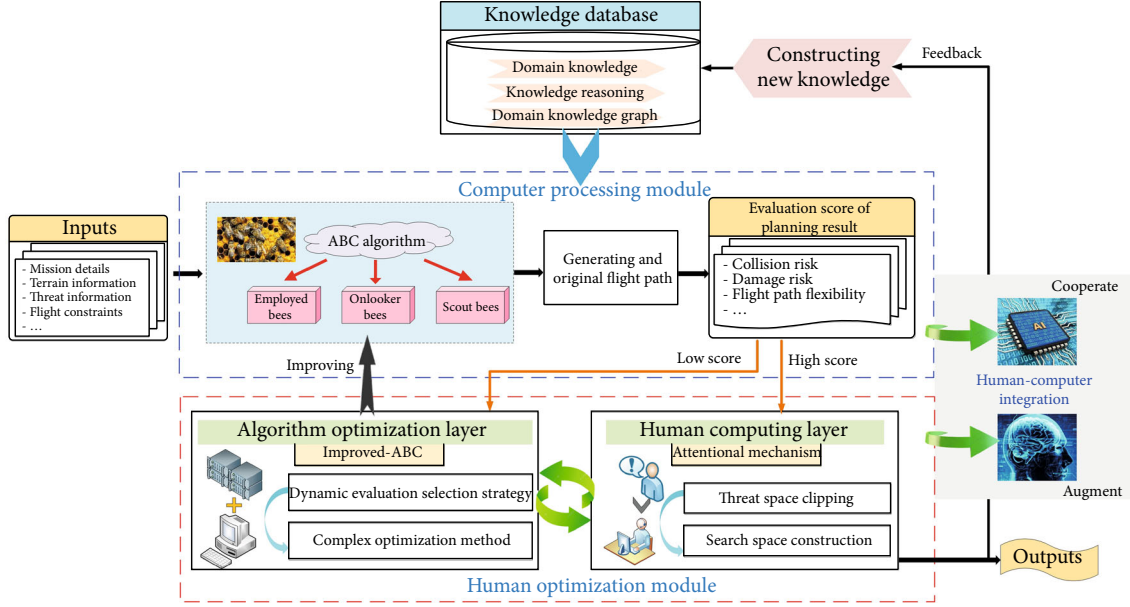
FIGURE 4: Framework of the human-computer hybrid augmented intelligence.

the algorithmic precocity, and plans a high confidence flight path according to the actual task requirements.

(ii) Human computing layer

The path planning based on the I-ABC algorithm can basically meet the mission demands. However, the time of the flight path planning can still be further reduced to improve the efficiency of the planning system. In the human computing layer, the efficiency of the planning system can be further improved by space clipping operation based on the attention mechanism. This step can be used as empirical knowledge to feed back into the expert knowledge base to guide and monitor the entire path planning system, thus enabling human-computer hybrid-augmented features.

4. UAH Path Planning Based on I-ABC Algorithm

The ABC algorithm is the most important component of the path planning system for the UAH. The rationality and efficiency of the ABC algorithm in dealing with complex problems will directly affect the results of the path planning system for the UAH. Although the ABC algorithm is widely used at present, it still exposes many problems when it faces complex multiconstraints optimization problems.

In this section, the traditional ABC algorithm will be improved in the algorithm optimization layer to optimize the path planning of the UHA. In the algorithm optimization layer, there are mainly two optimization measures as follows.

(i) First, in order to prevent the ABC algorithm premature convergence problem, a dynamic evaluation selection strategy is proposed to optimize the follow way of on-looker bees for improving the searching efficiency

(ii) Second, in order to improve the quality of the flight path, the complex method is used to guide the optimization of the swarm. The nectar source is updated during each iteration, the search efficiency of the nectar source is improved, and the algorithm convergence is accelerated

The specific process of the I-ABC algorithm is as follows.

4.1. Initialization. There are three kinds of bees, i.e., employed bees, onlooker bees, and scout bees in the ABC algorithm [27]. Accordingly, the triple search capability of the ABC includes of three search phases: employed bee stage, onlooker stage, and scout stage [28]. The position of the nectar source (initial 2D track) is represented by a $D \times NP$-dimensional matrix $E = \{e_{ij}\}$, the vector in the ith row of the matrix is represented as

$$X_i = \left(x_{i1}, \cdots, x_{ij}, \cdots, x_{iD}\right), \tag{7}$$

where $1 \leq i \leq N_p$, $2 \leq j \leq (D-1)$, and $x_{i1} = x_{iD} = 0$.

All the NP employed bees need to be randomly initialized by equation (8) [27]. In other words, each track point is randomly generated within a specified range in the feasible solution space.

$$x_{ij} = x_i^{\min} + \text{rand } (0,1) \cdot \left(x_i^{\max} - x_i^{\min}\right), \tag{8}$$

where $x_i^{\min}$ and $x_i^{\max}$ are the constraints of the ith parameter, rand $(0,1)$ is a random number range from 0 to 1.

Then, the corresponding fitness is calculated as

$$\text{fitness}(i) = \begin{cases} \dfrac{1}{[1 + obj(i)]}, & obj(i) \geq 0, \\[2mm] 1 + |obj(i)|, & obj(i) < 0, \end{cases} \tag{9}$$
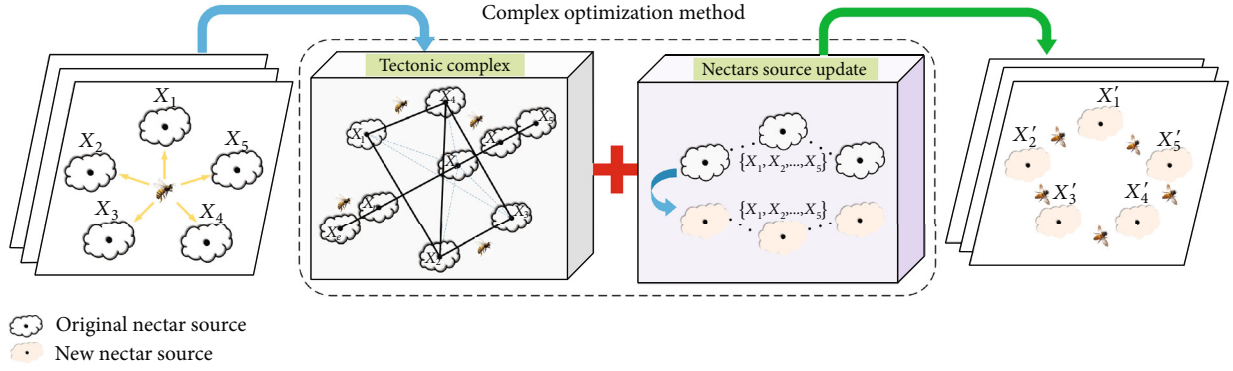
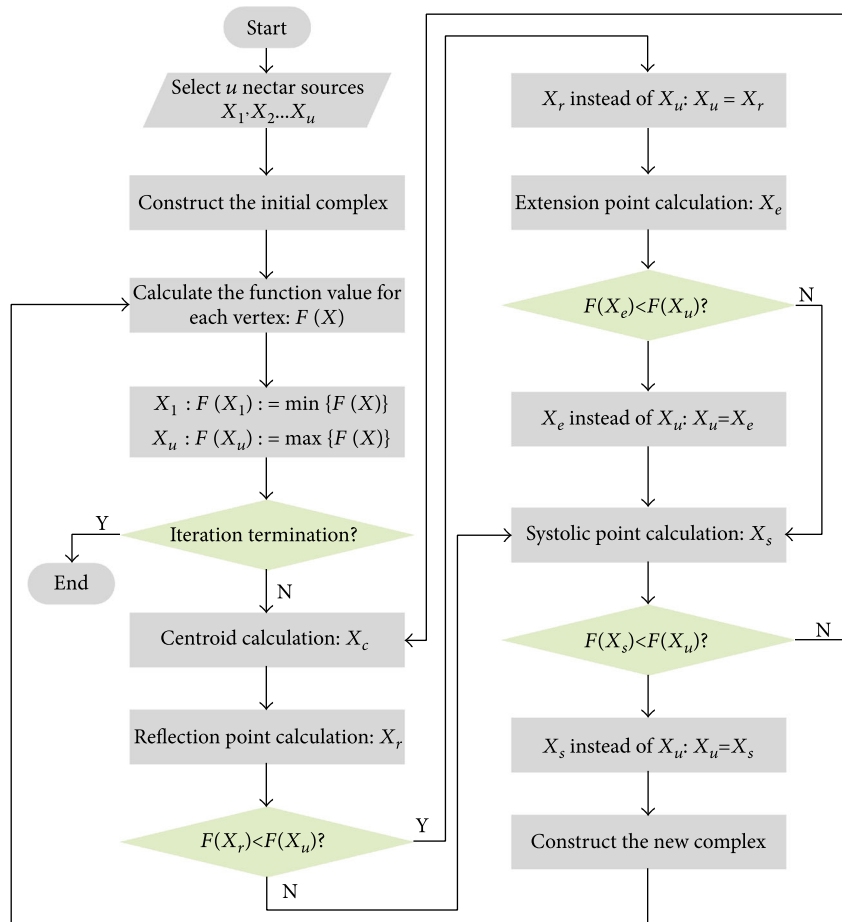FIGURE 5: Schematic model of the complex method for five nectar sources.

FIGURE 6: Flow diagram of the complex optimization method.

where $\mathrm{obj}(i)$ represents the objective function value with respect to X_i.

4.2. Employed Bee Phase. Each employed bee is related with only one nectar source. The employed bee finds a new candidate solution through changing the nectar source position in its memory based on the local information. In each iteration cycle, the employed bee adopts (10) to

search for the better nectar source V_{ij} around the current nectar source [27].

$$v_{ij} = x_{ij} + \mathrm{rand}\,(1,-1) \cdot \left(x_{ij} - x_{kj}\right), \qquad (10)$$

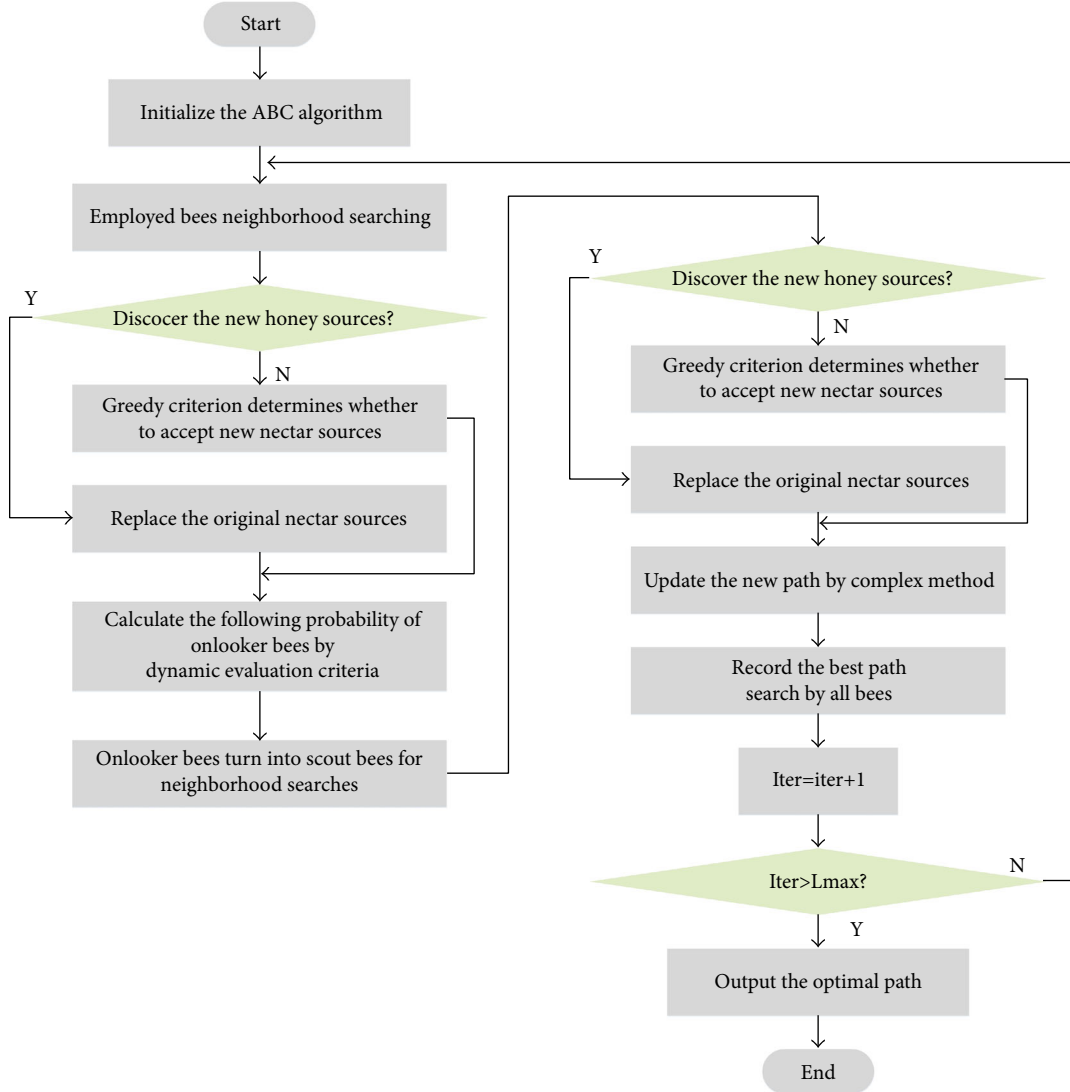where $i = 1, 2, \cdots, NP$, $k \in \{1, 2, \cdots, NP\}$, $j \in \{1, 2, \cdots, D\}$,

FIGURE 7: Flow diagram of the I-ABC.

k is a random integer different from i, j is a random integer, and rand $(1,-1)$ is a random number range from -1 to 1.

During the neighborhood search process, when the location of the nectar source searched by the employed bees exceeds the search boundary constraint, the bee is selected according to (11).

$$x_{ij} = \begin{cases} x_j^{\min} + \partial\left(x_j^{\max} - x_j^{\min}\right), & x_{ij} \leq x_j^{\min}, \\ x_j^{\min} - \partial\left(x_j^{\max} - x_j^{\min}\right), & x_{ij} > x_j^{\min}, \end{cases} \tag{11}$$
$$\partial \in \text{rand}\,(0,1)$$

where $x_j^{\min}$ and $x_j^{\max}$ are the constraints of the jth parameter.

4.3. Onlooker Bee Phase. In the onlooker bee phase, the amounts and positions of their nectar sources will be transmitted by the employed bees to onlooker bees. Then, the onlooker bee evaluates all the information of nectar sources transmitted by employed bees, and chooses one

nectar source site according to the probability value p_i formulated as [27]

$$p_i = \frac{\text{fitness}(i)}{\sum_{j=1}^{NP}\text{fitness}(i)}, \tag{12}$$

where NP represents the number of bees.

However, the traditional following selection method overly focus on the development of excellent nectar sources. Ignoring other potential nectar sources will cause the algorithm's global search capability to decline and reduce the algorithms solution efficiency. For solving this problem, we propose a dynamic evaluation selection strategy instead of the traditional method to follow the location of the nectar sources.

(1) Defining the nectar source dynamic evaluation integral value

In this section, the dynamic evaluation integral values of nectar sources are defined as $\Psi_1(i)$ and $\Psi_2(i)$. When the

nectar source i maintains unchanged, $\Psi_1(i)$ is calculated according to equation (13), and $\Psi_2(i)$ is 0. When the nectar source position i is replaced by a better one, $\Psi_2(i)$ is calculated according to equation (14), and $\Psi_1(i)$ is 0.

$$\Psi_1(i) = \begin{cases} \Psi_1(i) + \text{step}, & \Psi_1(i) < \text{limit}_d, \\ \text{limit}_d, & \Psi_1(i) \geq \text{limit}_d, \end{cases} \quad (13)$$

$$\Psi_2(i) = \begin{cases} \Psi_2(i) + \text{step}, & \Psi_2(i) < \text{limit}_d, \\ \text{limit}_d, & \Psi_2(i) \geq \text{limit}_d, \end{cases} \quad (14)$$

where $\Psi_1(i)$ is the number of searching near the nectar source i but the position has not changed, $\Psi_2(i)$ is the number of searching near the nectar source i and the position has changed, and limit_d is the dynamic update limit parameter, step is the search unit length.

(2) Building the dynamic evaluation function

Considering that there will be more greater possibility of finding excellent nectar sources near the continuously optimized nectar sources. The dynamic evaluation function $F(i)$ is constructed according to the dynamic evaluation criteria which is written as

$$F(i) = \begin{cases} \zeta \left(1 - \dfrac{\Psi_1(i)}{\text{limit}_d} \right), & \Psi_1(i) \neq 0, \\ \zeta \left(1 + \dfrac{\Psi_2(i)}{\text{limit}_d} \right), & \Psi_1(i) = 0, \end{cases} \quad (15)$$

where ζ is the base score of the nectar source.

(i) When $\Psi_1(i) = 0$, this nectar source may be trapped in a local optimum. Onlooker bees should try to avoid selecting this type of nectar source for further development

(ii) When $\Psi_1(i) \neq 0$, this nectar source has been continuously optimized multiple times. Onlooker bees need to try to select this type of nectar source for neighborhood search

The evaluation integral value of the nectar source is calculated by the evaluation function $F(i)$. The basic score of the nectar source is ζ, and the selection possibility of all individuals is guaranteed. The scores of nectar sources with continuously changing positions are more than ζ, which has a greater possibility of being followed. The optimized selection probability is calculated by (16).

$$p'_i = \frac{F(i)}{\max(F(i))}. \quad (16)$$

4.4. Scout Bee Phase. In each iteration cycle, the exhausted nectar source will be checked by ABC algorithm after all the employed bees and onlookers complete their searches. If the new position X_i is not improved continuously for a certain time, then the corresponding nectar source will be abandoned by the employed bee. At this time, this employed bee will become a scout bee. According to (10), a new nectar source v_{ij} will be generated by the scout bee.

In order to improve the efficiency of the nectar source and the quality of the flight path, a complex optimization method is used to guide the swarm. The complex optimization method is a direct search algorithm for finding constrained optimization problems [47]. In solving nonlinear problems, by virtue of its freedom from the constraints of the research problem and the objective function, it has a wide range of applicability and can be embedded in many other algorithms to guide the problem to an optimal or suboptimal solution.

From the nectar sources before the end of each iteration, u nectars are selected to construct the complex geometry, which is ordered as $(X_1, X_2, \cdots X_u)$ according to the size of the objective function value from best to worst (as shown in Figure 5). The process of the complex optimization method is as follows.

Step 1. Centroid calculation

$$X_c = \frac{1}{u - 1} \sum_{i=1}^{u-1} X_i. \quad (17)$$

Step 2. Reflection point calculation

$$X_r = X_c + \alpha(X_c - X_u), \quad (18)$$

where $\alpha \in (0, 1)$ is the reflection coefficient. If X_r is better than X_u, X_u will be replaced by X_r; otherwise, go to Step 4.

Step 3. Extension point calculation

$$X_e = X_r + \beta(X_r - X_c), \quad (19)$$

where $\beta \in (0, 1)$ is the extension coefficient. If X_e is better than X_u, X_u will be replaced by X_e; otherwise, go to Step 4.

Step 4. Systolic point calculation

$$X_s = X_u + \chi(X_c - X_u), \quad (20)$$

where $\chi \in (0, 1)$ is the systolic coefficient. If X_s is better than X_u, X_u will be replaced by X_s; otherwise, go to Step 1.

In this paper, the purpose of the complex optimization method is to emphasize the "replacement" effect, not the traditional "search for excellence." So when the convergence termination condition is setted, it is only necessary for the algorithm to reach the required number of iterations, which is related to the number of selected nectar sources. The convergence termination is as follows.

$$N_{\text{complex}} = u. \quad (21)$$

The flow diagram of the complex optimization method is shown in Figure 6.

By the complex optimization method, the original u worst nectar sources in the search space are replaced.

```
 1: initialize solution population using (8)
 2: set trial(i) = 1(i = 12, ⋯NP)
 3: for iter = 1 : MCN do
 4:    for item = 1 : SN do
 5:       generate positions for employed bees by (10)
 6:       evaluate and greedily select employed bees
 7:       if better position is found then
 8:          trial(item) ⟵ 1
 9:       else
10:          trial(item) ⟵ trial(item) + 1
11:       end if
12:    end for
13:    calculate the dynamic evaluation probability p′ᵢ by (16)
14:    set item = 1
15:    while item < SN do
16:       generate positions for employed bees by (10)
17:       evaluate and greedily select employed bees
18:       if position is changed then
19:          trial(item) ⟵ 1
20:       else
21:          trial(item) ⟵ trial(item) + 1
22:       end if
23:    end while
24:    if trial(item) ≥ Limit then
25:       initialize the position by (8)
26:    end if
27:    while item < u do
28:       calculate centroid Xc by (17)
29:       calculate reflection point Xr by (18)
30:       if F(Xr) < F(Xu) then
31:          Xu = Xr
32:          calculate extension point Xe by (19)
33:          if F(Xe) < F(Xu) then
34:             Xu = Xe
35:          end if
36:       else
37:          calculate systolic point Xs by (20)
38:          if F(Xs) < F(Xu) then
39:             Xu = Xs
40:             item = item + 1
41:          end if
42:       end if
43:    end while
44:    if trial(item) ≥ Limit then
45:       initialize the position by (8)
46:    end if
47:    record current best solution
48:    iter ⟵ iter + 1
49: end for
50: output global optimal solution
```

ALGORITHM 1: The I-ABC algorithm.

The complex geometry keeps reflecting, extending, and contracting in the above ways to make the nectar source approach the optimal location. Since the nectar source is updated during each iteration, the problem of dimensionality reduction due to failure to take into account the correlation and number of points of the initial complex is avoided.

To overcome the defects exposed by the traditional ABC algorithm in dealing with the path planning of the UAH, an I-ABC algorithm is proposed based on the dynamic evaluation selection strategy and complex optimization method in the algorithm optimization layer. The I-ABC algorithm avoids the premature convergence problem, reduces the algorithm search time, and improves the quality of the flight
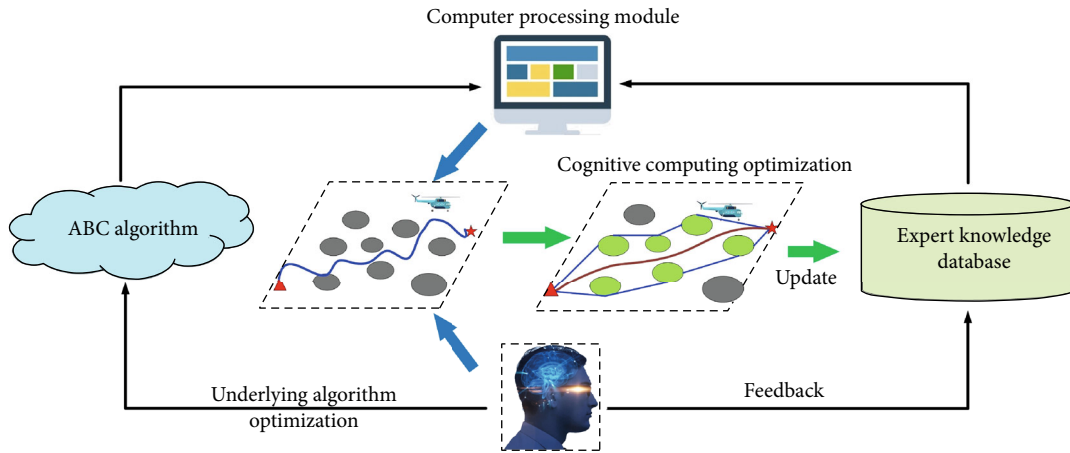
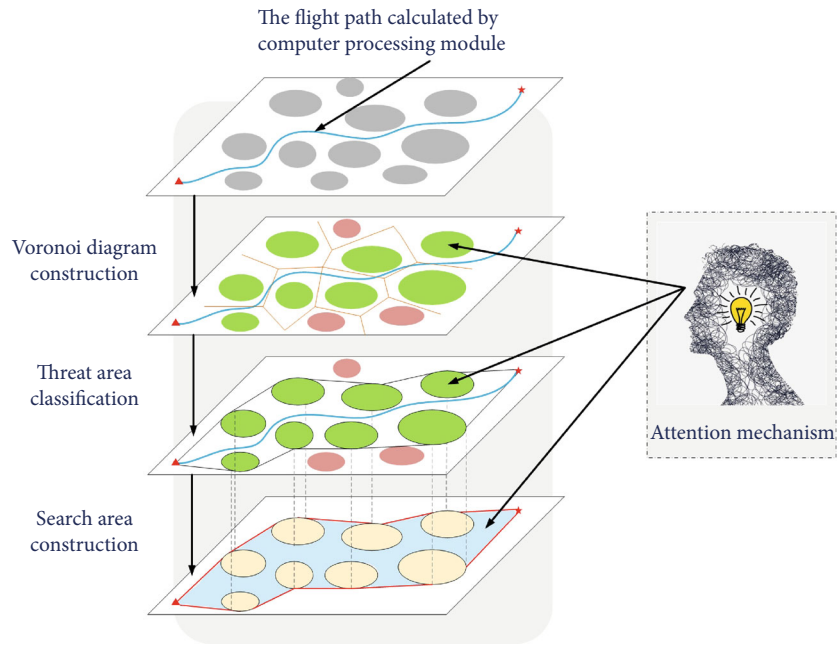FIGURE 8: Diagram of human-computer collaboration.

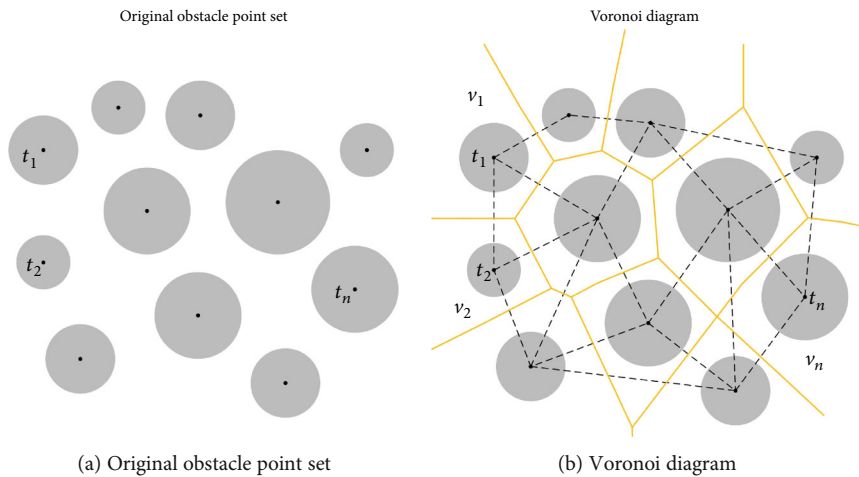FIGURE 9: Space clipping operation based on attentional mechanism.

(a) Original obstacle point set

(b) Voronoi diagram

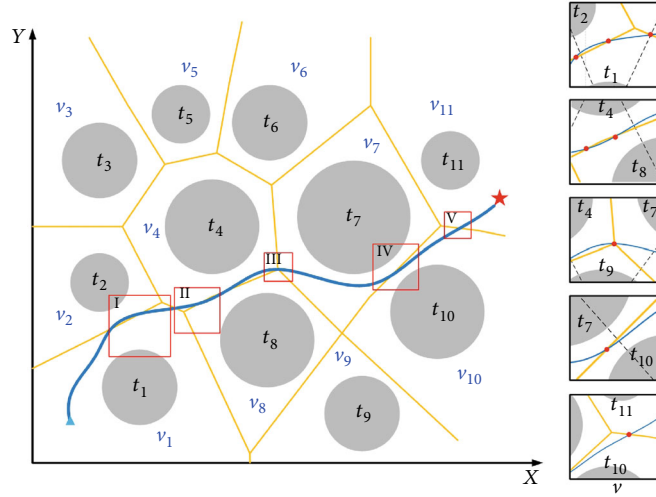FIGURE 10: Process of the space clipping operation.

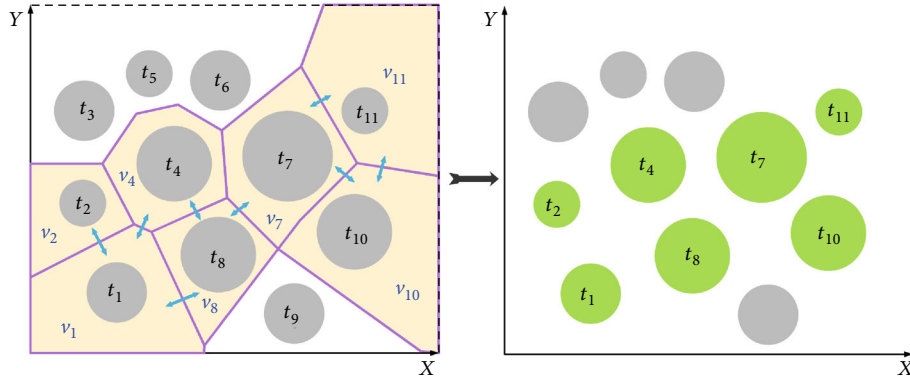FIGURE 11: The distribution of $\Xi_{i,j}^{k}$.

FIGURE 12: Associated convex polygonal areas and threats.

path. The flow diagram of the I-ABC algorithm is shown in Figure 7.

During the iteration, if an employed bee searches globally but finds no better nectar source, or if an onlooker bee searches around an employed bee and finds no better nectar source, the invalid trail time trial plus one. On the other hand, when any better nectar source can be searched by the ith employed bee, the relevant trial(i) is set to zero immediately. At the end of each iteration, it is necessary to determine whether any trial(i) outpaces a certain threshold Limit. If trial$(i) >$ Limit, the ith employed bee will be diametrically replaced by a scout bee. A scout bee still uses (8) to point to a randomly initialized location in the food source. The pseudocode of the I-ABC algorithm are shown in Algorithm 1.

Remark 1. The I-ABC algorithm can be divided into two phases. As the first phase of the program, the initialization phase is executed one time at the start, and the other phases are executed in each cycle. The computational complexity is mostly affected by the phase of the algorithm.

The computation is applied to the population with size of N, the individuals' position in population is a vector with size of D, and O is the asymptotic time complexity. The computing complexity $T(n)$ of each phase is shown as follows:

(i) Initialization phase

$$T_1(n) = \max\left[O(1), O(N \cdot D)\right] = O(N \cdot D). \tag{22}$$

(ii) Optimization phase

$$\begin{aligned} T_2(n) &= \max\left[O(N^2 \cdot D), O(N \cdot D), O(N \cdot D), O(N \cdot D)\right] \\ &= O(N^2 \cdot D). \end{aligned} \tag{23}$$
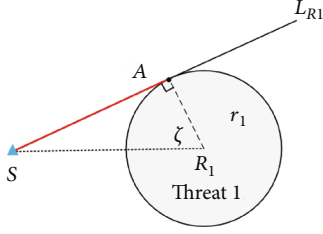
FIGURE 13: The boundary docking point of the start/end point to the associated threat.

FIGURE 14: The boundary docking point on the same associated threat.

(a) $R_1 = R_2$
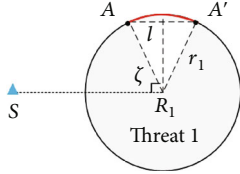
(b) $R_1 < R_2$

(c) $R_1 > R_2$

FIGURE 15: The boundary docking point of different associated threats.

Therefore, the maximum computing complexity of the I-ABC algorithm is shown as

$$T(n) = T_1(n) + T_2(n) = \max \left[O(N \cdot D), O(N^2 \cdot D) \right]$$
$$= O(N^2 \cdot D). \tag{24}$$

So it proves that this algorithm owns fast execution speed.

5. Space Clipping Operation Based on Attentional Mechanism

As shown in Figure 8, the incorporation of human intelligence is an important feature of the human-computer hybrid augmented intelligence framework. In order to further reduce the flight path search time and improve the efficiency of the UAH path planning system, a human computing layer is designed for the refinement operation of the flight path. Combined with the prior knowledge and human intelligence, the UAH flight space will be clipped by the spatial attention mechanism.

5.1. Attentional Mechanism. The main idea of the human computing layer presented in this paper is the space clipping based on the attention mechanism. The attention mechanism is a brain signal processing mechanism that is unique to humans [48]. The flight space clipping operation mimics the process by which the human brain rapidly sifts high-value information from large amounts of data through limited attentional resources [49]. The principle of the mechanism is shown in Figure 9.

The division of the threat areas will be determined by the degree of association between the flight path and the threat. Eventually, the spatial area of the path planing will be redrawn.

FIGURE 16: Diagram of new spatial search area.

5.2. Space Clipping Operation. In order to meet the constraint of minimizing the length of the path, the flight path can only through the safe space between a part of the threat areas. Numerous studies show that the unrelated threats will increase the search time of the swarm intelligence algorithm and reduce the efficiency of the path planning system. Therefore, it is crucial to distinguish different threat areas by the human computing layer. Combining the priori path planning results knowledge, the UAH flight space can be clipped by following steps.

(1) Voronoi diagram construction

The Voronoi diagram is widely used in terrain processing and other areas of division [8]. The points on each edge of the Voronoi diagram polygon are equidistant from the corresponding two threat points. In other words, all points on

FIGURE 17: Structure of the simulation.

the edge of the Voronoi diagram are as far away as possible from the threat.

Step 1. As shown in Figure 10(a), the threat region is considered as a point t_n, and all threat regions form a scatter set T of finite distances, that is, $T = \{t_1, t_2, \cdots, t_n\}$.

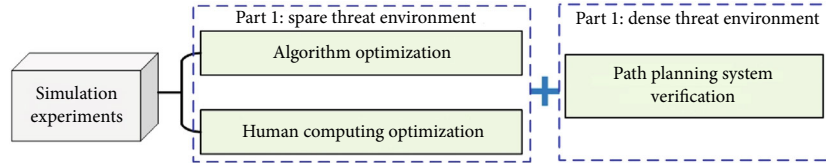*Step 2.* As shown in Figure 10(b), construct Delaunay triangles by connecting discrete points t_n into triangles. Find the edges of the Voronoi diagram by traversing the triangle chain table and draw the Voronoi diagram based on the final result.

The Voronoi diagram divides the flight path planning space into n convex polygons $v_i (i = 1, 2, \cdots, n)$ centered on the threat area, and the Voronoi diagram composed of $V = \{v_1, v_2, \cdots, v_n\}$ satisfies the following two conditions.

(1) Each convex polygon includes one and only one threat area, that is, $\forall v_i \in V$, $\exists! t_j \in T$, where $t_j \in v_i$, $\forall t_k \in T (k \neq j)$, $t_k \notin v_j$.

(2) Suppose that $d(x, y)$ is the Euclid distance on $\mathbb{R}^2$, if $t_j \in v_i$, so that $\forall t \in v_i$, $\forall t_k \neq t_j$, where $d(x, x_j) \leq d(x, x_k)$, $i, j = 1, 2 \cdots, n$.

(2) Threat areas classification

Assume that the flight path is C_{path}, the edge between threat t_i and threat t_j is $L_{v_{i,j}}$, the intersection of the flight path and the edge of the convex polygon is $\Xi_{i,j}^k$, where $i \neq j$ and $i, j = 1, 2, \cdots, n$, and k is the number of the intersection. The guidelines for determining the type of threat areas are as follows.

(i) If C_{path} intersects $L_{v_{i,j}}$ and the intersection point is $\Xi_{i,j}^k$ ($k \geq 1$), threat t_i and threat t_j are defined as associated threats. When the position of Ξ is located at the intersection of the three edges of the convex polygon, $\Xi = \Xi_{h,i,j}^k (h \neq i \neq j, h, i, j = 1, 2, \cdots, n, k = 1)$; threat t_h, threat t_i, and threat t_j are defined as associated threats

(ii) If there is no intersection between C_{path} and $L_{v_{ij}}$, that is, Ξ_{ij}^k does not exist, threat t_i and threat t_j are defined as unrelated threats

As shown in Figure 11, the flight space includes 11 threat areas ($T = \{t_1, t_2, \cdots, t_{11}\}$), and the Voronoi diagram divides them into 11 convex polygons ($V = \{v_1, v_2, \cdots, v_{11}\}$).

TABLE 1: The information of threat areas.

Threat number	Threat center (km)	Threat radius (km)
t_1	[125,-50]	50
t_2	[95,70]	60
t_3	[250,-110]	50
t_4	[230,7]	50
t_5	[240,160]	90
t_6	[310,50]	30
t_7	[360,-45]	60
t_8	[420,90]	70

TABLE 2: The information of threat areas.

Threat number	Threat center (km)	Threat radius (km)
t_1	[0,-150]	90
t_2	[60,-60]	40
t_3	[70,80]	55
t_4	[150,-90]	38
t_5	[130,0]	30
t_6	[210,160]	90
t_7	[230,7]	50
t_8	[250,-110]	50
t_9	[330,80]	60
t_{10}	[360,-50]	60
t_{11}	[500,-170]	110
t_{12}	[440,60]	40
t_{13}	[420,230]	100

According to the above theory, the threat areas in Figure 11 can be judged as follows.

(i) Part I

C_{path} and $L_{v_{1,2}}$ intersect at $\Xi_{1,2}^1$ and $\Xi_{1,2}^2$; threat t_1 and threat t_2 are associated threats; C_{path} and $L_{v_{1,4}}$ intersect at $\Xi_{1,4}^1$; threat t_1 and threat t_4 are associated threats.

(ii) Part II

C_{path} and $L_{v_{4,8}}$ intersect at $\Xi_{4,8}^1$ and $\Xi_{4,8}^2$; threat t_4 and threat t_8 are associated threats.
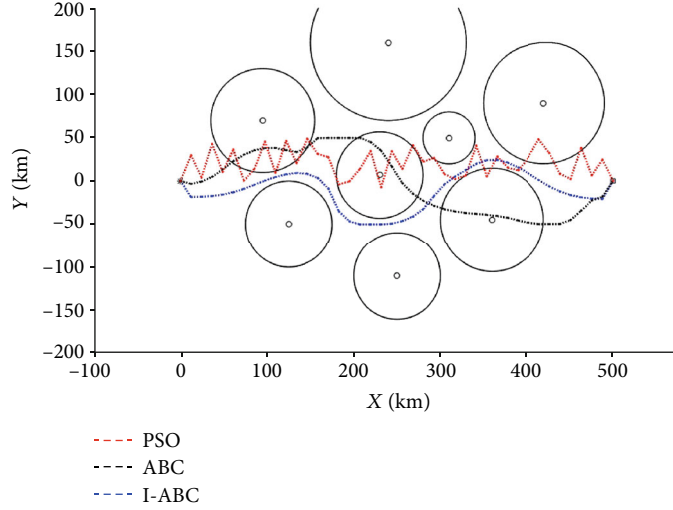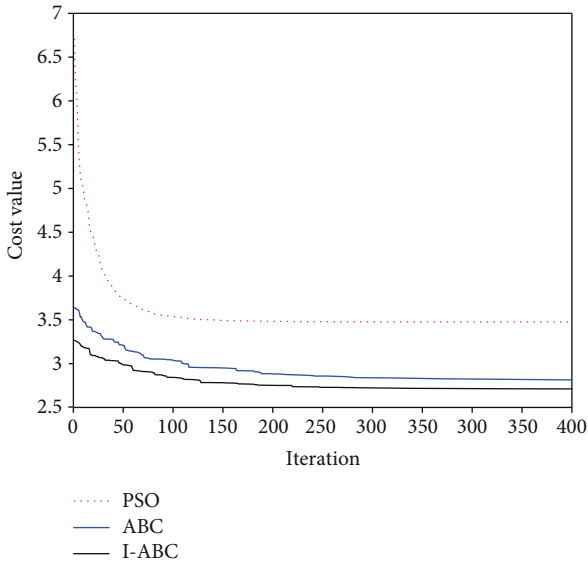
FIGURE 18: The comparative path planning results.

FIGURE 19: The evolution curves of three algorithms.

(a) Statistical cost value for the three algorithms
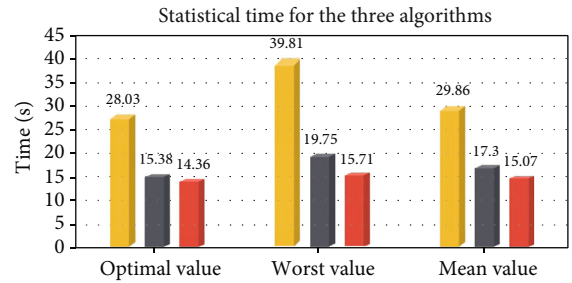
(b) Statistical time for the three algorithms

FIGURE 20: The statistical results for the three algorithms.

(iii) Part III

C_{path} and $L_{v_{4,7}}$ and $L_{v_{4,8}}$ and $L_{v_{7,8}}$ at $\Xi^1_{4,7,8}$; threat t_4, threat t_7, and threat t_8 are associated threats.

(iv) Part IV

C_{path} and $L_{v_{7,10}}$ intersect at $\Xi^1_{7,10}$; threat t_7 and threat t_{10} are associated threats.

(v) Part V

C_{path} and $L_{v_{10,11}}$ intersect at $\Xi^1_{10,11}$; threat t_{10} and threat t_{11} are associated threats.

In summary, as shown in Figure 12, the associated convex polygonal areas set is $V_{\text{associated}} = \{v_1, v_2, v_4, v_7, v_8, v_{10}, v_{11}\}$, and the associated threat set is $T_{\text{associated}} = \{t_1, t_2, t_4, t_7, t_8, t_{10}, t_{11}\}$.

5.3. Search Space Construction. The new spatial search area consists of a closed-loop space connected in sequence by the mission starting point S, the boundary docking points, and the mission end point T. The key of constructing a

TABLE 3: Results comparison for three algorithms.

Algorithms	PSO		ABC		I-ABC	
Performances	Value	Time(s)	Value	Time(s)	Value	Time(s)
Optimal	3.39	28.03	2.82	15.38	2.71	14.36
Worst	6.96	39.81	3.64	19.75	3.28	15.71
Mean	5.02	29.86	3.23	17.3	2.99	15.07

(a) Voronoi diagram construction

(b) $\Xi_{1,2}^1$

(c) $\Xi_{1,4}^1, \Xi_{1,4}^2, \Xi_{1,4}^3$

(d) $\Xi_{3,4}^1, \Xi_{3,4}^2$

(e) $\Xi_{4,6}^1$

(f) $\Xi_{6,7}^1$

(g) $\Xi_{7,8}^1$
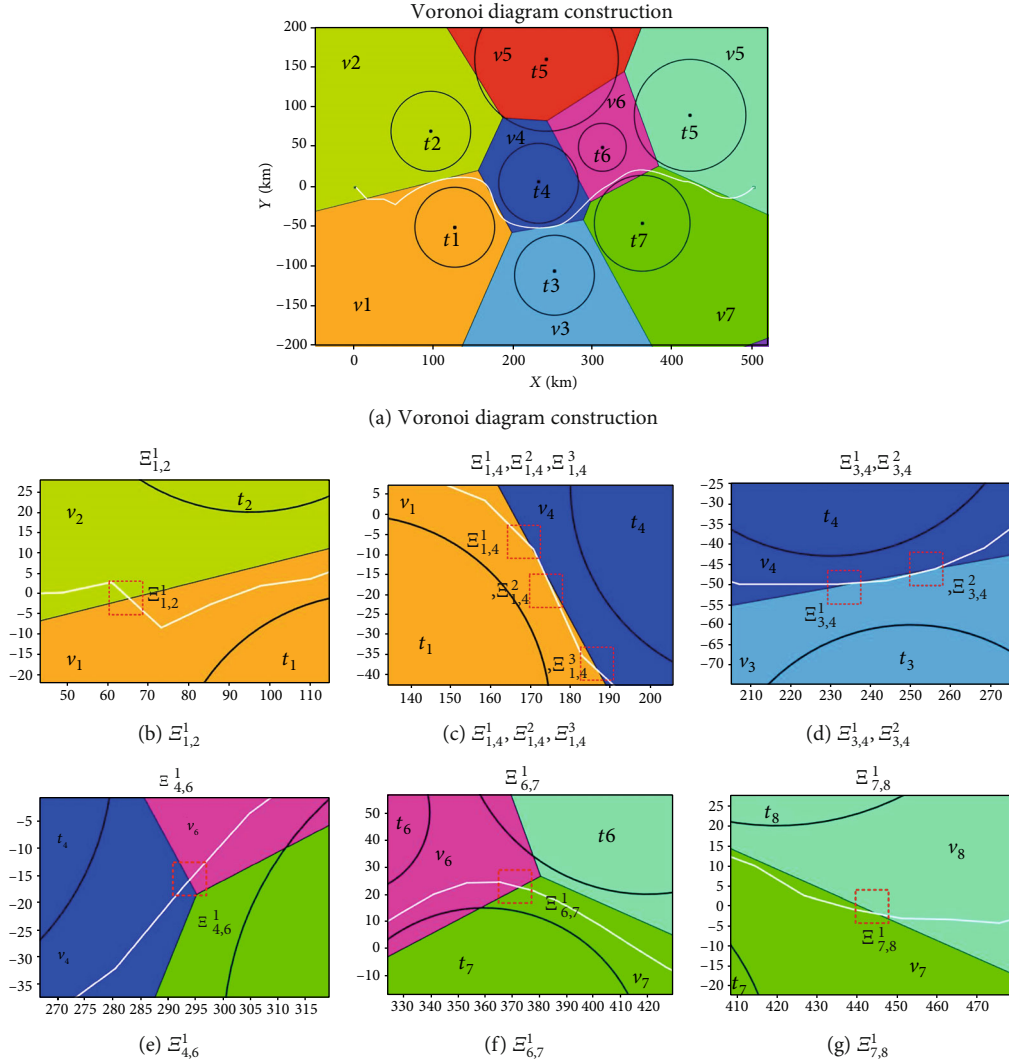
FIGURE 21: Threat space division.

new spatial search area are to find the reasonable boundary docking points and determine the spatial search boundary. In order to ensure that the results of the path planning in the new spatial search area remain largely optimal and the computation time of the swarm intelligence algorithm can be reduced. The spatial search area and the boundary docking point should satisfy the following properties.

(i) The spatial search area is a closed loop area

(ii) The spatial search area contains all associated threats $V_{\text{associated}}$

(iii) The positions of the boundary docking points should be on the associated threats' borders

To simplify the calculation, according to the properties of the space search boundary, the method for determining the boundary docking points is designed as follows.

(i) $r_1 = r_2$

Case 1. Shown in Figure 13, make the tangent line L_{R1} of the threat circle 1 through S (start or end point), and the tangent

(a) Threat area classification

(b) Search space construction

FIGURE 22: Threat space tailoring.

TABLE 4: Associated convex polygons and threats.

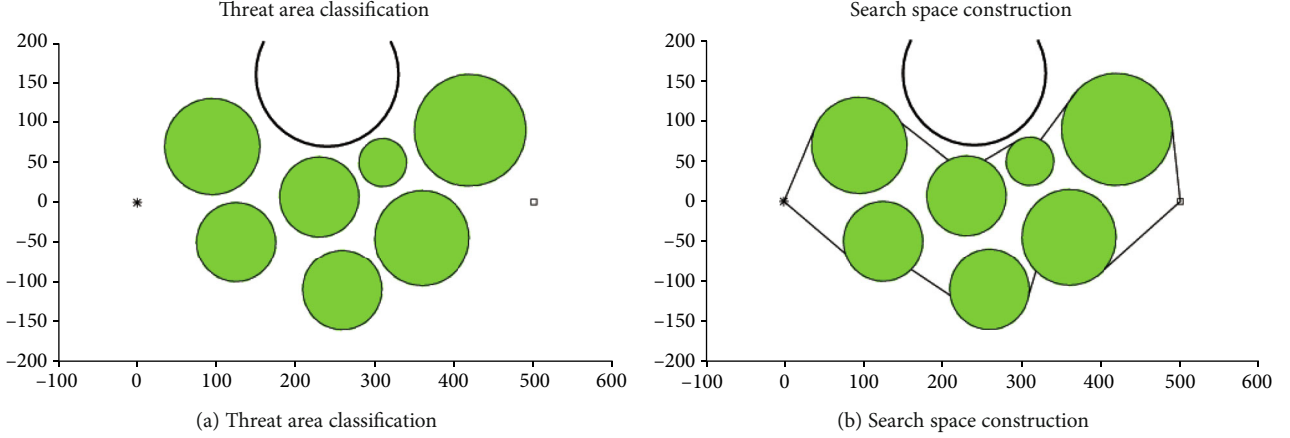| Intersection points	Associated convex polygons	Associated threats
$\Xi_{1,2}^1$	v_1, v_2	t_1, t_2
$\Xi_{1,4}^1$	v_1, v_4	t_1, t_4
$\Xi_{1,4}^2$	v_1, v_4	t_1, t_4
$\Xi_{1,4}^3$	v_1, v_4	t_1, t_4
$\Xi_{3,4}^1$	v_3, v_4	t_3, t_4
$\Xi_{3,4}^2$	v_3, v_4	t_3, t_4
$\Xi_{4,6}^1$	v_4, v_6	t_4, t_6
$\Xi_{6,7}^1$	v_6, v_7	t_6, t_7
$\Xi_{7,8}^1$	v_7, v_8	t_7, t_8

TABLE 5: Boundary docking points.

Associated threats	Boundary docking points	Distribution coordinates (km)
t_1	A	[77.45,−65.45]
	A'	[160.4,−85.36]
t_2	B	[41.54,97.24]
	B'	[148.5,97.24]
t_3	C	[209.6,−117.8]
	C'	[308.4,−117.8]
t_4	D	[207.3,51.55]
	D'	[252.7,51.55]
t_6	E	[296.4,76.73]
	E'	[323.6,76.73]
t_7	F	[317.6,−87.43]
	F'	[402.4,−87.43]
t_8	G	[370.5,139.5]
	G'	[489,101]

point A is the boundary docking point of the start/end point to the associated threat, where $AR_1 \perp L_{R1}$, $\angle AR_1 S = \zeta$.

Case 2. Shown in Figure 14, cross point R_1 to make the vertical line l of SR_1, find the symmetrical point A' of A with respect to l, and the point A' is the boundary docking point on the associated threat.

Case 3. Different associated threats have different threat radius. According to the radius of adjacent associated threats, the location of the boundary docking point can be divided into three situations.

As shown in Figure 15(a), straight line l' is the vertical bisector of line segment R_1R_2, and point B is the symmetry point of A' with respect to l'. For $r_1 = r_2$, the point B must be on the boundary of threat 2, so point B is the boundary docking point of associated threat 1 to associated threat 2.

(ii) $r_1 < r_2$

As shown in Figure 15(b), straight line l' is the vertical bisector of line segment R_1R_2, and point P is the symmetry

point of A' with respect to l', where P is within the threat 2. Crossing point P to make the vertical line $A'P$ of R_1R_2, and the intersection point B with threat 2 is the boundary docking point of associated threat 1 to associated threat 2, where $\angle AR_1R_2 = \angle BR_2R_1 = \zeta$.

(iii) $r_1 > r_2$

As shown in Figure 15(c), straight line l' is the vertical bisector of line segment R_1R_2, and point P is the symmetry point of A' with respect to l', where P is out of the threat 2. Crossing point P to make the vertical line $A'P$ of R_1R_2, and the intersection point B with threat 2 is the boundary docking point of associated threat 1 to associated threat 2, where $\angle A R_1R_2 = \angle BR_2R_1 = \zeta$.
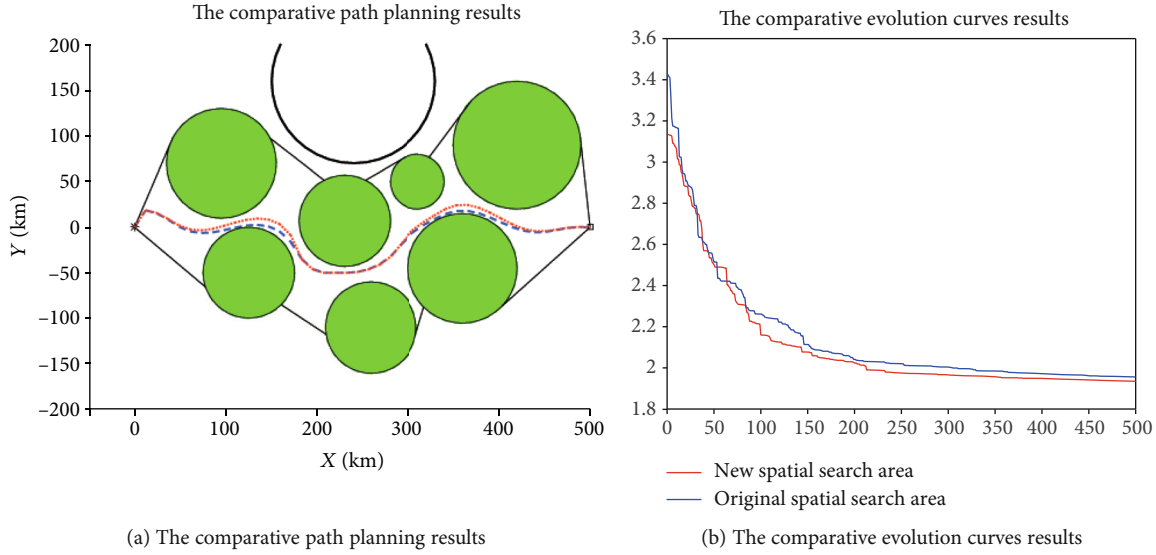
(a) The comparative path planning results

(b) The comparative evolution curves results

FIGURE 23: The comparative results for two different spatial search areas.
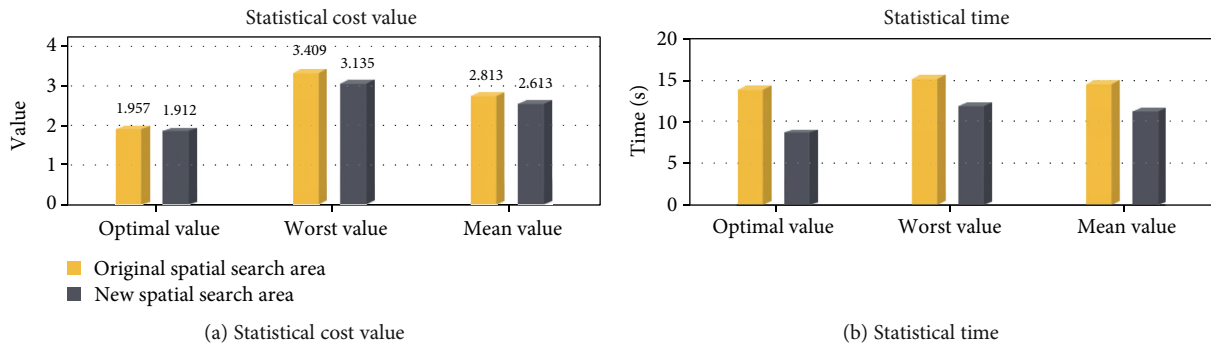

(a) Statistical cost value

(b) Statistical time

FIGURE 24: The statistical results of two different spatial search areas.

TABLE 6: Results comparison for two spatial search areas.

| | Original spatial search area | | New spatial search area | |
Performances	Cost value	Time (s)	Cost value	Time (s)
Optimal	1.957	14.36	1.912	9.05
Worst	3.409	15.71	3.135	12.34
Mean	2.813	15.07	2.613	11.69

Due to the location of the associated threats and the mission start/end point were known, the boundary docking point and the search boundary information can be determined by the prior knowledge. The new spatial search area is shown in Figure 16.

6. Simulation Experiments

In this section, for verifying the feasibility and effectiveness of the path planning system based on human-computer hybrid augmented intelligence framework for the UAH, a complete set of simulation experiments are designed. The simulation experiment process is shown in Figure 17.

The simulation experiments are conducted in two-dimension (2D) field. In the 2D field, the size of the UAH flight space is 700km ∗ 400km. The start point is set to [0,0]. The target point is set to [500,0]. We assume that the threats in the planning space are denoted by several circular areas. The related information of the threats is shown in Tables 1 and 2.

6.1. Algorithm Optimization Simulation Comparison Results. To show the superiority of the I-ABC algorithm improved in the algorithm optimization layer, the PSO algorithm, the ABC algorithm, and the I-ABC algorithm are simulated and compared under the same environment model. The maximum iteration number is set as 500, the dimension D
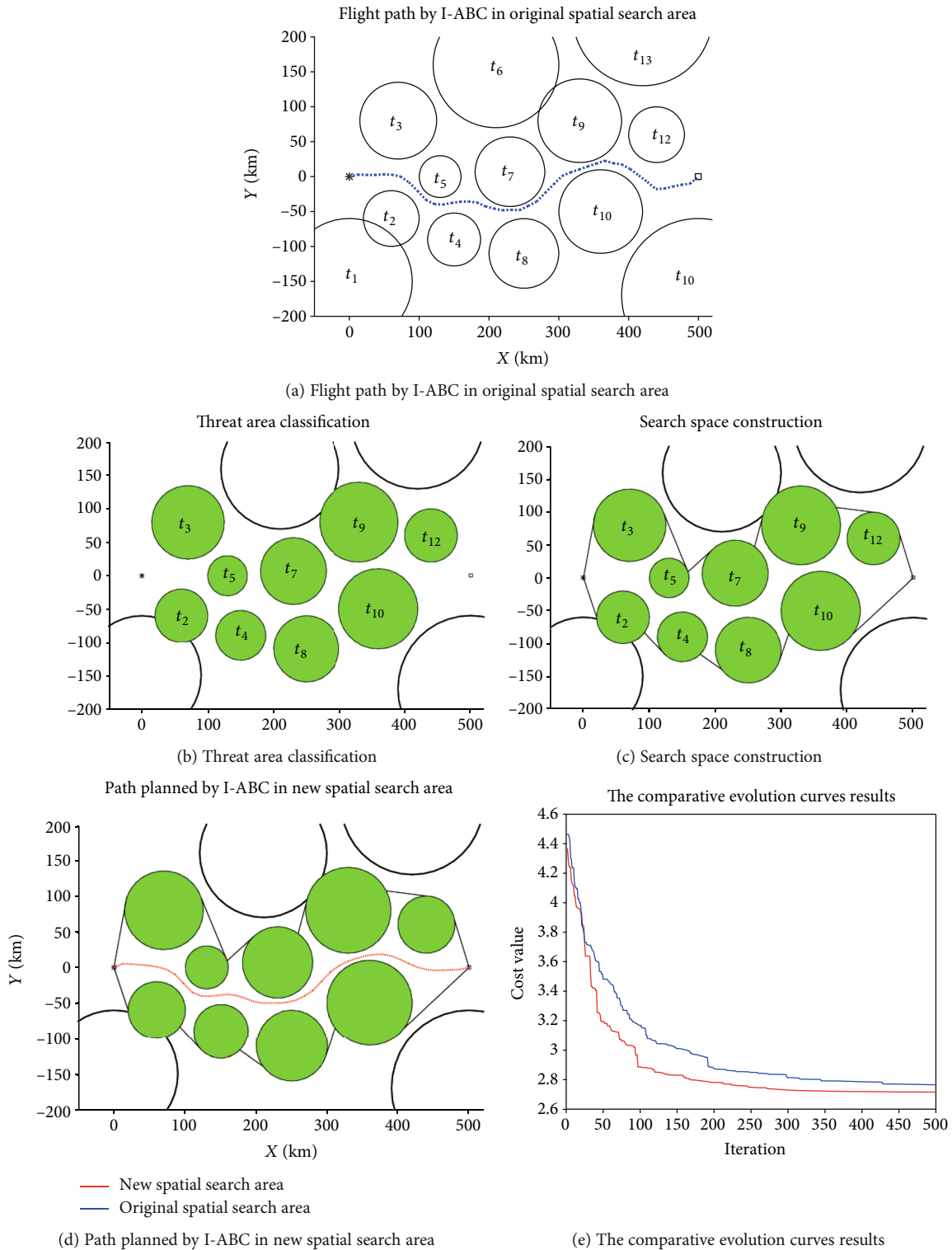
Flight path by I-ABC in original spatial search area

(a) Flight path by I-ABC in original spatial search area

Threat area classification

(b) Threat area classification

Search space construction

(c) Search space construction

Path planned by I-ABC in new spatial search area

The comparative evolution curves results

— New spatial search area
— Original spatial search area

(d) Path planned by I-ABC in new spatial search area

(e) The comparative evolution curves results

FIGURE 25: The planning results by I-ABC for the two different spatial search areas.

is set as 40, and the population size NP is set as 60. The results are averaged 20 independent runs, and the results of the simulation are illustrated in Figures 18, 19, and 20. The comparison result is listed in Table 3. In this table, the mean, worst, and optimal represent the mean fitness value, the worst fitness value, and the optimal fitness value, respectively.

Figure 18 shows that the intuitive differences of the experimental results between three algorithms in the 2D planning field. We can determine that the result planned by I-ABC algorithm can satisfy the requirements of the UAH path planning. The flight path planned by the PSO algorithm and the standard ABC algorithm have touched and crossed

(a) Statistical cost value
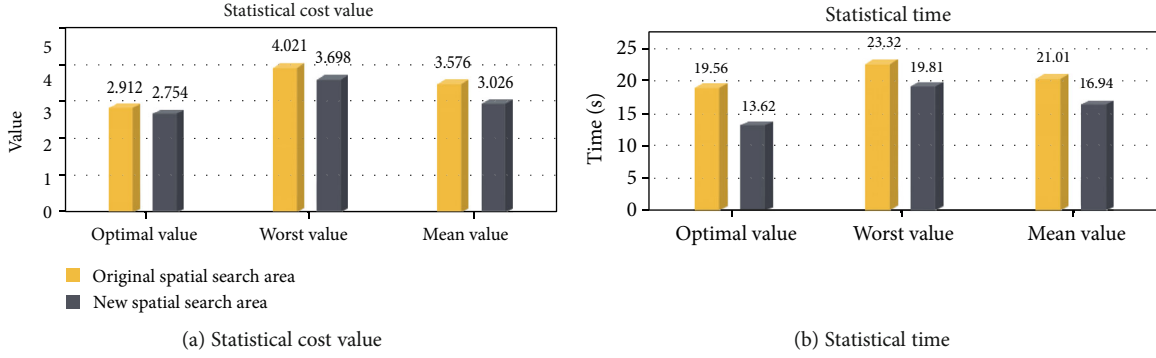

(b) Statistical time

FIGURE 26: The statistical results of two different spatial search areas.

the edge of the threats, and the result of the PSO algorithm is oscillatory.

The convergence curves of the three algorithms are shown in Figure 19. It can be obviously seen from these curves that the convergence effect of the I-ABC algorithm is better than other algorithms. The I-ABC algorithm attains the global optimal value in iteration 221. But the standard ABC algorithm approaches to their optimal value in iteration 278. What is worse is that the PSO algorithm still cannot find its optimal value when the number of iterations is reached.

The intuitive quantitative statistical results are shown in Figure 20 and Table 3. The statistical results of the PSO algorithm are poor, the optimal value is 3.39, the worst value is 6.96, and the average running time is 29.86 s. For the standard ABC algorithm, the optimal value is 2.82, the worst value is 3.64, and the average running time is 17.3 s. Compared with the ABC and PSO algorithms, the planning results of the I-ABC algorithm has smaller cost value and average running time. These denote that the I-ABC algorithm can search for the optimal path stably and also verify the superior performance of the algorithm optimization layer in improving the traditional standard ABC algorithm.

6.2. Human Computing Optimization Simulation Comparison Results. To further reduce the search time of the flight path and keep its feasibility, the flight space needs to be clipped according to the prior knowledge for constructing a new spatial search space. The process of the space clipping operation is demonstrated in Figures 21 and 22 and Tables 4 and 5. The results are averaged 20 independent runs in the different search spaces, and the results of the simulation are demonstrated in Figures 23 and 24 and Table 6.

Figure 21(a) shows that the original spatial search space was divided into 8 parts include $v_1, v_2, \cdots v_8$ by Voronoi diagram. According to the intersections of the flight path and the boundary of the convex polygon, the associated threats can be determined. Shown in Figures 21(b)–21(g)) and Table 4, the intersection point set is $\Xi_{1,2}^1$, $\Xi_{1,4}^1$, $\Xi_{1,4}^2$, $\Xi_{1,4}^3$, $\Xi_{3,4}^1$, $\Xi_{3,4}^2$, $\Xi_{4,6}^1$, $\Xi_{6,7}^1$, and $\Xi_{7,8}^1$, the associated convex polygons are $v_1, v_2, v_3, v_4, v_6, v_7$, and v_8, and the associated threats are $t_1, t_2, t_3, t_4, t_6, t_7$, and t_8. Table 5 shows the coordinates of the boundary docking points, by connecting the start/end points with the boundary docking points, we can get the new search space construction as shown in Figure 22.

TABLE 7: Results comparison for two spatial search areas.

Performances	Original spatial search area		New spatial search area	
	Cost value	Time (s)	Cost value	Time (s)
Optimal	2.912	19.56	2.754	13.6
Worst	4.021	23.3	3.698	19.81
Mean	3.576	21.01	3.026	16.94

Figure 23(a) shows the path planning results of the I-ABC algorithm in two different spatial search areas. From the planning results, we can intuitively find that the final results planned by the I-ABC algorithm have almost no differences in two different spatial search areas. The convergence curves of two spatial search areas are illustrated in Figure 23(b). Similarly, from these curves, it can be find that the convergence effect by I-ABC algorithm in the new spatial search areas is slightly better than the original spatial search areas; however, the differences can be almost ignored in practical applications. In other words, the flight path result in the new spatial search area still maintains its optimality in the original spatial search area.

The difference between the flight path of two spatial search areas can be found through the data in Figure 24. For the original spatial search area, the optimal value is 1.957, the worst value is 3.409, and the mean value is 2.813. The optimal search time of the I-ABC algorithm witch runs 20 times is 14.56 s, the worst search time is 15.71 s, and the mean search time is 15.07 s. For the new spatial search area, the optimal value is 1.912, the worst value is 2.778, and the mean value is 2.346. The optimal search time of the I-ABC algorithm witch runs 20 times is 9.05 s, the worst search time is 12.34 s, and the mean search time is 11.69 s. By the comparison, the change in cost value is minimal. And the smoothness and feasibility of the flight path have not changed in the new spatial search area. These denote that the space clipping operation in the human computing layer can indeed reduce the search time of the flight path under the planning result within acceptable changes. Furthermore, the efficiency of the path planning system can be improved.

6.3. UAH Path Planning System Simulation Results. To further verify the effect of the space clipping operation for

the path planning in the complex environment, a complicated environment model with more threats will be used for the simulation. The information of the threats in new environment is shown in Table 2. The maximum iteration number of the I-ABC algorithm is set as 500, the dimension D is set as 40, the population size NP is set as 60. The simulation results of the path planning system for the UAH in the new map are shown as Figures 25(a)–25(e) and 26 and Table 7.

Figure 25(a) shows that the path planning result by the I-ABC algorithm in the original spatial search area. We can see that the result planned by I-ABC algorithm avoids all threat areas, and it can meet the constraints of the UAH path planning.

Figures 25(b) and 25(c), respectively, show the process of the threat area classification and the search space construction. As shown in Figure 25(b), the associate threats can be identified as $t_2, t_3, t_4, t_5, t_7, t_8, t_9, t_{10}$, and t_{12}. The new spatial search area consists of associated threats and black spatial search boundaries (as shown in Figure 25(c)).

Figure 25(d) shows the path planning results of the I-ABC algorithm in the new spatial search area. Compare with Figure 25(a), the flight path planned in the new spatial search area is similar to the flight path planned in the original spatial search area, both in terms of path smoothness and flight length. The convergence curves of the two spatial search areas are displayed in Figure 25(e). According to these curves, it can be obviously found that the convergence effect by I-ABC algorithm in the new spatial search areas is also similar to the original spatial search areas.

The difference between the planning results of two spatial search areas can be found through the data in Figure 26. For the original spatial search area, the optimal value is 2.912, the worst value is 4.021, and the mean value is 3.576. The optimal search time of the I-ABC algorithm witch runs 20 times is 19.56 s, the worst search time is 23.32 s, and the mean search time is 21.01 s. For the new spatial search area, the optimal value is 2.754, the worst value is 3.698, and the mean value is 3.026. The optimal search time of the I-ABC algorithm witch runs 20 times is 13.62 s, the worst search time is 19.81 s, and the mean search time is 16.94 s. By the comparison, the space clipping operation can maintain the excellent performance of the previous flight path planned by I-ABC algorithm. Moreover, the search time of the path planning system can be further reduced.

7. Conclusion

In this paper, a path planning system is presented based on human-computer hybrid augmented intelligence framework for improving the rapidity and optimality of the UAH path planner. At first, the proposed I-ABC algorithm optimizes the follow way of on-looker bees and the update strategy of nectar source. The dynamic evaluation selection strategy and complex optimization method accelerate the convergence rate and maintain the exploration ability of the population. In addition, the space clipping operation based on the attention mechanism reconstructs the spatial search area of the UAH. Unlike adjusting the number of iterations or

modifying the parameters of the algorithm, the space clipping operation is directly applied to the optimal solution searched by the I-ABC algorithm. Benefit from this operation, the subsequent search time of the flight path for the path planning system is further reduced. Eventually, the experimental results show that the path planning system based on human-computer hybrid augmented intelligence framework for the UAH can combine the advantages of human and computer for planning a high-quality flight path. This study provides promising results for actual military mission.

Data Availability

The experimental data of this study are included within the article.

Conflicts of Interest

All authors declare no conflict of interest.

Acknowledgments

References

[1] F. Santoso, M. A. Garratt, and S. G. Anavatti, "State-of-the-art intelligent flight control systems in unmanned aerial vehicles," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 613–627, 2018.

[2] W. Liu, Z. Zheng, and K.-Y. Cai, "Bi-level programming based real-time path planning for unmanned aerial vehicles," *Knowledge-Based Systems*, vol. 44, pp. 34–47, 2013.

[3] C. Sun, Y.-C. Liu, R. Dai, and D. Grymin, "Two approaches for path planning of unmanned aerial vehicles with avoidance zones," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 8, pp. 2076–2083, 2017.

[4] Y.-C. Du, M.-X. Zhang, H.-F. Ling, and Y.-J. Zheng, "Evolutionary planning of multi-UAV search for missing tourists," *IEEE Access*, vol. 7, pp. 73480–73492, 2019.

[5] W.-L. Liu, Y.-J. Gong, W.-N. Chen, Z. Liu, H. Wang, and J. Zhang, "Coordinated charging scheduling of electric vehicles: a mixed-variable differential evolution approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5094–5109, 2020.

[6] R. Thangaraj, M. Pant, A. Abraham, and P. Bouvry, "Particle swarm optimization: hybridization perspectives and experimental illustrations," *Applied Mathematics and Computation*, vol. 217, no. 12, pp. 5208–5226, 2011.

[7] F. J. Rodriguez, C. Garcia-Martinez, and M. Lozano, "Hybrid metaheuristics based on evolutionary algorithms and simulated annealing: taxonomy, comparison, and synergy test," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 6, pp. 787–800, 2012.

[8] Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of

autonomous UAV," *Aerospace Science and Technology*, vol. 16, no. 1, pp. 47–55, 2012.

[9] I. S. AlShawi, L. Yan, W. Pan, and B. Luo, "Lifetime enhancement in wireless sensor networks using fuzzy approach and A-star algorithm," *IEEE Sensors Journal*, vol. 12, no. 10, pp. 3010–3018, 2012.

[10] Y.-b. Chen, G.-c. Luo, Y.-s. Mei, J.-q. Yu, and X.-l. Su, "UAV path planning using artificial potential field method updated by optimal control theory," *International Journal of Systems Science*, vol. 47, no. 6, pp. 1407–1420, 2014.

[11] M. Kothari and I. Postlethwaite, "A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees," *Journal of Intelligent & Robotic Systems*, vol. 71, no. 2, pp. 231–253, 2013.

[12] D. Manjarres, I. Landa-Torres, S. Gil-Lopez et al., "A survey on applications of the harmony search algorithm," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 8, pp. 1818–1831, 2013.

[13] W. Guo, M. Chen, L. Wang, Y. Mao, and Q. Wu, "A survey of biogeography-based optimization," *Neural Computing and Applications*, vol. 28, no. 8, pp. 1909–1926, 2016.

[14] M. T. Younis and S. Yang, "Hybrid meta-heuristic algorithms for independent job scheduling in grid computing," *Applied Soft Computing*, vol. 72, pp. 498–517, 2018.

[15] J. Kennedy and R. Eberhart, "Eberhart particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw*, pp. 1942–1948, Piscataway, 1995.

[16] J. D. S. Arantes, M. D. S. Arantes, and C. F. M. Toledo, "Heuristic and genetic algorithm approaches for UAV path planning under critical situation," *International Journal on Artificial Intelligence Tools*, vol. 26, no. 1, pp. 1793–6349, 2017.

[17] R. Storn and K. Price, "Differential evolution a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[18] W. Deng, J. J. Xu, Y. J. Song, and H. M. Zhao, "Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem," *Applied Soft Computing*, no. article 106724, 2020.

[19] Y. J. Song, D. Q. Wu, W. Deng et al., "MPPCEDE: multipopulation parallel co-evolutionary differential evolution for parameter optimization," *Energy Conversion and Management*, vol. 228, 2021.

[20] W. Deng, J. J. Xu, X. Z. Gao, and H. M. Zhao, "An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–10, 2020.

[21] Y. Zhao, Z. Zheng, and Y. Liu, "Survey on computational-intelligence-based UAV path planning," *Knowledge-Based Systems*, vol. 158, pp. 54–64, 2018.

[22] C. Y. Bo, M. Y. Song, Y. J. Qiao, S. X. Long, and X. Nuo, "Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm," *Neurocomputing*, vol. 266, pp. 445–457, 2017.

[23] S. Zhang, Y. Zhou, Z. Li, and W. Pan, "Grey wolf optimizer for unmanned combat aerial vehicle path planning," *Advances in Engineering Software*, vol. 99, pp. 121–136, 2016.

[24] G. G. Wang, H. E. Chu, and S. Mirjalili, "Three-dimensional path planning for UCAV using an improved bat algorithm," *Aerospace Science and Technology*, vol. 49, pp. 231–238, 2016.

[25] P. Kumar, S. Garg, A. Singh, S. Batra, N. Kumar, and I. You, "MVO-based 2D path planning scheme for providing quality of service in UAV environment," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1698–1707, 2018.

[26] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25–33, 2017.

[27] Z. Yin, X. Liu, and Z. Wu, "A multiuser detector based on artificial bee colony algorithm for DS-UWB systems," *The Scientific World Journal*, vol. 13, Article ID 547656, 454 pages, 2013.

[28] F. Kang, J. Li, and Z. Ma, "Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions," *Information Sciences*, vol. 181, no. 16, pp. 3508–3531, 2011.

[29] D. Haijun and F. Qingxian, "Artificial bee colony algorithm based on boltzmann selection strategy," *Computer Engineering and Applications*, vol. 45, no. 32, pp. 53–55, 2009.

[30] Y. Li, Y. Wang, and B. Li, "A hybrid artificial bee colony assisted differential evolution algorithm for optimal reactive power flow," *International Journal of Electrical Power and Energy Systems*, vol. 52, pp. 25–33, 2013.

[31] M. S. Kiran and M. Gunduz, "A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems," *Applied Soft Computing*, vol. 13, no. 4, pp. 2188–2203, 2013.

[32] H. Duan, Z. Xing, and C. Xu, "An improved quantum evolutionary algorithm based on artificial bee colony optimization," *Advances in Computational Intelligence*, vol. 116, pp. 269–278, 2009.

[33] S. Mehdizadeh, F. Fathian, and J. F. Adamowski, "Hybrid artificial intelligence-time series models for monthly streamflow modeling," *Applied Soft Computing*, vol. 80, pp. 873–887, 2019.

[34] N. N. Zheng, Z. Y. Liu, P. J. Ren et al., "Hybrid-augmented intelligence: collaboration and cognitionility," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 2, pp. 153–179, 2017.

[35] W. Zeng, J. Yuan, C. Yuan, Q. Wang, F. Liu, and Y. Wang, "Classification of myocardial infarction based on hybrid feature extraction and artificial intelligence tools by adopting tunable-Q wavelet transform (TQWT), variational mode decomposition (VMD) and neural networks," *Artificial Intelligence in Medicine*, vol. 106, article 547656, 2020.

[36] H. Yu, L. Zhu, L. Cai et al., "Variation of functional brain connectivity in epileptic seizures: an EEG analysis with cross-frequency phase synchronization," *Cognitive Neurodynamics*, vol. 14, no. 1, pp. 35–49, 2020.

[37] H. Jianfeng and M. Jianliang, "Automated detection of driver fatigue based on EEG signals using gradient boosting decision tree model," *Cognitive Neurodynamics*, vol. 12, no. 4, pp. 431–440, 2018.

[38] X. Chunfang, H. Duan, and F. Liu, "Chaotic artificial bee colony approach to Uninhabited Combat Air Vehicle (UCAV) path planning," *Aerospace Science and Technology*, vol. 14, no. 8, pp. 535–541, 2010.

[39] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Systems with Applications*, vol. 37, no. 8, pp. 5682–5687, 2010.

[40] F. Yangguang, M. Ding, and C. Zhou, "Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV," *Systems,*

Man and Cybernetics, IEEE Transactions, vol. 42, no. 2, pp. 511–526, 2012.

[41] H. Duan, S. Liu, and J. Wu, "Novel intelligent water drops optimization approach to single UCAV smooth trajectory planning," *Aerospace Science and Technology*, vol. 13, no. 8, pp. 442–449, 2009.

[42] M. Francia, M. Golfarelli, and S. Rizzi, "A-BI+: a framework for Augmented Business Intelligence," *Information Systems*, vol. 92, article 101520, 2020.

[43] K. Meng, Y. Cao, X. Peng, V. Prybutok, and K. Youcef-Toumi, "Smart recovery decision-making for end-of-life products in the context of ubiquitous information and computational intelligence," *Journal of Cleaner Production*, vol. 272, article 122804, 2020.

[44] J. Radianti, T. A. Majchrzak, J. Fromm, and I. Wohlgenannt, "A systematic review of immersive virtual reality applications for higher education: design elements, lessons learned, and research agenda," *Computers and Education*, vol. 147, article 103778, 2020.

[45] K. Vartia, J. Taponen, J. Heikkinen, and H. Lindeberg, "Effect of education on ability of AI professionals and herd-owner inseminators to detect cows not in oestrus and its relation with progesterone concentration on day of re-insemination," *Theriogenology*, vol. 102, no. 10, pp. 23–28, 2017.

[46] A. D. Duru and M. Assem, "Investigating neural efficiency of elite karate athletes during a mental arithmetic task using EEG," *Cognitive Neurodynamics*, vol. 12, no. 1, pp. 95–102, 2018.

[47] F. Erdogdu, S. E. Zorrilla, and R. P. Singh, "Effects of different objective functions on optimal decision variables: a study using modified complex method to optimize hamburger cooking," *LWT - Food Science and Technology*, vol. 38, no. 2, pp. 111–118, 2005.

[48] A. A. Abbasi, L. Hussain, I. A. Awan et al., "Detecting prostate cancer using deep learning convolution neural network with transfer learning approach," *Cognitive Neurodynamics*, vol. 14, no. 4, pp. 523–533, 2020.

[49] L. Gang and G. Jiabao, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, 2019.