

Article

End-Cloud Collaboration Navigation Planning Method for Unmanned Aerial Vehicles Used in Small Areas

Hua jie Xiong, Baoguo Yu *, Qingwu Yi and Chenglong He

State Key Laboratory of Satellite Navigation System and Equipment Technology, Shijiazhuang 050081, China

* Correspondence: yubg@sina.cn

Abstract: Unmanned aerial vehicle (UAV) collaboration has become the main means of indoor and outdoor regional search, railway patrol, and other tasks, and navigation planning is one of the key, albeit difficult, technologies. The purpose of UAV navigation planning is to plan reasonable trajectories for UAVs to avoid obstacles and reach the task area. Essentially, it is a complex optimization problem that requires the use of navigation planning algorithms to search for path-point solutions that meet the requirements under the guide of objective functions and constraints. At present, there are autonomous navigation modes of UAVs relying on airborne sensors and navigation control modes of UAVs relying on ground control stations (GCSs). However, due to the limitation of airborne processor computing power, and background command and control communication delay, a navigation planning method that takes into account accuracy and timeliness is needed. First, the navigation planning architecture of UAVs of end-cloud collaboration was designed. Then, the background cloud navigation planning algorithm of UAVs was designed based on the improved particle swarm optimization (PSO). Next, the navigation control algorithm of the UAV terminals was designed based on the multi-objective hybrid swarm intelligent optimization algorithm. Finally, the computer simulation and actual indoor-environment flight test based on small rotor UAVs were designed and conducted. The results showed that the proposed method is correct and feasible, and can improve the effectiveness and efficiency of navigation planning of UAVs.

Keywords: navigation planning; unmanned aerial vehicles; end-cloud collaboration; improved particle swarm optimization



Citation: Xiong, H.; Yu, B.; Yi, Q.; He, C. End-Cloud Collaboration

Navigation Planning Method for Unmanned Aerial Vehicles Used in Small Areas. *Sensors* **2023**, *23*, 7129. <https://doi.org/10.3390/s23167129>

Received: 23 May 2023

Revised: 23 June 2023

Accepted: 24 June 2023

Published: 11 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Navigation planning is one of the key technologies for UAVs to synergistically perform regional search [1], communication support [2], logistics transportation [3], forest prevention and control [4], and other tasks. Navigation planning refers to the formulation of the optimal flight path from the initial point to the target point via the UAV platform or background assistance after task assignment, meeting the constraints of the UAV performance and environmental conditions, and taking into account the terrain, threats and other factors [5]. The cluster of UAV collaborative navigation planning is different from single-UAV navigation planning, and the factors to be considered are more complex, as they include collision avoidance between UAVs and the arrival of synchronization time in the flight process, making the whole cluster mission benefit the most or the flight cost the least [6]. Conducting research on navigation planning technology can ensure the safe and reliable flight of UAVs in complex environments and achieve mission objectives. In general, the main ways to solve the current UAV navigation planning rely on traditional remote control and ground control stations [7], as well as autonomous sensors and navigation algorithms such as the visual navigation system and inertial navigation system (INS), to achieve simple autonomous control planning [8].

As an important component of the UAV system, the ground control station is the man-machine interface of UAVs [9]. UAVs return information and images to the ground

station through the downlink and receive navigation control commands from the ground station through the uplink. One type of control command comprises UAV flight control commands, such as speed, acceleration, heading angle, etc., which can be sent via a remote controller or ground station; another is the flight path target point position information, which is generally sent by the ground station to make the UAV fly to the target position [10]. At present, the common UAV ground station system developed by the mission planner can realize the configuration and simple flight control of the UAV [11]. DJI has developed DJI GS PRO ground station based on iPad, which can realize UAV fixed-path planning and automatic return based on a remote sensing map [12]. In addition to being directly selected by the operator, the location information of the path target point can also be analyzed and calculated by the intelligent algorithm. For example, in [13,14], the deep migration reinforcement learning method and gray wolf algorithm, respectively, were used to achieve the task assignment of multiple UAVs. In [15], the authors used an improved mayfly algorithm to realize UAV path planning. The advantage of background control and navigation planning based on a ground control station is that if accurate environmental situation information can be obtained, the flight path obtained by high background computing power can often be a high-quality solution. However, under the conditions of a complex mission environment and limited communication, it is difficult to ensure the real-time and accurate return of environmental situation information and transmit flight instructions to the front end of the UAV in real time [16].

On the other hand, autonomous navigation based on visual positioning and onboard navigation processing has always been the research focus of UAV navigation control [17]. UAVs realize autonomous positioning through the information fusion of an airborne visual sensor and IMU, and then realize autonomous navigation capabilities such as obstacle-avoidance flight partly through the airborne autonomous path planning algorithm. For example, reference [18] realized obstacle avoidance of a UAV in a dynamic environment based on point-cloud image; reference [19] adopted deep reinforcement learning to realize an end-to-end obstacle-avoidance decision of UAVs. The advantage of navigation decision planning based on airborne autonomous sensors and processors is that the real-time requirement can be satisfied. However, depending on the existing autonomous navigation algorithm in a relatively complex task environment, there are often problems such as insufficient processing capacity and credibility of the autonomous navigation decision of UAVs. In particular, when facing the problem of cluster navigation planning, the key is to design the navigation planning algorithm and technology at a fast speed and with high accuracy [20].

In terms of path planning algorithms, those used at present mainly include three types: traditional algorithms, algorithms based on graph search, and heuristic intelligent search algorithms. Specifically, the authors in [21] proposed a method based on an improved A* algorithm for three-dimensional path planning, and in [22], the coverage path planning problem of autonomous heterogeneous UAVs was studied over a limited number of areas, the region classified into clusters, and approximate optimal point-to-point paths for drones obtained. In [23], the authors used the particle swarm optimization beetle antennae search (PSO-BAS) algorithm to plan the spatial 3D route. The three types of algorithms have their own characteristics. Traditional algorithms can search for the optimal solution, but do not meet real-time requirements due to the high computational intensity. Algorithms based on graph search reduce the computational intensity, but often lack the ability to search for excellent solutions. In terms of balancing the search speed and solution quality, as well as improvement of the algorithm, heuristic intelligent search algorithms have more advantages. The focus of this study is to solve the navigation planning problem of UAVs in complex and communication-limited environments. To this end, an end-cloud collaborative navigation planning algorithm for UAVs was proposed, which combined the background cloud navigation planning algorithm based on an improved PSO and the autonomous navigation control algorithm for UAVs based on a multi-objective hybrid swarm intelligence algorithm. This end-cloud collaborative navigation planning method

combines fast real-time path search algorithms on the UAV onboard part and high-quality path planning algorithms on the ground control part, solving the contradiction between the high-quality and real-time path planning that other navigation planning algorithms are difficult to balance. Simulation verification based on software and actual dynamic flight experiment verification based on UAVs and ground station were carried out. The results showed that the algorithm can meet the real-time requirements and improve the accuracy and efficiency of UAV navigation planning. Compared with other navigation planning algorithms, the results show that this algorithm has advantages in search speed and objective function evaluation value.

2. System Architecture and Algorithm Design

2.1. UAV Navigation Planning Modeling

UAV navigation planning mainly comprises three components: environment modeling, objective function modeling, and navigation planning algorithm design.

Environment modeling includes the total area size and location of the UAV flight tasks, the positions of takeoff start points and mission target points, as well as obstacle and threat information (coordinates, range, altitude, etc.) in the area. Environmental modeling is a prerequisite for UAVs to successfully complete navigation planning. The specific modeling process is shown in Section 3.1.1. The algorithm design part is described in detail in Section 2.2. This section focuses on the modeling and design of the objective function for UAV navigation planning.

The objective function is a key indicator for evaluating the performance of navigation planning, consisting of multiple constraint terms. Constraints usually involve UAVs themselves, and the task and environment, which together determine the quality of the navigation planning solution. For the convenience of research, the article does not consider the UAV flight control model, and adds dynamic constraints of UAVs in the objective function model section. The main constraints are as follows:

- Flight distance

Assuming that the flight speed of the UAV and the average energy consumption remain constant, due to the limited energy capacity of the UAV itself and the limited communication distance with the background, a maximum range must be set to constrain it. On the premise of completing the task, the smaller the range, the better. Assuming that the UAV trajectory consists of N trajectory nodes and $N - 1$ segments, the length of the i segment trajectory is L_i and the flight distance constraint of the UAV is:

$$L = \sum_{i=1}^{N-1} L_i \leq L_{max} \quad (1)$$

- Flight altitude

During flight, the UAV should maintain a low flight altitude to ensure communication with the ground station, but this will increase the probability of colliding with obstacles such as the ground and mountains. Therefore, a minimum value was set to minimize the flight altitude of the UAV, while ensuring that the altitude is not less than this value. The flight altitude constraint is represented as follows:

$$H = \sum_{i=1}^N H_i \geq H_{min} \quad (2)$$

In the formula, H_i is the flight altitude of the UAV at the i th track point and H_{min} is the minimum flight altitude value for UAV.

- Distance from the UAV to obstacles

Contrary to flight altitude constraints, in order to ensure effective collision avoidance, the larger the distance between the UAV and the obstacle, the better.

$$\lambda = \sum_{i=1}^N \lambda_i \geq \lambda_{min} \quad (3)$$

In the formula, λ_i is the distance from the UAV to obstacles at the i th track point and λ_{min} is the minimum distance value.

- Distance between UAVs

This parameter is the sum of the distances between UAVs in the cluster during flight. The distance between UAVs should not be small enough to increase the risk of collision, and it should not be large enough to cause communication barriers between UAVs.

$$\rho = \rho_{min} \leq \sum_{i=1}^N \rho_i \leq \rho_{max} \quad (4)$$

In the formula, ρ_i is the average of the distances between the UAV and other UAVs at the i th track point. ρ_{max} is the maximum average distance between UAVs that can ensure cluster communication and ρ_{min} is the minimum average distance between UAVs to ensure collision avoidance.

- Dynamic constraints of UAVs

The flight dynamics constraints of UAVs include the turning radius and climbing angle. When UAVs encounter obstacles or threats, they need to turn or climb to avoid them. When turning, the larger the turning radius while avoiding threats or obstacles, the smoother the trajectory. If the turning radius is too small, it will cause the UAV to lose control and crash. When climbing, in contrast to turning, the smaller the climbing angle, the smoother the trajectory while achieving obstacle avoidance. If the climbing angle is too large, it can also cause the UAV to lose control. The constraints are represented as follows:

$$R = \sum_{i=1}^N R_i \leq R_{max} \quad (5)$$

$$\theta = \sum_{i=1}^N \theta_i \geq \theta_{min} \quad (6)$$

In the formulas, R_i is the turning radius of the UAV at the i th track point, R_{max} is the maximum turning radius value, θ_i is the climbing angle of the UAV at the i th track point, and θ_{min} is the minimum climbing angle value.

Based on the above constraint analysis, different constraint terms are selected for combination and coefficient allocation according to different needs. Through single objective or multi-objective optimization, the UAV navigation planning objective function (detailed in Sections 2.2.2 and 2.2.3, respectively) can be obtained, which is an evaluation function for the quality of path points, used to judge the quality of the solutions searched by the navigation planning algorithm.

2.2. UAV Navigation Planning Algorithm Design

2.2.1. End-Cloud Collaboration Navigation Planning Algorithm Architecture

The study designed the architecture of the UAV end-cloud collaborative navigation planning system, as shown in Figure 1. In the background cloud navigation planning part, an improved PSO algorithm is used to plan the approximate track points for the UAVs. In the UAV onboard navigation planning part, an improved multi-objective hybrid swarm intelligence algorithm and B-spline curve are used to be responsible for the local navigation control and path optimization of the UAVs.

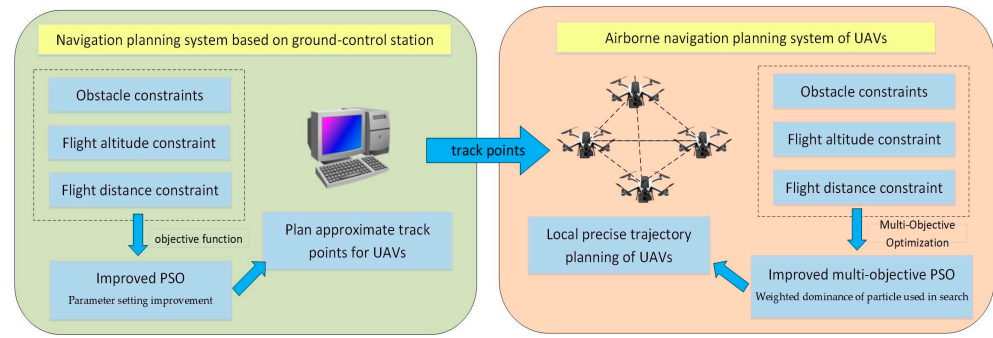


Figure 1. End-cloud collaborative navigation planning system architecture.

The design of the navigation planning algorithm based on the background cloud processor can rely on the high computational power, so the more complex intelligent optimization algorithm can be used. However, the problem of real-time navigation planning due to the limitations of communication links should be taken into consideration, so only the approximate route point planning with lower frequency can be carried out for UAVs. If the communication conditions with front-end UAVs are non-existent, the UAVs can only rely on an airborne intelligent planning algorithm to achieve navigation flight. When the communication link from the front-end to the background cloud is available, the reasonable track points that meet the requirements are planned based on an improved PSO and sent to the front-end UAVs to realize the navigation flight based on the approximate track points.

2.2.2. Background Cloud Navigation Planning Part

Particle swarm optimization (PSO) is an optimization algorithm jointly proposed by Kennedy, Ph.D. of American social psychology, and Eberhart, Ph.D. of electronic engineering in 1995 [24]. In classical PSO, the individual is regarded as a particle without mass and volume, and the target is regarded as the solution of the optimized problem. The movement direction and speed of each particle are affected by their own, as well as group information. The information sharing and cooperation between individuals in the group is used to make the group move toward the direction of the optimal solution, completing the search of the whole particle swarm in the solution space.

The velocity and position update formula of PSO particles are as follows [25]:

$$v_i^t = \omega v_i^{t-1} + c_1 r_1 (pBest_i - x_i^{t-1}) + c_2 r_2 (gBest - x_i^{t-1}) \quad (7)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (8)$$

x_i^t represents the position of the i th particle after t th iteration and v_i^t represents the velocity of the i th particle after the t iteration. ω is the inertia weight factor, c_1 and c_2 are the learning factors, $pBest_i$ represents the historical optimal position of the i th particle itself, and $gBest$ represents the historical optimal position of the entire population. Usually, N_s represents the number of particles and $f(x_i)$ is the evaluation function of the optimized problem. The particle position depends on the value of $f(x_i)$. Each particle performs the following operations: obtaining the particle position x_i^{t-1} after $t - 1$ th iteration, the corresponding $f(x_i^{t-1})$ can be calculated. According to Formulas (1) and (2), the speed and position are updated and the next iteration is carried out to get $f(x_i^t)$. Then, the above operation is repeated until $f(x_i)$ meets the requirements or reaches the maximum number of iterations [26].

Figure 2 is a schematic diagram of solving optimization problems with PSO. The search space is two-dimensional. The global optimal solution is at the black spot. The particles are updated from the initial position to the updated position after iteration, where v_1 is the original velocity of the particle, v_2 is the velocity caused by $pBest$, and v_3 is the velocity caused by $gBest$. The final velocity of particle v is determined by v_1 , v_2 , and v_3 .

together, to make the particles arrive at the updated position from the initial position, and then the speed and position are updated in the same way. The particles will gradually approach the optimal solution position [27].

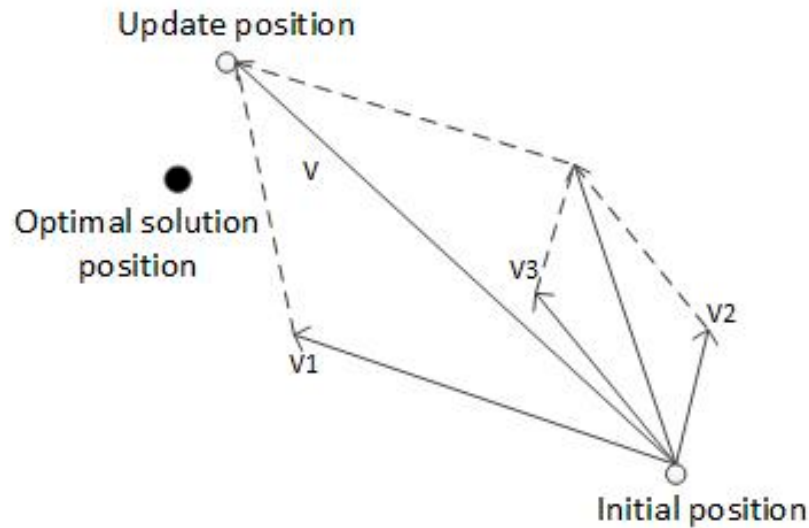


Figure 2. Schematic diagram of PSO iterative updating.

Formulas (1) and (2) are the speed and position update formula of standard PSO, and the main parameters involved include inertia weight ω , learning factors c_1 and c_2 , particle population size N_s , and maximum flight velocity of the particles v_{max} . According to traditional parameter settings, it is easy to encounter situations where the algorithm searches too fast and skips over high-quality solutions, or falls into local optima. Faced with the problem of UAV navigation planning, a series of improvements need to be made to these parameters [28].

- Inertia weight ω

In PSO, small inertia weight is conducive to the local precise search of the algorithm, while large inertia weight is conducive to the fast global search of the algorithm. Therefore, dynamically changing inertia weight can achieve better optimization results than a fixed value of inertia weight. The improvement methods of this study are as follows:

$$\omega = \begin{cases} \omega_{min}, & \delta \geq \lambda_1 \\ \omega_{max}, & \delta \leq \lambda_2 \end{cases} \quad (9)$$

In the above formula, δ is the degree of particle position change after continuous k iterations. If the change of particle position after continuous k iterations exceeds λ_1 , then it can be judged that the algorithm is in the fast global search stage. In such case, ω is made into a smaller value, ω_{min} , to slow down the search speed and prevent the algorithm from missing the optimal solution due to the fast search speed. If the change of particle position does not exceed λ_2 , then it can be judged that the algorithm has fallen into local optimum, in which case ω is made a larger value, ω_{max} , to improve the search speed of the algorithm and make the algorithm jump out of the local optimum. k , λ_1 , and λ_2 are determined by the complexity of the UAV navigation planning problem, and the required search speed and accuracy.

- Learning factors c_1 and c_2

Learning factors c_1 and c_2 give particles the ability to learn from their own experience and the best ones in the population. A small learning factor makes for a particle search repeatedly outside the target area, and a large learning factor makes the particles fly to or even cross the target area quickly. Similar to inertia weight, ω , the adaptive adjustment strategy is segmented based on particle position changes. When the algorithm is in the

global search stage, c_1 is made a larger value, and c_{max} , c_2 a smaller value, c_{min} to prevent the premature convergence of the algorithm. When the algorithm is in the local search stage, c_1 is made a smaller value, c_{min} , and c_2 a larger value, c_{max} , to ensure the diversity of the algorithm during the local search.

$$c_1 = \begin{cases} c_{max}, & \delta \geq \lambda_1 \\ c_{min}, & \delta \leq \lambda_2 \end{cases} \quad (10)$$

$$c_2 = \begin{cases} c_{min}, & \delta \geq \lambda_1 \\ c_{max}, & \delta \leq \lambda_2 \end{cases} \quad (11)$$

- Population size and topological structure second bullet

For PSO, a large population size means that the particles have strong cooperation performance and high global search ability, but the search time of the algorithm is long. However, a small population size will reduce the available global information of the algorithm, which will lead to premature convergence and fall into local optimization. The population topology of PSO represents the way that a single particle connects with other particles, and represents which particles can share information and cooperate with the iteration. Generally, topological structures can be divided into global and local types. Global topology means that each particle can share and exchange information with any other particle, which will speed up the convergence of the algorithm, but it is easy to fall into the local optimum. On the contrary, local topology can only exchange and share information with neighboring particles. This topology will slow the convergence of the algorithm, but it is not easy to fall into local optimum. The two population topologies are shown in Figure 3. The disadvantage of the fixed population size and topology strategy is that it cannot balance the contradiction between a global search and local search, and cannot achieve a good optimization effect in the UAV navigation planning problem [29].

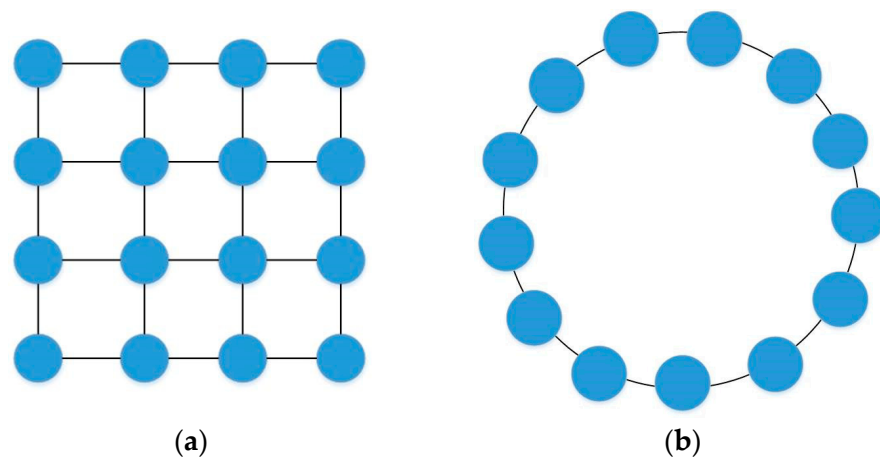


Figure 3. Population topology of PSO: (a) global topology; (b) local topology.

The study proposed a population size and topology improvement strategy that can adaptively change population characteristics based on the current iteration update status of particles, as shown in Figure 4.

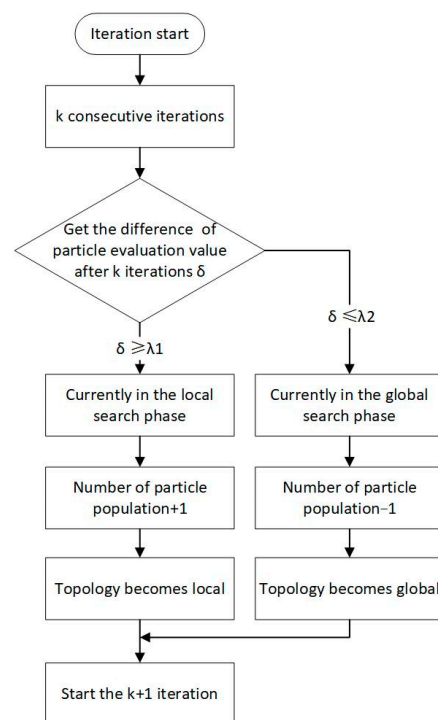


Figure 4. Population size and topology improvement strategies.

- Maximum particle flying speed, v_{max}

The maximum speed of traditional PSO is a fixed value, but it is difficult to consider both algorithm convergence speed and search accuracy at the same time. Therefore, a speed regulation mechanism improvement strategy is proposed as follows:

$$v_{max} = \begin{cases} v_1, & \delta \geq \lambda_1 \\ v_2, & \delta \leq \lambda_2 \end{cases} \quad (12)$$

In the above formula, $v_1 > v_2$. The speed regulation mechanism can meet the requirements of both a global fast search and local accurate search, and can flexibly adjust parameters to meet the versatility of the algorithm in navigation planning.

Using the two-dimensional Ackley function as the objective function, a performance comparison test was conducted on the traditional PSO and the improved PSO. Figure 5 displays the comparison test results, where the red dot is the true maximum and the blue dot is the searched maximum. The test results show that the maximum value searched using an improved PSO is more concentrated and distributed around the true value of 0, while the maximum value searched using traditional PSO is more dispersed, indicating that an improved PSO has better optimization effects.

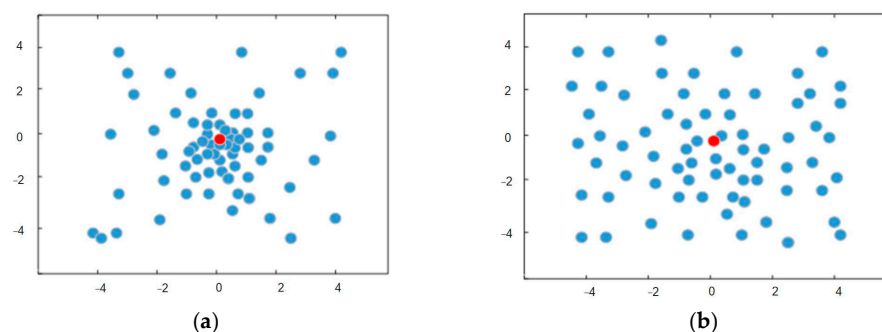


Figure 5. Comparison of algorithm optimization effects: (a) by an improved PSO; (b) by traditional PSO.

Section 2.1 has already described the constraints of navigation planning. The article combines flight distance, flight altitude, and distance from UAV to obstacles into an objective function, with the distance between UAVs, turning radius, and climbing angle as boundary constraints. The resulting navigation planning evaluation function is shown by the following formula:

$$\begin{cases} f = \sum_i (\varphi_1 L_i - \varphi_2 \lambda_i - \varphi_3 h_i), i = 1 \dots N \\ \rho_{min} < \rho_i < \rho_{max}, i = 1 \dots N \\ R_i < R_{max}, i = 1 \dots N \\ \theta_i > \theta_{min}, i = 1 \dots N \end{cases} \quad (13)$$

In the above formula, φ_1 , φ_2 , and φ_3 are the weight coefficients. The influence of various constraints on the navigation planning evaluation function is changed by using artificial weighting methods to adjust their sizes. During the iteration process, the point that minimizes the value of f is the best track point.

The background cloud navigation planning algorithm based on an improved PSO is shown in Figure 6.

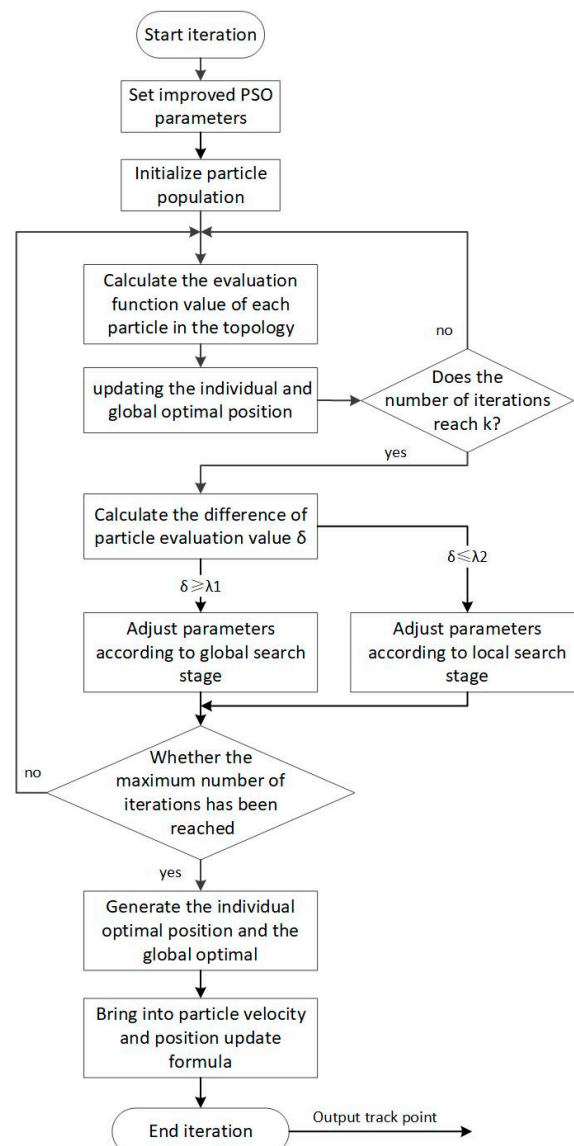


Figure 6. Background cloud navigation planning algorithm based on an improved PSO.

2.2.3. UAV Onboard Navigation Planning Part

The UAV onboard navigation planning part mainly solves the autonomous local path planning and navigation control problems of UAVs after the background cloud provides them with approximate flight paths. Moreover, UAV onboard processing platforms are often limited by low computational power, and due to the need for fast algorithm convergence speed in the local navigation control of UAVs, the algorithms are more often to fall into local optimization. Although the above series of improvements to PSO can help solve the local optimization problem of navigation planning, making it easier for the algorithm to search for high-quality solutions, it also increases the computational burden and is not suitable for onboard navigation planning parts with high real-time requirements; hence, the algorithm and optimization method that have fast search speed and low computational complexity are necessarily needed. Based on the advantages of PSO, the study used multi-objective optimization and tabu search to solve the UAV onboard navigation planning problem [30].

(1) Improved multi-objective PSO

The navigation planning evaluation function of Formula (7) is to add various constraints and transform it into single-objective optimization. This approach can obtain an optimal solution to a certain extent when the background cloud computer has sufficient resources, but it cannot meet the requirements of a low computational burden and high real-time performance for UAV onboard navigation planning. Therefore, multi-objective optimization can be adopted to solve this problem.

Multi-objective optimization (MOP) problems can generally be defined as the optimization problem of jointly optimizing multiple objective functions under given constraints. The definition of multi-objective optimization can be expressed in the following standard form [31]:

$$\begin{cases} \mathbf{y} = \min[f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})] \\ \text{s.t. } g_i(\mathbf{x}) \leq 0 \end{cases} \quad (14)$$

In the above formula, $\mathbf{y} = (y_1, y_2, \dots, y_m)$ represents the target function, $\mathbf{x} = (x_1, x_2, \dots, x_n)$ represents decision variables to be optimized, and $\{g_1(\mathbf{x}) \leq 0, g_2(\mathbf{x}) \leq 0, \dots, g_p(\mathbf{x}) \leq 0\}$ represents constraints of the decision variable $\mathbf{x}$.

The difference between multi-objective optimization and single-objective optimization is that a multi-objective optimization problem does not require finding a unique optimal solution; therefore, the complexity and computational time of the algorithm will be reduced and it is more suitable for the onboard navigation planning processing of UAVs with high real-time requirements. In the process of solving multi-objective optimization problems, the key is to find some “better solutions” that can satisfy the situation where each objective function achieves a better performance. This set of solutions is called the Pareto optimal solution set. Pareto domination is defined as follows:

If at least one objective function of the feasible solution $\mathbf{p}$ is better than that of the feasible solution $\mathbf{q}$, and all objective functions of $\mathbf{p}$ are no worse than that of the individual $\mathbf{q}$, that is:

$$\begin{cases} \forall i \in \{1, 2, \dots, m\}, f_i(\mathbf{p}) \leq f_i(\mathbf{q}) \\ \exists i \in \{1, 2, \dots, m\}, f_i(\mathbf{p}) < f_i(\mathbf{q}) \end{cases} \quad (15)$$

Then, the feasible solution $\mathbf{p}$ Pareto dominates the feasible solution $\mathbf{q}$. Pareto dominance in the application of high-dimensional multi-objective optimization problems will cause a significant increase in the number of non-dominated solutions, making it unable to select a better solution from among them, and the algorithm unable to converge. Therefore, the study proposed a new type of dominance-weighted dominance—combined with the traditional Pareto dominance, that can reduce the computational burden of the algorithm and optimize the performance.

The definition of weighted dominance is given here:

For all objective function values for the problem $f_1(x), f_2(x), \dots, f_k(x)$, make:

$$F_i(x_1, x_2) = f_i(x_1) - f_i(x_2) + \rho_i \sum_{j=1}^k f_j(x_1) - f_j(x_2) \quad (16)$$

In the above formula, x_1 and x_2 are both feasible solutions to the problem and ρ_i is the weighted dominance factor. If:

$$\begin{cases} \forall i, F_i(x_1, x_2) \leq 0 \\ \exists i, F_i(x_1, x_2) < 0 \end{cases} \quad (17)$$

then x_1 is weighted domination x_2 . The weighted dominance factor ρ_i is related to the weight ratio of the i th objective function. If the weight of the i th objective function is larger, the weighted dominance factor is smaller and vice versa. Compared to Pareto domination, weighted domination is easier to achieve between particles. This makes it easier for the algorithm to find a better solution through weighted dominance, rather than using a large amount of computing resources to search for the Pareto optimal solution.

An important feature of PSO based on multi-objective optimization is to store all non-dominated solutions generated in a set (called external file) after each iteration according to the dominant relationship between the particles. The global optimal particle position $gBest$ that guides the flight of a particle population is selected from the external file. Based on the weighted dominance, this study proposed an improved external file update strategy, as shown in Figure 7.

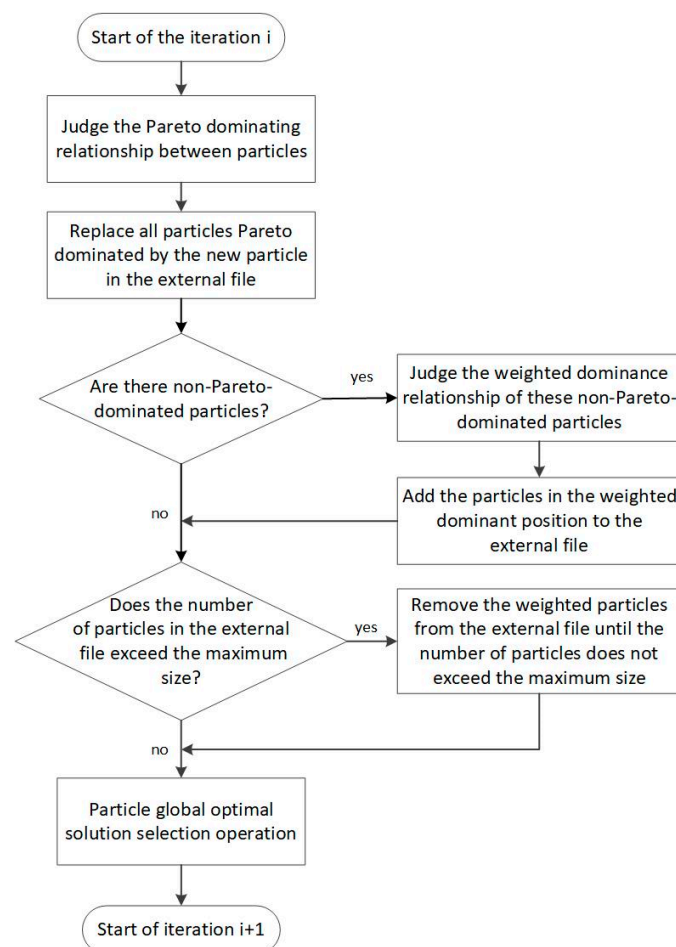


Figure 7. Improved external file update strategy.

The global optimal particle location selection strategy is as follows: If there are particles in the Pareto-dominant position added to the external file for consecutive k generations, it is judged that the diversity of the population is qualified, while the convergence is not qualified. The particles Pareto-dominated in the external file are weighed and sorted, and the particle with the highest ranking in the sorting is selected as the global optimal particle. If no Pareto-dominated particles have been added to the external file for consecutive k generations, it is judged that the convergence of the population is qualified, but the diversity is unqualified, and it is possible to fall into a local optimum. In such case, one of the particles in the external file that is weight-dominated, but not Pareto-dominated, is randomly selected as the global optimal particle.

This external file update strategy can greatly reduce the number of particles entering the external file in each iteration, which is conducive to reducing the computational strength of the algorithm, and the adaptive global optimal particle location selection strategy can balance the relationship between diversity and convergence of solutions, better meeting the needs of UAV navigation planning.

The external file solves the problem of selecting the global optimal location of particles, but the selection of the individual optimal location of particles is still different from the single-objective PSO. It is not to compare the magnitude of the evaluation function values between two particles to determine the good and bad, but to compare the weighted dominance relationship between two particles, which can be divided into the following three situations:

- If the individual optimal particle in the previous iteration weighted dominates the new particle, the individual optimal particle remains unchanged and the optimal position of the particle is not updated;
- If the individual optimal particle in the previous iteration is weight-dominated by the new particle, the new particle becomes the individual optimal particle, and the historical optimal position of the particle is updated;
- If the two are not weight-dominated to each other, a 50% probability is used to choose whether to update the optimal position of the particle or not.

The external file update strategy of the multi-objective particle swarm optimization and the selection strategy of particle global optimal and individual optimal position improvement have been completed. Combined with the hybrid swarm intelligence algorithm below, it can further reduce the computational burden of UAV onboard navigation planning and improve the solution efficiency.

(2) UAV onboard hybrid swarm intelligent algorithm and trajectory optimization

Based on the multi-objective PSO, this section introduced the neighborhood subset of the tabu search and the tabu list, further reducing the computational complexity of UAV onboard navigation planning algorithms, while optimizing the trajectory to meet the flight requirements of UAVs.

The tabu search is a neighborhood search algorithm, characterized by the fact that it is not necessary to search all the solutions in the neighborhood structure every time in the iteration, but to randomly select a subset of the neighborhood to search, the spatial range of each iteration search is small, so it can reduce the computational intensity of the algorithm. Another feature of the tabu search is the setting of the tabu list, which is used to store the generated local optimal solution after each iteration. The solutions in the tabu list will not be searchable in subsequent iterations, and reduce the search solution space, while preventing the algorithm from falling into local optimum. After several iterations, if the aspiration criterion is met, in other words, the evaluation value of the tabu object is better than any solution of the neighborhood subset in that iteration, it will be released and can participate in the next iteration search.

The objective terms of the evaluation function for the background cloud navigation planning of UAVs mentioned in the previous section refer to the length of the flight path L , the distance between the obstacle and the target λ , the flight altitude h , the average distance between UAV ρ , turning radius R , and climbing angle θ . In this section, the UAV

onboard navigation planning adopts a multi-objective optimization approach, with the first three items as the objective items and the last three item as the constraint conditions. Therefore, the multi-objective optimization modeling of UAV onboard navigation can be represented as:

$$\begin{cases} \mathbf{y} = \min \left[\sum_{k=1}^M L_k, \sum_{k=1}^M \lambda_k, \sum_{k=1}^M h_k \right] \\ \rho_{\min} < \rho_k < \rho_{\max}, k = 1, 2, \dots, M \\ R_k < R_{\max}, k = 1 \dots M \\ \theta_k > \theta_{\min}, k = 1 \dots M \end{cases} \quad (18)$$

The flowchart of the UAV onboard hybrid swarm intelligent algorithm is shown in Figure 8.

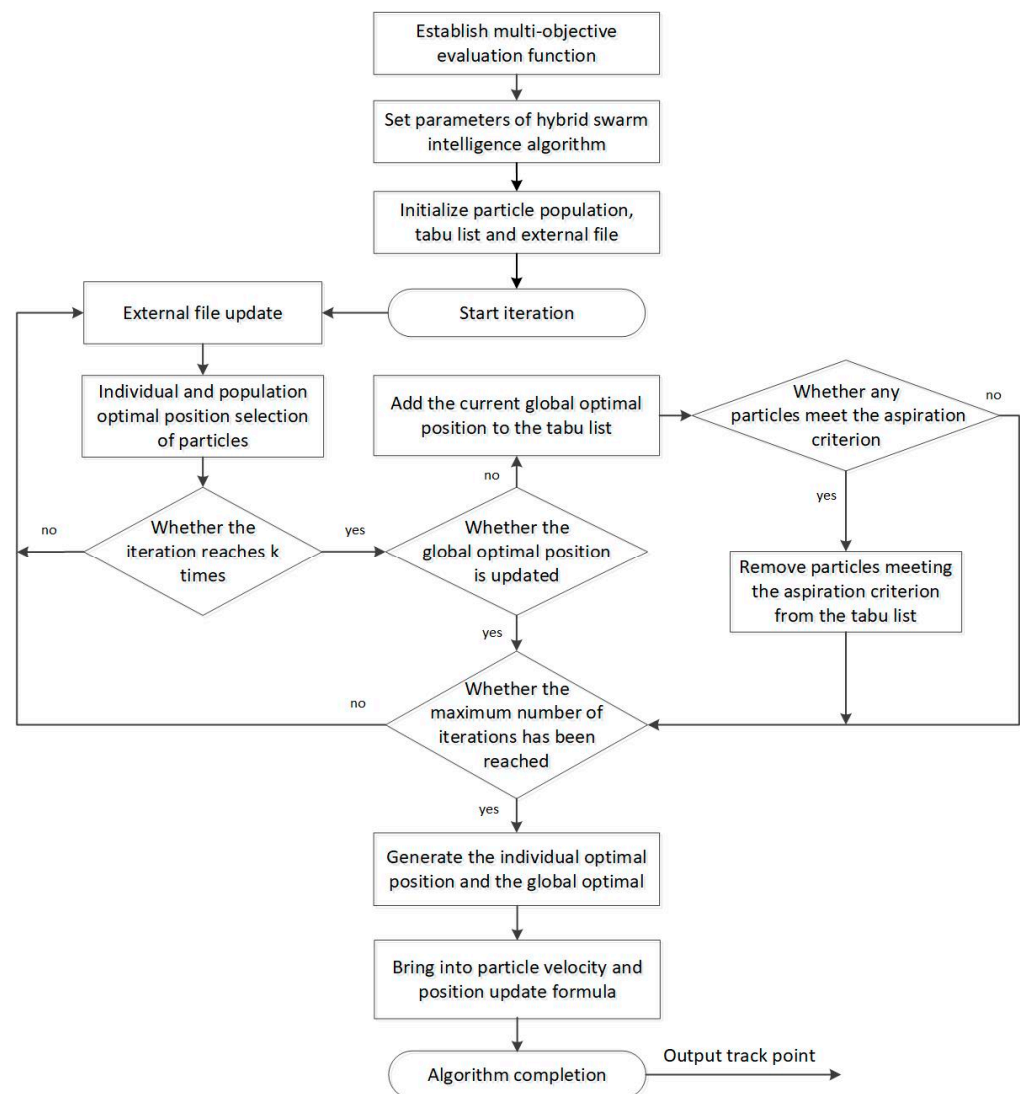


Figure 8. The flowchart of the UAV onboard hybrid swarm intelligent algorithm.

Usually, after obtaining the approximate path points planned by the background cloud, the UAVs conduct local navigation planning between each path point to form trajectories. However, due to the flight control system limitation of UAVs, the trajectories cannot be directly used by the UAVs. Therefore, it is necessary to optimize the trajectories obtained by the UAV onboard hybrid swarm intelligence algorithm, and generate smooth trajectories.

The study optimized the planned trajectory based on B-spline curves. B-spline curves can be locally reconstructed from the original trajectory, without changing the overall

shape of the trajectory. The B-spline curve is determined by the B-spline basis function. Given $n + 1$ track points and a node vector in space $U = [u_0, u_1, \dots, u_{n+k+1}]$, the k -power B-spline curve is represented as:

$$P(u) = \sum_{i=0}^n N_i^k(u) P_i \quad (19)$$

In the above formula, P_i is the vertex of the feature polygon that forms the B-spline curve and $N_i^k(u)$ is the B-spline basis function, using the Cox–deBoor recursive formula as [32]:

$$\begin{cases} N_i^0(u) = \begin{cases} 1, & u_i \leq u \leq u_{i+1} \\ 0, & \text{otherwise} \end{cases} \\ N_i^k(u) = \frac{u - u_i}{u_{i+k} - u_i} N_i^{k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1}^{k-1}(u) \end{cases} \quad (20)$$

In this study, the cubic B-spline curve is used to smooth the trajectory, where $k = 3$, a B-spline curve can be constructed by every four adjacent track sampling points. By using the formula, it can be obtained that for the four adjacent points P_0, P_1, P_2, P_3 , there is

$$P(U) = \frac{1}{6} \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (21)$$

The specific steps for using B-spline curves to smooth the trajectory are shown in Figure 9.

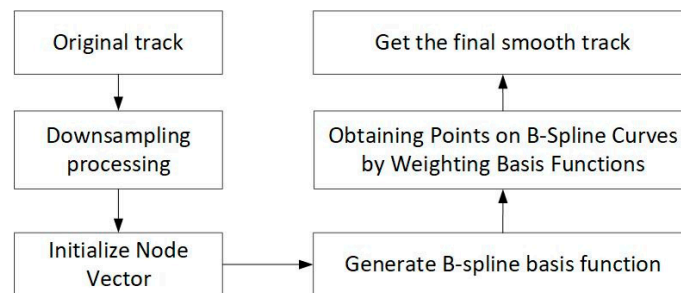


Figure 9. The specific steps for using B-spline curves to smooth the trajectory.

3. Experimental Design and Analysis

For the UAV end-cloud collaborative navigation planning system architecture and improved algorithm proposed in this study, this part carried out simulation comparison experiments and analysis, and used the rotor drones and background cloud navigation planning platform to carry out actual flight experiment verification.

3.1. Simulation Experiment Verification and Analysis

Based on the MATLAB environment, the scenario of several UAVs avoiding obstacles and reaching the target point position after navigation planning was simulated. Navigation planning experiments based on the end-cloud collaborative improved PSO (E-CPSO) proposed in this paper, with traditional PSO onboard, two types of improved PSO onboard (improved PSO with parameter changes, PSO combined with genetic algorithm), clustering-based algorithm (CA) [33], and ant colony algorithm (ACA) [34] were conducted, and the effectiveness of navigation planning was compared and analyzed.

3.1.1. Environmental Modeling and Algorithm Parameter Setting

The modeling of the environment includes three parts: flight range, threat area, and obstacle area [35]. The flight range is set to the Cartesian coordinate system area. Threat

areas generally refer to electromagnetic interference areas and enemy detection areas. A hemispherical model can be used to model the threat area, which is mathematically described as:

$$\begin{cases} W_i(x, y, z) = \sum_i (x - x_i)^2 + (y - y_i)^2 + z^2 = r_i^2 \\ z \geq 0 \end{cases} \quad (22)$$

In the formula, $(x_i, y_i, 0)$ represent the center of the threat area, $W_i(x, y, z)$ represents the i th threat area, and r_i represents the radius of the threat area. The modeling of obstacle areas adopts the mountain peak model, which is mathematically described as:

$$z(x, y) = \sum_i h_i \exp\left[-\left(\frac{x - x_i}{x_{si}}\right)^2 - \left(\frac{y - y_i}{y_{si}}\right)^2\right] \quad (23)$$

In the formula, (x_i, y_i) represents the geographic center coordinates of the mountain peak, (x, y) represents the coordinates of each point in the terrain projected onto the plane, h_i represents the height of the i th mountain, x_{si}, y_{si} represent the slope vectors of the peaks in the x -axis and y -axis directions, and $z(x, y)$ represents the height of each point in the terrain.

The terrain parameter settings are shown in Table 1.

Table 1. Parameters of the terrain.

	Flight Range Parameters (m)	Obstacle Area Parameter OPR ¹ (m)	Threat Area Parameter TPR ¹ (m)
Parameters	2000 × 2000 × 1000	(600, 500, 100) (500, 1500, 200)	(1000, 1000, 400)

¹ where OPR represent the center positions and bottom radius (x, y, r) of the obstacle area, and TPR represent the center positions and radius (x, y, r) of the threat area.

The terrain environment model is shown in Figure 10. The red part represents the threat area, and the green parts represent the peaks.

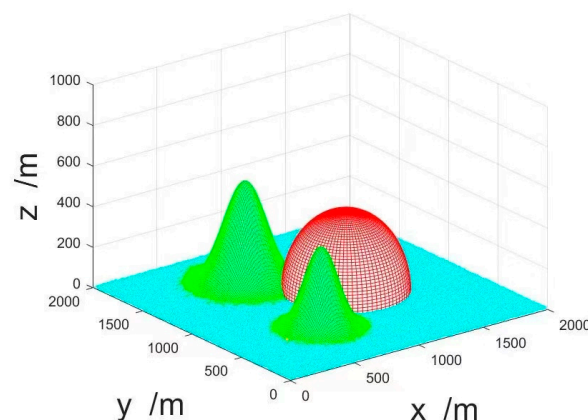


Figure 10. The terrain environment model.

The number of UAVs was set to 5, which arrange at $(0, 0, 0)$, and the number of task target points was set to 5, which were spread at $(2000, 1200, 900)$, $(2000, 700, 750)$, $(2000, 680, 280)$, $(2000, 1100, 140)$, $(2000, 1400, 500)$. The results after task allocation are shown in Figure 11. Where * are the target points after task allocation.

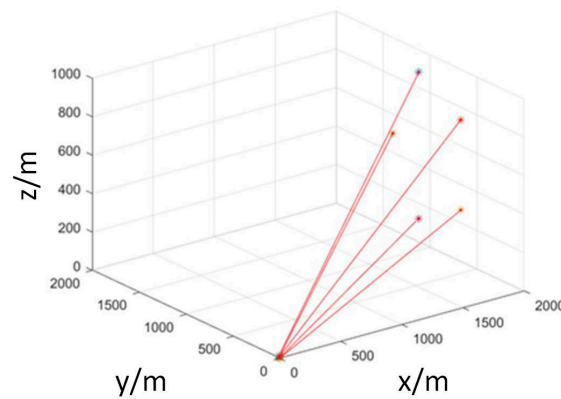


Figure 11. UAV target point task allocation.

As a comparison, the traditional PSO as UAV navigation planning algorithm was unilaterally set according to the most common way, inertia weight $\omega = 0.6$, learning factor $c_1 = c_2 = 2$, number of particle populations $N_s = 300$, maximum number of iterations $T = 200$, maximum particle flight speed $v_{max} = 2$. The parameter settings of the improved PSO with parameter changes (IPSO) was according to the onboard navigation planning section of the article as follows. The parameter of the PSO combined with the genetic algorithm (GAPSO) was set in a traditional way: the crossover probability was 0.3 and the mutation probability was 0.1. Other parameters were set according to traditional PSO. The parameters of clustering-based algorithm (CA) were as follows: the number of clusters $k = 8$, the selection probability $\beta = 0.7$, and the population size and evolution times were the same as those of IPSO. The parameters of the ant colony algorithm (ACA) were as follows: the pheromone factor weight $\alpha = 1$, the heuristic factor weight $\gamma = 7$, and the pheromone evaporation and enhancement coefficient $\mu_1 = 0.3, \mu_2 = 1$.

The parameter settings for E-CPSO proposed in this article were as follows. In the dynamic parameter adjustment strategy, considering the actual needs of UAV navigation planning, $\lambda_1 = 0.3, \lambda_2 = 0.1, k = 5$, the maximum and minimum inertia weight $\omega_{max} = 0.9, \omega_{min} = 0.4$, the maximum and minimum learning factors $c_{max} = 3.5, c_{min} = 0.5$, the initial population size $N_s = 300$, and the maximum particle flight speed $v_1 = 3, v_2 = 1.5$. The maximum and minimum average distance constraint between UAVs $\rho_{max} = 300, \rho_{min} = 50$, the weight coefficients in single-objective optimization functions $\varphi_1 = 0.2, \varphi_2 = 0.3, \varphi_3 = 0.1$, the initial external file size in multi-objective optimization $M_s = 50$, the weighted dominance coefficient $\rho_1 = 1 - \varphi_1 = 0.8, \rho_2 = 1 - \varphi_2 = 0.7, \rho_3 = 1 - \varphi_3 = 0.9$.

3.1.2. Experimental Steps and Result Analysis

The testing steps are as follows:

- Step1: Start the simulation software and select the UAV end-cloud collaborative navigation planning algorithm;
- Step2: Set environmental conditions and task target points and allocate them;
- Step3: Set the parameters of the E-CPSO navigation planning algorithm;
- Step4: Start UAV background cloud and onboard navigation planning simulation, obtain the approximate trajectory and precise trajectory of UAVs;
- Step5: Set the parameters of traditional PSO, IPSO, GAPSO, CA and ACA, and start the UAV navigation planning simulation;
- Step6: Observe and record the navigation planning results of the algorithms;
- Step7: Draw graphs of the number of iterations and evaluation function separately and conduct comparative analysis;
- Step8: End the experiment.

The results of UAV navigation planning based on the algorithms are shown in Figure 12. The black dots and connecting lines represent the trajectory results of the navigation plan. From the figures, it can be intuitively seen that the navigation planning

result based on E-CPSO was smoother compared to the others, and the track length of each UAV was shorter.

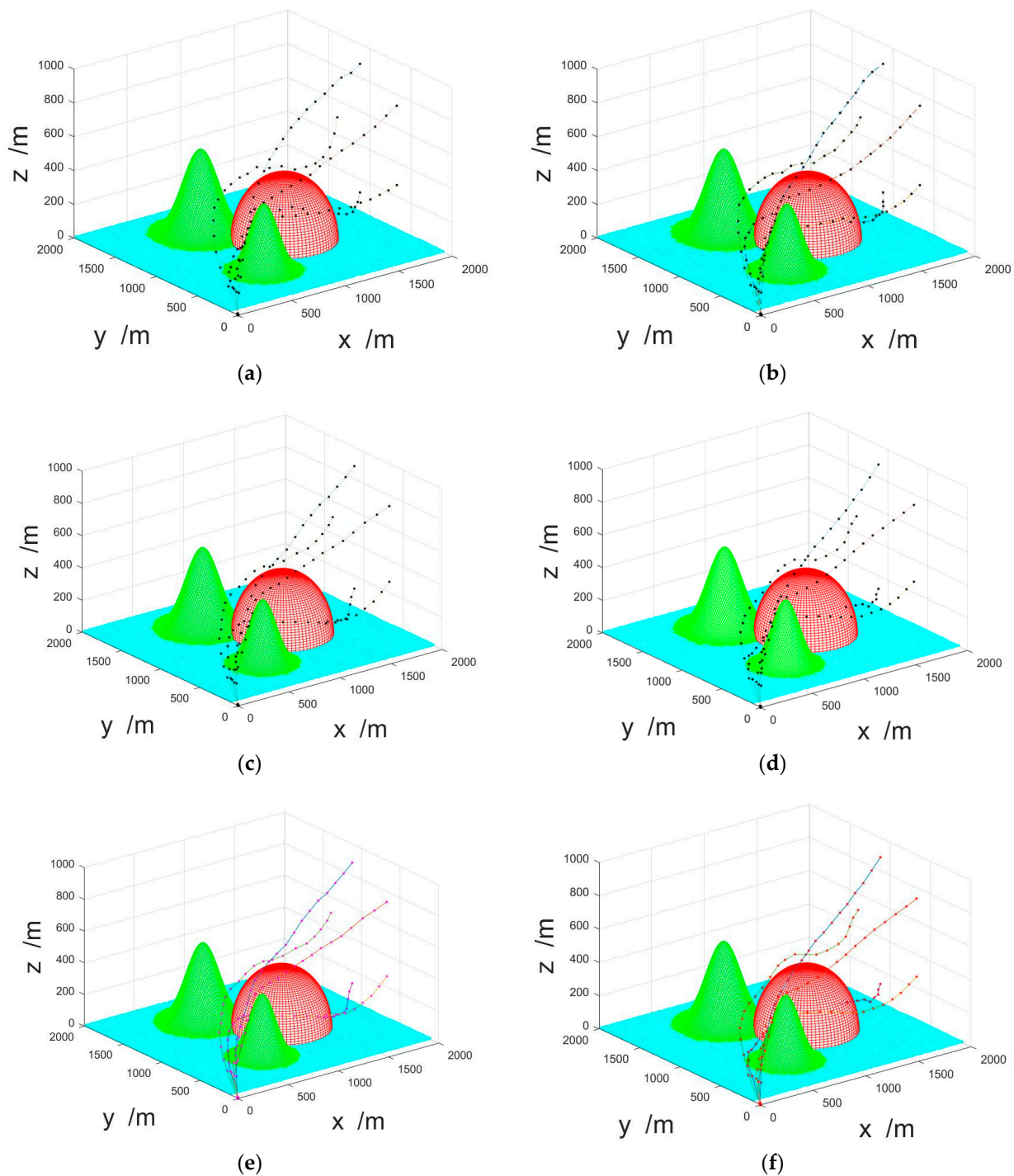


Figure 12. The results of UAV navigation planning based on the four algorithms: (a) E-CPSO; (b) traditional PSO; (c) IPSO; (d) GAPSO; (e) CA; (f) ACA.

In order to evaluate the performance of the E-CPSO, the relationship between the number of iterations and the objective function value of the four algorithms were compared as shown in the following figure. As the algorithm proposed in this study involves multi-objective optimization, its objective function value was directly calculated from the evaluation value of the track point position. The objective function values of ACA and CA were also calculated based on each track point position.

From Figure 13a, it can be seen that under the same environmental conditions, number of UAVs and location of task target points, the objective function of UAV traditional PSO, IPSO, and GAPSO tended to flatten out at around 80, 90, and 100 iterations. The objective function of the E-CPSO began to flatten after about 50 iterations, indicating that the convergence speed of the latter was faster than that of the formers. Overall, the objective function value of the E-CPSO was smaller and separately decreased by approximately 42%, 25%, and 13% compared to the traditional PSO, IPSO, and GAPSO when convergence was completed, which indicated that the algorithm proposed in this article had better performance in UAV navigation planning. Figure 13b shows that compared with CA and ACA, E-CPSO still had advantages in the rate of convergence and objective function value when convergence was completed.

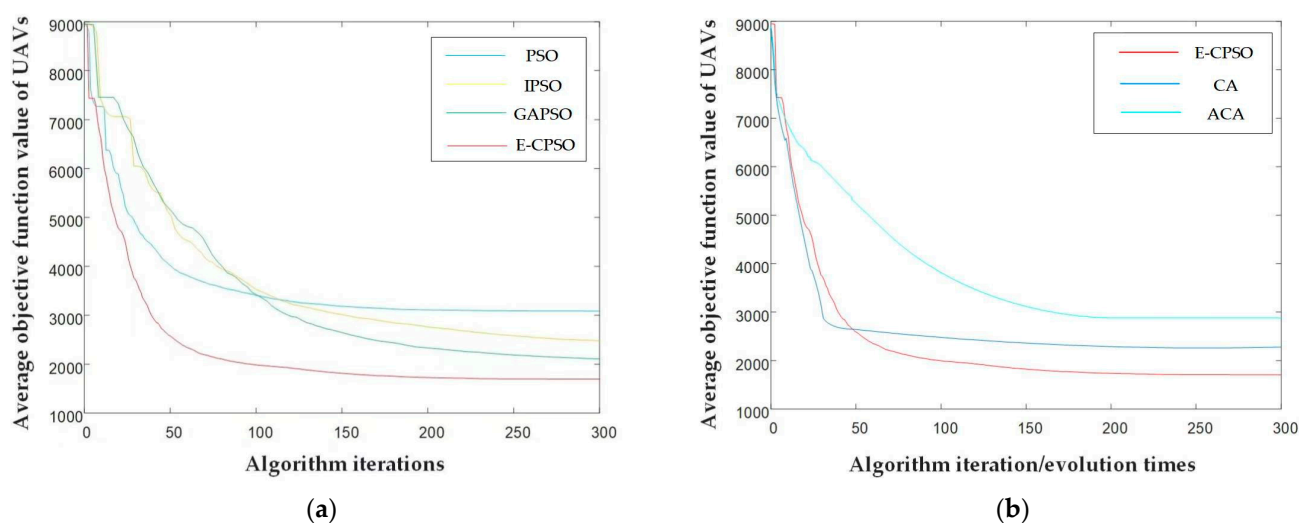


Figure 13. The relationship between the number of iterations and the average objective function value of UAVs of the algorithms: (a) E-CPSO compared to PSO, IPSO, and GAPSO; (b) E-CPSO compared to CA and ACA.

The terrain parameters and the number of UAVs were changed, and the performance of the algorithms tested separately. The results are shown in Table 2. From the data in the table, it can be seen that the E-CPSO can complete navigation planning of UAVs in different terrain environments, achieved better target evaluation values for track points, and got a lower average number of iterations compared to traditional PSO, IPSO, GAPSO, CA, and ACA. When the number of UAVs increased to 10, the traditional PSO and GAPSO were no longer able to complete navigation planning in complex environments, while the E-CPSO was still valid, though increasing the number of iterations. The results showed that the effectiveness, timeliness, and adaptability of the algorithm proposed in this article are superior than that of the other algorithms.

Table 2. Algorithm performance comparison under different conditions.

Terrain Parameter (m)	Algorithm	IBC ¹ of 5 UAVs	FVC ¹ of 5 UAVs	IBC ¹ of 10 UAVs	FVC ¹ of 10 UAVs
1 threat area, 2 obstacle areas	PSO	82	3128.6	113	6034.5
	E-CPSO	56	1834.4	86	3179.6
	IPSO	88	2465.2	121	4528.3
	GAPSO	96	2104.3	134	4019.1
	CA	72	3104.6	98	6194.4
	ACA	105	2923.5	107	5478.5

Table 2. Cont.

Terrain Parameter (m)	Algorithm	IBC ¹ of 5 UAVs	FVC ¹ of 5 UAVs	IBC ¹ of 10 UAVs	FVC ¹ of 10 UAVs
2 threat areas, 1 obstacle area	PSO	78	3364.7	121	6823.1
	E-CPSO	52	2023.2	92	4435.3
	IPSO	84	2675.4	125	5287.2
	GAPSO	95	2392.9	138	4963.4
	CA	74	2989.6	111	5892.3
	ACA	96	3045.2	103	6102.2
2 threat areas, 2 obstacle areas	PSO	134	3578.6	--	--
	E-CPSO	112	2149.1	157	4752.8
	IPSO	146	2538.3	172	6272.2
	GAPSO	151	2414.7	--	--
	CA	128	3192.2	169	6490.5
	ACA	123	3234.7	174	6189.8

¹ where IBC represents iterations at the beginning of convergence and FVC represents objective function value at convergence.

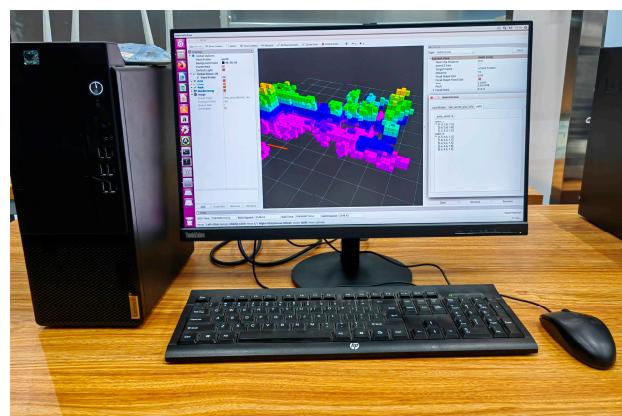
3.2. Verification of Actual Flight Experiments of UAVs

The actual flight tests by using drones and a ground control station computer loaded with the UAV end-cloud collaborative navigation planning algorithm were conducted.

The UAVs used in the experiment were small quadcopter drones, equipped with visual sensors and inertial sensors to achieve environmental perception and self-positioning, along with equipped with communication equipment to receive control commands and transmit image information to the ground station. The patrol speed was set to 3 m/s, the flight speed during obstacle avoidance was 1 m/s, and the flight task time did not exceed half an hour. Figure 14 shows the equipment used in the flight testing.



(a)



(b)

Figure 14. Actual flight testing equipment: (a) drones; (b) ground control station computer and software.

The flight test site was selected as an indoor environment of 20 m × 15 m × 4 m, which included two simulated obstacles. The drones flew from the starting point to the target point according to the calculated trajectory of the algorithm. Figure 15 shows the indoor drone flight testing site.



Figure 15. The indoor drone flight testing site.

The specific steps of the experiment are as follows:

- Step1: Start drones and ground control station loaded with the UAV end-cloud collaborative navigation planning algorithms;
- Step2: The ground control station calculates the track point position and sends it to the drones based on prior environmental information;
- Step3: The drones fly toward the target trajectory point position, continuously calculating and optimizing their own trajectory during the flight process, and transmit the trajectory and observed environmental information back to the ground station;
- Step4: The ground station further calculates the trajectory points based on the environmental information transmitted by the drone and sends them to the drone;
- Step5: Repeat step 3 and 4 until the drones complete obstacle avoidance and reach the final target point, and display the dynamic trajectory of the drones.

The test trajectory results of navigation planning of the two drones were, respectively, displayed in rviz as shown in Figure 16. From the figure, it can be seen that the two drones loaded with the E-CPSO started from the starting point, flew along the trajectory planned by the algorithm, avoided obstacles and reached the task area, and ultimately reached the target point. The experimental results showed that the algorithm is correct and feasible, and can meet the needs for drone navigation planning.

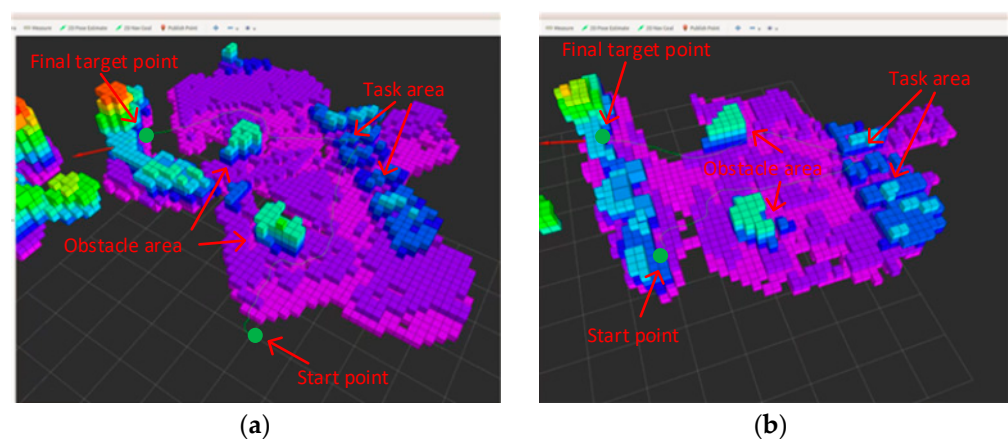


Figure 16. Drone flight trajectory in rviz: (a) flight trajectory of drone 1; (b) flight trajectory of drone 2.

4. Discussion

The study proposed an algorithm of UAV end-cloud collaborative navigation planning, combining the onboard navigation planning part of UAVs with the background control station part. In addition, an improved PSO for the navigation control onboard and a multi-objective hybrid swarm intelligent navigation planning algorithm for the navigation control of background were, respectively, designed.

For the algorithm proposed in the article, simulation testing and actual flight experiment verification of UAVs were conducted. The simulation results showed that the algorithm converged with lower objective function values and fewer iterations compared to traditional PSO, IPSO, GAPSO, CA, and ACA navigation planning methods, indicating that the optimization ability and timeliness of the algorithm were significantly improved, and the actual flight test results of UAVs showed that the algorithm can meet the needs of UAV navigation planning in mission and obstacle environments.

There are still some shortcomings in the application of the algorithm in navigation planning under the limited communication resources of UAVs. Follow-up studies will focus on the actual communication situation and other constraints under practical application conditions to improve the algorithm, conducting actual flight tests with more drones in multi-task scenarios.

Author Contributions: Conceptualization, B.Y.; Formal analysis, B.Y.; Methodology, H.X.; Project administration, B.Y.; Resources, Q.Y.; Software, H.X.; Supervision, Q.Y.; Validation, H.X. and C.H.; Writing—original draft, H.X.; Writing—review and editing, C.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Oh, D.; Han, J. Smart Search System of Autonomous Flight UAVs for Disaster Rescue. *Sensors* **2021**, *21*, 6810. [\[CrossRef\]](#)
- Han, S.I.; Baek, J. Optimal UAV Deployment and Resource Management in UAV Relay Networks. *Sensors* **2021**, *21*, 6878. [\[CrossRef\]](#)
- Zhang, H.; Wu, S.; Feng, O.; Tian, T.; Huang, Y.; Zhong, G. Research on Demand-Based Scheduling Scheme of Urban Low-Altitude Logistics UAVs. *Appl. Sci.* **2023**, *13*, 5370. [\[CrossRef\]](#)
- Costa, L.S.; Sano, E.E.; Ferreira, M.E.; Munhoz, C.B.R.; Costa, J.V.S.; Rufino Alves Júnior, L.; de Mello, T.R.B.; da Cunha Bustamante, M.M. Woody Plant Encroachment in a Seasonal Tropical Savanna: Lessons about Classifiers and Accuracy from UAV Images. *Remote Sens.* **2023**, *15*, 2342. [\[CrossRef\]](#)
- Wang, S.; Jiang, Z.; Bao, X. Autonomous Trajectory Planning Method for Multi-UAV Collaborative Search. In Proceedings of the 2021 5th International Conference on Automation, Control and Robots (ICACR), Nanning, China, 25–27 September 2021; pp. 84–88.
- Goel, U.; Varshney, S.; Jain, A.; Maheshwari, S.; Shukla, A. Three dimensional path planning for UAVs in dynamic environment using glow-worm swarm optimization. *Procedia Comput. Sci.* **2018**, *133*, 230–239. [\[CrossRef\]](#)
- Ju, C.; Son, H.I. Multiple UAV Systems for Agricultural Applications: Control, Implementation, and Evaluation. *Electronics* **2018**, *7*, 162. [\[CrossRef\]](#)
- Tong, P.; Yang, X.; Yang, Y.; Liu, W.; Wu, P. Multi-UAV Collaborative Absolute Vision Positioning and Navigation: A Survey and Discussion. *Drones* **2023**, *7*, 261. [\[CrossRef\]](#)
- Sadiq, M.S.; Ruan, C.; Nawaz, H.; Ullah, S.; He, W. Horizontal Polarized DC Grounded Omnidirectional Antenna for UAV Ground Control Station. *Sensors* **2021**, *21*, 2763. [\[CrossRef\]](#)
- Li, J.; Ran, M.; Xie, L. Efficient trajectory planning for multiple nonholonomic mobile robots via prioritized trajectory optimization. *IEEE Robot* **2021**, *6*, 405–412.
- Suparnunt, C.; Boonvongsobhon, C.; Baig, F.E.; Leelalerthpat, P.; Hematulin, W.; Jarawan, T.; Kamsing, P. Practical Parallel of Autonomous Unmanned Aerial Vehicle by Mission Planner. In Proceedings of the IGARSS 2022—2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia, 17–22 July 2022; p. 7831.
- Virbora, N.; Sokoeun, U.; Saran, M.; Sovicheyratana, S.; Channareth, S.; Saravuth, S. Implementation of Matrix Drone Show Using Automatic Path Generator with DJI Tello Drones. In Proceedings of the 2022 International Conference on Engineering and Emerging Technologies (ICEET), Kuala Lumpur, Malaysia, 27–28 October 2022; p. 1.
- Ouahouah, S.; Bagaa, M.; Prados-Garzon, J.; Taleb, T. Deep-Reinforcement-Learning-Based Collision Avoidance in UAV Environment. *IEEE Internet Things J.* **2022**, *9*, 4015–4030. [\[CrossRef\]](#)
- Xu, J.; Guo, Q.; Xiao, L.; Li, Z.; Zhang, G. Autonomous Decision-Making Method for Combat Mission of UAV based on Deep Reinforcement Learning. In Proceedings of the 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China, 20–22 December 2019; p. 538.

15. Zhao, J.; Luo, X. Three-dimensional Path Planning for Unmanned Aerial Vehicle (UAV) Based on Improved Mayfly Algorithm. In Proceedings of the 2022 IEEE International Conference on Unmanned Systems (ICUS), Guangzhou, China, 28–30 October 2022; p. 32.
16. Çintaş, E.; Özyer, B.; Şimşek, E. Vision-Based Moving UAV Tracking by Another UAV on Low-Cost Hardware and a New Ground Control Station. *IEEE Access* **2020**, *8*, 194601. [\[CrossRef\]](#)
17. Wei, W.; Tan, L.; Jin, G.; Lu, L.; Sun, C. A Survey of UAV Visual Navigation Based on Monocular SLAM. In Proceedings of the 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 14–16 December 2018; p. 1849.
18. Ji, T.; Guo, Y.; Wang, Q.; Wang, X.; Li, P. ECONOMY: Point Clouds-Based Energy-Efficient Autonomous Navigation for UAVs. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2885.
19. Zhang, S.; Li, Y.; Ye, F.; Geng, X.; Zhou, Z.; Shi, T. A Hybrid Human-in-the-Loop Deep Reinforcement Learning Method for UAV Motion Planning for Long Trajectories with Unpredictable Obstacles. *Drones* **2023**, *7*, 311. [\[CrossRef\]](#)
20. Yan, C.; Xiang, X.; Wang, C. Fixed-Wing UAVs flocking in continuous spaces: A deep reinforcement learning approach. *Robot. Auton. Syst.* **2020**, *131*, 103594. [\[CrossRef\]](#)
21. Ma, N.; Cao, Y.; Wang, X.; Wang, Z.; Sun, H. A Fast path re-planning method for UAV based on improved A* algorithm. In Proceedings of the 2020 3rd International Conference on Unmanned Systems (ICUS), Harbin, China, 27–28 November 2020; pp. 462–467.
22. Chen, J.; Du, C.; Zhang, Y.; Han, P.; Wei, W. A Clustering-Based Coverage Path Planning Method for Autonomous Heterogeneous UAVs. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25546–25556. [\[CrossRef\]](#)
23. Zhang, H.; Gan, X.; Li, S.; Chen, Z. UAV safe route planning based on PSO-BAS algorithm. *J. Syst. Eng. Electron.* **2022**, *33*, 1151–1160. [\[CrossRef\]](#)
24. Huang, H.; Qin, H.; Hao, Z.; Lim, A. Example-based Learning Particle Swarm Optimization for Continuous Optimization. *Inf. Sci.* **2012**, *182*, 125–138. [\[CrossRef\]](#)
25. Petalas, Y.; Parsopoulos, K.; Vrahatis, M. Memetic particle swarm optimization. *Ann. Oper. Res.* **2007**, *156*, 99–127. [\[CrossRef\]](#)
26. Góez, G.D.; Velásquez, R.A.; Botero, J.S. UAV route planning optimization using PSO implemented on microcontrollers. *IEEE Lat. Am. Trans.* **2016**, *14*, 1705–1710. [\[CrossRef\]](#)
27. Li, Y.-L.; Shao, W.; You, L.; Wang, B.-Z. An Improved PSO Algorithm and Its Application to UWB Antenna Design. *IEEE Antennas Wirel. Propag. Lett.* **2013**, *12*, 1236–1239. [\[CrossRef\]](#)
28. Yu, Z.; Si, Z.; Li, X.; Wang, D.; Song, H. A Novel Hybrid Particle Swarm Optimization Algorithm for Path Planning of UAVs. *IEEE Internet Things J.* **2022**, *9*, 22547–22558. [\[CrossRef\]](#)
29. Pahlevi, R.; Murti, M.A.; Susanto, E. The implementation of PID using particle swarm optimization algorithm on Networked Control System. In Proceedings of the 2014 International Conference on Industrial Automation, Information and Communications Technology, Bali, Indonesia, 28–30 August 2014; pp. 35–38.
30. Xue, L.; Wu, S.; Xu, Y.; Ma, D. A Simulation-Based Multi-Objective Optimization Design Method for Pump-Driven Electro-Hydrostatic Actuators. *Processes* **2019**, *7*, 274. [\[CrossRef\]](#)
31. Borujeni, M.M.; Rashidi, A.; Saghaeian Nejad, S.M. Optimal four quadrant speed control of switched reluctance motor with torque ripple reduction based on EM-MOPSO. In Proceedings of the 6th Power Electronics, Drive Systems & Technologies Conference (PEDSTC2015), Tehran, Iran, 3–4 February 2015; pp. 310–315.
32. Chen, S.; Hong, X.; Gao, J.; Harris, C.J. Complex-Valued B-Spline Neural Networks for Modeling and Inverting Hammerstein Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 1673–1685. [\[CrossRef\]](#)
33. Chen, J.; Zhang, Y.; Wu, L.; You, T.; Ning, X. An Adaptive Clustering-Based Algorithm for Automatic Path Planning of Heterogeneous UAVs. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 16842–16853. [\[CrossRef\]](#)
34. Chen, J.; Ling, F.; Zhang, Y.; You, T.; Liu, Y.; Du, X. Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system. *Swarm Evol. Comput.* **2022**, *69*, 101005. [\[CrossRef\]](#)
35. Boon, W.L.; Chai, P.Y.; Zakir, M.K.; Fawzi, F.B. A Distributed Method for UAV Swarm Path Planning. In Proceedings of the 2022 International Conference on Computer and Drone Applications (IConDA), Kuching, Malaysia, 28–29 November 2022; pp. 12–17.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.