applied sciences

MDPI

Article

Research on Dynamic Target Search for Multi-UAV Based on Cooperative Coevolution Motion-Encoded Particle Swarm Optimization

Yiyuan Li [1], Weiyi Chen [1], Bing Fu [1,*], Zhonghong Wu [1], Lingjun Hao [2] and Guang Yang [1]

[1] College of Weapon Engineering, Naval University of Engineering, Wuhan 430033, China; lyy_simons@163.com (Y.L.); wychennue@sina.com (W.C.); liguilanyangkai@163.com (G.Y.)
[2] Naval Research Academy, Beijing 100161, China; haolingjun1993@163.com
* Correspondence: d22182605@nue.edu.cn

Abstract: Effectively strategizing the trajectories of multiple Unmanned Aerial Vehicles (UAVs) within a dynamic environment to optimize the search for and tracking of mobile targets presents a formidable challenge. In this study, a cooperative coevolution motion-encoded particle swarm optimization algorithm called the CC-MPSO search algorithm is designed to tackle the moving target search issue effectively. Firstly, a Markov process-based target motion model considering the uncertainty of target motion is investigated. Secondly, Bayesian theory is used to formulate the moving target search as an optimization problem where the objective function is defined as maximizing the cumulative probability of detection of the target in finite time. Finally, the problem is solved based on the CC-MPSO algorithm to obtain the optimal search path nodes. The motion encoding mechanism converts the search path nodes into a set of motion paths, which enables more flexible handling of UAV trajectories and improves the efficiency of dynamic path planning. Meanwhile, the cooperative coevolution optimization framework enables collaboration between different UAVs to improve global search performance through multiple swarm information sharing, which helps avoid falling into local optimal solutions. The simulation results show that the CC-MPSO algorithm demonstrates efficacy, reliability, and superior overall performance when compared to the five commonly used swarm intelligence algorithms.

Keywords: cooperative coevolutionary approach; particle swarm optimization; multi-UAV planning; moving target search

check for updates

Citation: Li, Y.; Chen, W.; Fu, B.; Wu, Z.; Hao, L.; Yang, G. Research on Dynamic Target Search for Multi-UAV Based on Cooperative Coevolution Motion-Encoded Particle Swarm Optimization. *Appl. Sci.* **2024**, *14*, 1326. https://doi.org/10.3390/app14041326

Academic Editors: Rodolfo Dufo-López and Vincent A. Cicirello

Received: 5 December 2023
Revised: 3 January 2024
Accepted: 31 January 2024
Published: 6 February 2024

1. Introduction

Due to their excellent maneuverability and adaptability, along with their ability to carry out diverse tasks in challenging environments, Unmanned Aerial Vehicles (UAVs) are increasingly being utilized in the fields of navigation and localization, target search and rescue, border patrol, agricultural production, and cargo transportation [1–4]. Conducting coordinated UAV searches in crucial areas is a typical kind of UAV task aimed at enhancing target detection and gathering intelligent information. For such tasks, the traditional approach is based on search theory, and the corresponding search strategy is formulated in advance, so that the target detection probability is maximized [5]. However, for dynamic target search, usually, the specific location information of the target is unknown or only part of the a priori information can be grasped. As the search time passes, the initial position information and environment information of the target will be attenuated to a large extent, and the probability of finding the target will decrease rapidly. Consequently, the search for moving targets utilizing UAVs relies on the starting position information of the target and the UAV's sensor detection capacity. Path planning is then conducted for the UAVs to achieve the highest possible target detection probability within a certain timeframe.

The key to solving the dynamic target search problem lies in the development of a path planning algorithm that exhibits efficient search performance.

Classical approaches to addressing UAV target search problems such as the Dijkstra method [6], A* method [7], and dynamic programming method [8] ensure the superiority of the final solution but require significant time and resource consumption during processing. In the work of [9], a comprehensive overview of UAV path planning algorithms is provided. The study revealed that the path planning method based on computational intelligence has the capability to achieve higher efficiency, greater flexibility, and enhanced adaptability in complex environments. Swarm-based intelligent algorithms are a subfield of computational intelligence methods, which have excellent performance in modeling collective intelligence and collaborative behavior in nature, making them suitable for use in a wide range of robotics applications. In recent years, swarm-based intelligent target search algorithms have received increasing attention and research (swarm-based intelligent target search algorithm) [10–12]. A dynamic target search method for UAV based on motion-encoded particle swarm optimization algorithm (MPSO) was first proposed in the literature [13]. The method encodes each search path of a UAV as a set of motion paths, which makes the particles search in motion space instead of Cartesian space. The method is simple, intuitive, and very conducive to engineering implementation. It lays theoretical guidance for the development of subsequent UAV intelligent search path planning techniques. However, the method may lack sufficient adaptability and perform sub-optimally when dealing with complex dynamic search problems. On this basis, reference [14] combines the charged particle optimization algorithm (ECPO) with the motion encoding (ME) mechanism and proposes the ECPO-ME algorithm for solving the UAV moving target search problem. Compared with the MPSO algorithm, the ECPO-ME algorithm has strong global search performance, but the increase in population diversity causes its convergence speed to become slower. Reference [15] proposed an ant colony path search optimization algorithm (ASPV) with visibility for the optimal search path problem (OSPV) for Markovian motion objectives, and the improved ASPV variant has good convergence performance. However, the algorithm usually involves some additional hyperparameters, such as Initial Pheromone Level, Pheromone Update Strength, etc. The selection of these parameters has a great impact on the performance of the algorithm, and it is difficult to find a generalized set of optimal parameters. Reference [16] proposed an improved hybrid ant colony optimization algorithm (JPIACO) based on the jump point search strategy, where the initial pheromone concentration distribution obtained by jump point search can guide the algorithm to find the search path quickly and reduce the inflection points of the path, which effectively improves the parameter sensitivity of the algorithm. However, the performance of this algorithm has not been verified in dynamic environments. All of the above algorithms have been applied to single-UAV target search tasks. Single-UAV systems are usually easy to control, easy to deploy, and suitable for small-scale tasks, but for large-scale and highly dynamic search areas, the search efficiency of single UAVs may be low.

Compared with single-UAV target search, multi-UAV collaborative target search has higher search efficiency and wider search scope, and it has gradually become a research hotspot. Ref. [17] models multi-UAV moving target search as a parallel optimization problem and proposes the MEGA-MPC algorithm, which combines the motion encoding mechanism (ME) and the multi-parent genetic algorithm (GA-MPC), for the solution. Unfortunately, the method does not consider mutual communication between UAVs, and the search efficiency is low when multiple areas with high probabilities (MPCs) exist at the same time. Reference [18] proposes a distributed robotic PSO (DRPSO) method to improve the exploration capability of clustered robots in dynamic multi-target search in unknown environments. The method can search out both static and dynamic targets simultaneously while considering realistic constraints such as unknown target initialization, different robot position initialization, limited communication, different obstacle layouts, and decentralized cooperation. Ref. [19] proposes a rule-inspired collaborative search algorithm based on multiple ant colony UAVs which utilizes association rules to drive target probability graph updates. The model is characterized by rule linking, which effectively improves the accuracy of target detection probability in unknown environments. Reference [20] proposes a dynamic discrete pigeon heuristic optimization algorithm for the UAV coordinated search and attack mission planning problem, which combines centralized task assignment and distributed path generation. The algorithm is more easily adaptable to complex, dynamic, and uncertain environments due to its dynamic adaptability and heuristic search properties. However, the algorithm needs to maintain the balance of task allocation and path generation simultaneously, which increases the complexity of the algorithm, and the computational expense is large when dealing with large-scale problems.

To summarize, while swarm intelligence algorithms offer numerous benefits in addressing dynamic target search issues, they are susceptible to getting trapped in local optimal solutions as the number of target regions grows and the impact of low-probability regions extends. Furthermore, some swarm intelligence algorithms may exhibit limited adaptability to changes in the environment. To effectively pursue a target that frequently changes its location or deviates significantly, the algorithms must enhance their adaptability. Intelligent optimization algorithms based on the Cooperative Coevolution framework usually decompose a complex problem into multiple sub-problems, each of which is evolved by a sub-swarm, while the sub-swarms share and compete with each other through the co-evolutionary mechanism for information sharing [21]. Such algorithms have excellent performance in solving large-scale, high-dimensional optimization problems and have been tested in real-world problems such as control parameter optimization, function approximation, and classification [22,23]. Learning from the research of co-evolutionary algorithms applied in other fields, this paper proposes a particle swarm optimization algorithm incorporating multiple swarm co-evolutionary and motion coding mechanisms and applies it to a dynamic target search problem. Specifically, the multi-UAV cooperative search path planning problem is decomposed into numerous single-UAV path planning subproblems, each of which is evolved by an independent particle swarm. By designing a cooperative evolution strategy among multiple swarms, it enables them to co-evolve to find the global optimal solution. In addition, the introduction of motion coding mechanism effectively changes and extends the search space, enhances the population diversity of the algorithm, and improves the probability of searching the target.

The main contributions of this paper are presented as follows: (1) We have proposed a novel particle swarm optimization algorithm incorporating multiple swarm co-evolutionary and motion encoding mechanisms, which effectively changes and expands the search space, strengthens the population diversity, and raises the probability of searching for the dynamic target. (2) Based on the problem decomposition strategy, a cooperative coevolution PSO framework is proposed for dealing with complex optimization problems, which improves the convergence speed and accuracy of the algorithm by means of parallel optimization computation. (3) A mathematical derivation of a target motion model that corresponds to a Markov process is given. The conclusion is that the initial prior information of the target will be highly attenuated and the probability distribution of the target will be extremely dispersed over time, thus enhancing the search difficulty. Based on this model, we have designed six dynamic search scenarios that are more complex compared to previous studies to assess the performance of the proposed algorithm.

The rest of the paper Is organized as follows: Section 2 introduces the steps to establish an objective function. The proposed CC-MPSO algorithm is specified in Section 3. Simulation results and a discussion of searching for moving targets by multi-UAVs are offered in Section 4. Finally, conclusions and plans for future research are drawn in Section 5.

2. Problem Formulation

This section analyzes and establishes the target motion model based on the Markov process, as well as the airborne sensor model and the Markov search probability model. Based on the aforementioned models, multi-UAV dynamic target search tasks are modeled as corresponding probability optimization problems.

2.1. Target Motion Model

In this work, we consider a target motion model that conforms to the Markov process [24], moving randomly through a series of grids, with each grid representing a fixed time step and each time step determining the target's position in the grid. The search area is divided into grids and this search problem is simplified to a path planning problem in the discrete domain range. As shown in Figure 1a, a 3 × 3 grid is arranged according to numerical order. As shown in Figure 1b, assuming that the target is located in grid 5 at time t, the target can only move to the neighboring grids at the time $t+1$ or remain stationary. The probability of the target moving from grid 5 to the other grid is shown in Figure 1c.

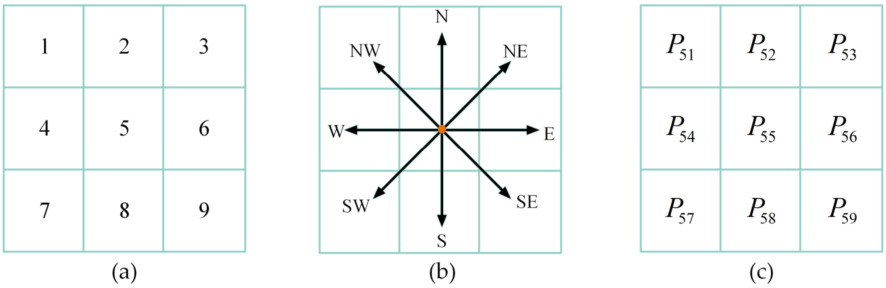


Figure 1. (**a**) Grid segmentation; (**b**) target position transfer direction; (**c**) target position transfer probability.

Before the search begins, the target location is modeled from the available information using a probability distribution function. Supposing that the search space is S, let the unknown variable indicate the position of the target at the time t. The initial probability distribution of the target at time $t=0$ is given by the following equation:

$$\rho(x_0) = P(x(0) = x), x_0 \in S \tag{1}$$

where $P(x(0) = x)$ denotes the probability that the initial position of the target is located at x and satisfies $\sum\limits_{x \in S} \rho(x_0) = 1$.

The position of the target motion is represented according to a certain transfer probability, and the conditional probability distribution of its future state depends only on the current state. Then, the single-step transfer probability at time t is:

$$P_{i,j}(t) = P[x_{t+1} = j | x_t = i] \tag{2}$$

where $P_{i,j}(t)$ is the single-step transfer probability conditional on $x_t = i$ to $x_{t+1} = j$ at time t to time $t+1$. It is characterized by two distinct attributes:

(1) $P_{i,j}(t) \geq 0, i,j \in S$ denotes the nonnegativity of the elements in the single-step transfer probability matrix;

(2) $\sum\limits_{i \in S} P_{i,j}(t) = 1, i \in S$ denotes that the sum of the elements in the single-step transfer probability matrix is 1.

Based on the single-step transfer probability of the moving target, the k-step transfer probability of the target can be further obtained as:

$$P^k_{i,j}(t) = P[x_{t+k} = j | x_t = i] \tag{3}$$

where $P^k_{i,j}(t)$ denotes the probability that the target is transferred from $x = i$ to $x = j$ at time t to time $t+k$. When $k = 1$, it is equivalent to the single-step transfer probability.

The k-step transfer probability of a Markov process satisfies the Kolmogorov–Chapman equation:

$$P^{(k+r)}_{i,j}(t) = \sum_{x \in S} P^{(k)}_{i,x}(t) P^{(r)}_{x,j}(t+k), i,j \in S \tag{4}$$

Based on the Markov motion model, the available information regarding the initial position distribution of the target can be employed to consistently estimate and revise the target's motion position. This allows for the acquisition of a precise posterior distribution for the target. Subsequently, a search algorithm can be employed to determine the optimal search node for the subsequent step.

2.2. Airborne Sensor Model

The airborne sensor model is the basic element of target search, which describes the detection and discovery relationship of the UAV to the search target. In this paper, we consider the UAV airborne infrared sensor, whose field of view is shown in Figure 2. When the UAV flies horizontally at a fixed altitude, the width of the ground detection range of the on-board sensors can be estimated by the following equation, assuming that the influence of roll angle is disregarded:

$$d_s = 2 \cdot h \cdot \frac{\tan \gamma_s}{\tan \alpha_s} \sqrt{1 + (\tan \alpha_s)^2} \tag{5}$$

where $h \leq h_{\max}$ is the altitude of the UAV, α_s is the mounting angle of the sensor, and γ_s is the field of view of the sensor, the value of which is related to the physical properties of the specific sensor.

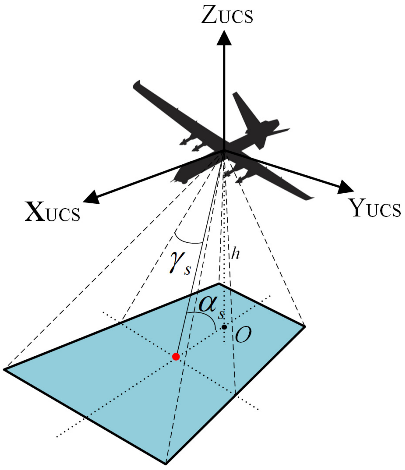


Figure 2. Simulation of the field of view range of the on-board sensors.

To simplify the problem, the detection range d_s of the airborne sensor is considered to cover a grid cell. The UAV searches for a target with the airborne sensors and makes one observation of a grid cell at one time step, and the results of each observation are independent of each other. Assume there are only two possible outputs, namely the detection of the target $O_t = D_t$ and the failure to detect the target $O_t = \overline{D}_t$. Consider that there is a certain detection error in the sensor. Even if the target exists at location x_t, it is not guaranteed that the target position can be detected. Then, the probability of not detecting the target when the target is located in x_t is as follows:

$$P\left(\overline{D}_t | x_t\right) = 1 - P(D_t | x_t) \tag{6}$$

where $P(D_t | x_t)$ indicates the probability that the sensor will detect the target at the location x_t.

2.3. Target Probability Map Model

After rasterizing the search space, the probability of target presence in each raster is represented by a target probability map model. The probability value depends on the initial probability distribution of the target and the detection information of the on-board sensors. The UAV formulates a corresponding search strategy based on the initial target probability map and, at the same time, dynamically updates the target probability map as the search process continues, including prediction updates based on the target motion model and detection updates based on the UAV's sensors. In the prediction update phase, the target probability map is recursive over time based on the target motion model. From Section 2.1, the single-step transfer probability of the target at the time $t-1$ is $P[x_t|x_{t-1}]$; then, at the time t, the predicted updated target probability map is:

$$\widetilde{\rho}(x_t) = \sum_{x_{t-1}\in S} P(x_t|x_{t-1})\rho(x_{t-1}) \tag{7}$$

where $\widetilde{\rho}(x_t)$ represents the target probability map for predicting the position of the target motion at time t, and $P(x_t|x_{t-1})$ denotes the conditional probability that the target moves from the position at time t to the position at time $t-1$.

During the detection update phase, the target probability map is revised by the integration of detection and observation information acquired from the onboard sensors of the UAVs. Suppose that the sequence of observations received by the sensor in t time steps is $O_{1:t}, O_{1:t} = \{O_1, \ldots, O_t\}$; then, the probability that the target is detected at location x_t at time t is $P(O_t|x_t)$. The target probability map after the detection event is updated based on the Bayesian formula as follows:

$$\rho(x_t) = \gamma_t P(O_t|x_t)\widetilde{\rho}(x_t) \tag{8}$$

where γ_t is a normalization factor. It makes the probability of the existence of the target in each grid cell sum to 1. The formula is as follows:

$$\gamma_t = 1/\sum_{x_t\in S} P(O_t|x_t)\widetilde{\rho}(x_t) \tag{9}$$

2.4. Fitness Function for Moving Target Search

The search process of multi-UAVs for moving targets, utilizing airborne sensors, can be conceptualized as a Markov process. The search results at time t are only relevant to the observations at that moment. The formula is as follows:

$$P(D_t|x_t) = P\left(D_t|x_t, \overline{D}_{t-1}, \overline{D}_{t-2}, \ldots, \overline{D}_0\right) \tag{10}$$

Suppose the probability that a single UAV failed to detect a target at time t is $\overline{p}_t$. According to Bayesian theory, its value is obtained by multiplying the probability of not detecting a target by its airborne sensors (Equation (6)) and the probability of target presence in the prediction update phase (Equation (7)). This probability can be expressed as in the whole search area:

$$\overline{p}_t = \sum_{x_t\in S} P\left(\overline{D}_t|x_t\right)\widetilde{\rho}(x_t) \tag{11}$$

Since the probability of detecting a target at each time is mutually independent, the probability Q_t of not detecting a target from time 1 to time t is:

$$Q_t = \prod_{k=1}^{t} \overline{p}_k \tag{12}$$

Therefore, the probability p_t of detecting the target for the first time at time t is:

$$p_t = Q_{t-1}(1-\overline{p}_t) \tag{13}$$

Then, the cumulative probability of detection of the target in t time steps is:

$$P_t = \sum_{k=1}^{t} p_t = 1 - \prod_{k=1}^{t} \overline{p}_t \tag{14}$$

Let M represent the number of UAVs and P_{jt} represent the probability that the j-th UAV finds a target in t time steps. The joint detection probability of multi-UAVs can be calculated as:

$$\max F = 1 - \prod_{j=1}^{M} (1 - P_{jt}) \tag{15}$$

3. Methodology

3.1. Standard PSO

The PSO algorithm belongs to a class of swarm intelligence algorithms inspired by the social behavior of bird flocks. It searches for the optimal solution to a problem by modeling the collaborative search of flock members in the solution space. In PSO, a potential solution in the solution space is called a particle. The position of the particle indicates the location of the solution, and the velocity of the particle indicates the direction and speed of the solution moving in the solution space. At the beginning of the algorithm, the particle population is first initialized with random positions and velocities, and the fitness value of each particle is calculated. For each particle, its optimal position and the global optimal position in the whole population are recorded.

The position and velocity of the i-th particle in the population are denoted as $x_i(t)$ and $v_i(t)$. Then, the updated formula for the particle is given as follows:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (y_i(t) - x_i(t)) + c_2 r_2 (\widetilde{y}_i(t) - x_i(t)) \tag{16}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{17}$$

where ω is the inertial weight, c_1 is the cognitive coefficient, c_2 is the social coefficient, r_1 and r_2 are random sequences sampled from a uniform probability distribution in the range $[0, 1]$. y_i represents the historical optimal position of the particle and $\widetilde{y}_i$ is the global historical optimal position among all particles.

According to Equations (16) and (17), the movement of a particle is directed by three factors, namely, following its way, moving toward its best position, or moving toward the swarm's best position. The ratio among those factors is determined by the values of w, c_1, and c_2.

3.2. Motion-Encoded Mechanism

The particle's encoding in PSO determines how a solution is represented in the search space, and this encoding is typically tailored to the particular optimization issue in question. In the context of the UAV cooperative target search problem, a widely used method of representing the position of a particle is by defining it as a set of nodes that represent the path taken by the UAV during the search. Utilizing this encoding technique allows for a clear and intuitive representation of the desired search path, which is advantageous for evaluating adaptability. However, it may not adequately encompass the surrounding nodes throughout the search, resulting in the creation of erroneous paths. In addition, a method based on angle encoding has also been proposed in [25], which has achieved better results in dealing with path planning problems with specific constraints. However, the phase angle of the particle is constrained within a specific range, $[-\pi/2, \pi/2]$. Additionally, it is worth noting that the particle's search capability is also limited.

To address the aforementioned issues, reference [13] proposed a motion-encoded particle swarm optimization algorithm (MPSO). It is assumed that the UAV can only move into the eight adjacent grid cells in each time step. The motion can be represented as a vector $\gamma_t = (\phi_t, \vartheta_t)$, where ϕ_t denotes the motion amplitude and ϑ_t denotes the motion direction at time t. Assuming that the UAV searches for a dynamic target in t time steps, the particle positions can be characterized accordingly as:

$$S = \{\gamma_1, \gamma_2, \ldots, \gamma_t\}, \quad t = 1, 2, \ldots, T \tag{18}$$

The search paths of multi-UAVs are coded simultaneously using motion encoding, as depicted in Figure 3. A search path, denoted as S_i, comprises T motion segments representing the position of each population individual. Therefore, if we have information on the initial position of the UAV and the direction of movement for each successive step, we can accurately represent the UAV's search path, as depicted in Figure 4. The color of the grid is indicative of the probability of the target materializing at that particular position. The red dot represents the location of the UAV, while the white arrow in the top left corner indicates the direction of movement of both the UAV and the target towards the adjacent eight grid cells.

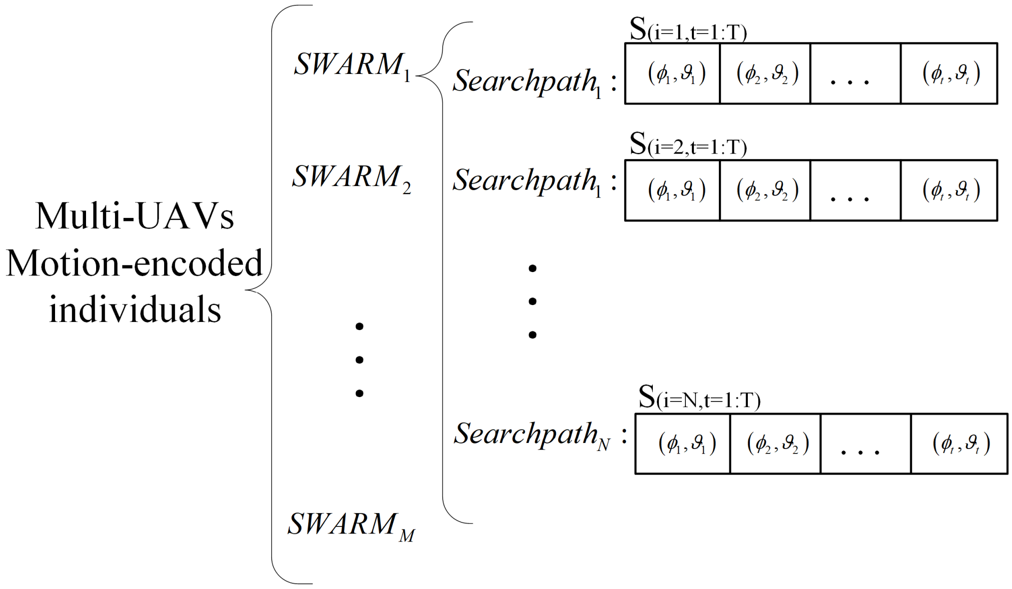

Multi-UAVs Motion-encoded individuals
$SWARM_1$
$SWARM_2$
$SWARM_M$
$Searchpath_1$:
$S(i=1,t=1:T)$
(ϕ_1,ϑ_1) (ϕ_2,ϑ_2) . . . (ϕ_t,ϑ_t)
$Searchpath_1$:
$S(i=2,t=1:T)$
(ϕ_1,ϑ_1) (ϕ_2,ϑ_2) . . . (ϕ_t,ϑ_t)
$Searchpath_N$:
$S(i=N,t=1:T)$
(ϕ_1,ϑ_1) (ϕ_2,ϑ_2) . . . (ϕ_t,ϑ_t)

Figure 3. Motion-encoded population individuals.

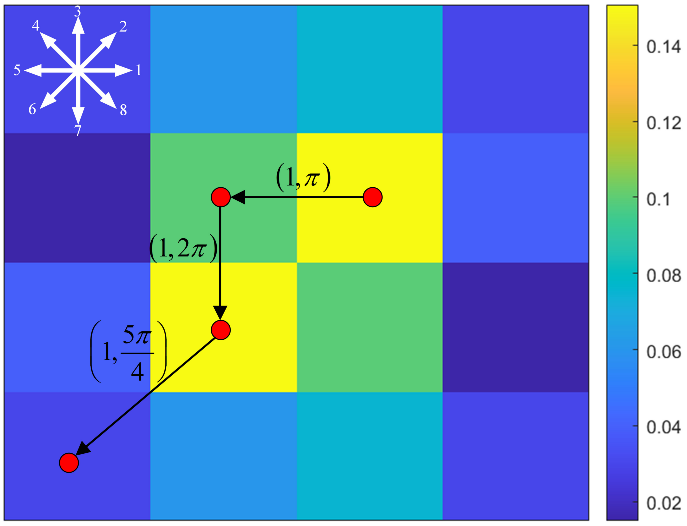


Figure 4. Motion-encoded based search path representation.

In the process of updating the particles, it is essential to assess particle adaptation by mapping the motion vectors of each step to the search route nodes of the UAV. The method is as follows.

Firstly, the motion amplitude and motion direction are normalized:

$$\phi_t{}' = 1 \tag{19}$$

$$\vartheta_t' = \frac{\pi}{4}\left\langle \vartheta_t / \frac{\pi}{4} \right\rangle \tag{20}$$

where the motion amplitude is normalized to 1, $\langle \cdot \rangle$ represents the rounding operator, and the motion direction is quantified as the direction from the current position to the 8 neighboring grids.

Then, the position vector of the current moment is added with the normalized motion vector to get the position vector of the next moment:

$$\chi_{\mathrm{i},t+1} = \chi_{i,t} + \gamma_{i,t}' \tag{21}$$

$$\gamma_{i,t}' = \phi_t{}'\left(\langle \cos \vartheta_t' \rangle, \langle \sin \vartheta_t' \rangle\right) \tag{22}$$

In general, MPSO improves the movement and variety of the particle swarm in the search area by using a more adaptable encoding method, which ultimately enhances the algorithm's ability to do global searches. Furthermore, this method ensures that the spacing between adjacent nodes in the trajectory aligns with the UAV's maneuverability, thereby ensuring the reachability of the nodes in the path. Consequently, the search pathways that have been described stay relevant even after each generation of swarm updates.

3.3. Proposed CC-MPSO Algorithm

Based on Bayesian theory, this paper constructs the multi-UAV dynamic target search problem as an optimization problem that maximizes the target detection probability in finite time. With the increase in the number of UAVs and targets, the complexity of the problem grows exponentially, and the performance of traditional algorithms can no longer meet the requirements. A cooperative coevolution framework (CC framework) is an important tool for solving optimization problems. The idea is to divide the original optimization problem into multiple subproblems, each corresponding to a sub-population. Through the coevolution between sub-problems, the global optimal solution can be searched for more efficiently. Potter. et al. applied this framework to genetic algorithms [26], where they divided the solution vector into multiple chromosomes, and each chromosome belonged to a different population. F. van den Bergh et al. applied a similar framework to PSO algorithms [27], where the space of n-dimensional solution vectors is partitioned into one-dimension solution space, and each solution space corresponds to a sub-population that is independent in the update.

Considering that the objective function in this paper is determined by the joint target detection probability of multiple UAVs, the CC-MPSO fusion algorithm is proposed based on the CC framework and the MPSO algorithm by drawing on the idea of cooperative evolution. Unlike traditional optimization algorithms, our approach views the search task of multi-UAVs as a co-evolutionary process, an idea that makes the algorithm more adaptive, flexible, and better adapted to complex environments. Another innovation is that we introduce a motion-encoded mechanism. Coding based on UAV motion paths not only optimizes their detection performance but also enables more effective information sharing and collaborative decision-making in collaborative tasks, thus improving overall performance. A schematic overview of numerous swarms cooperating to optimize the search paths of multi-UAVs is shown in Figure 5, and the basic steps of CC-MPSO are shown as follows:

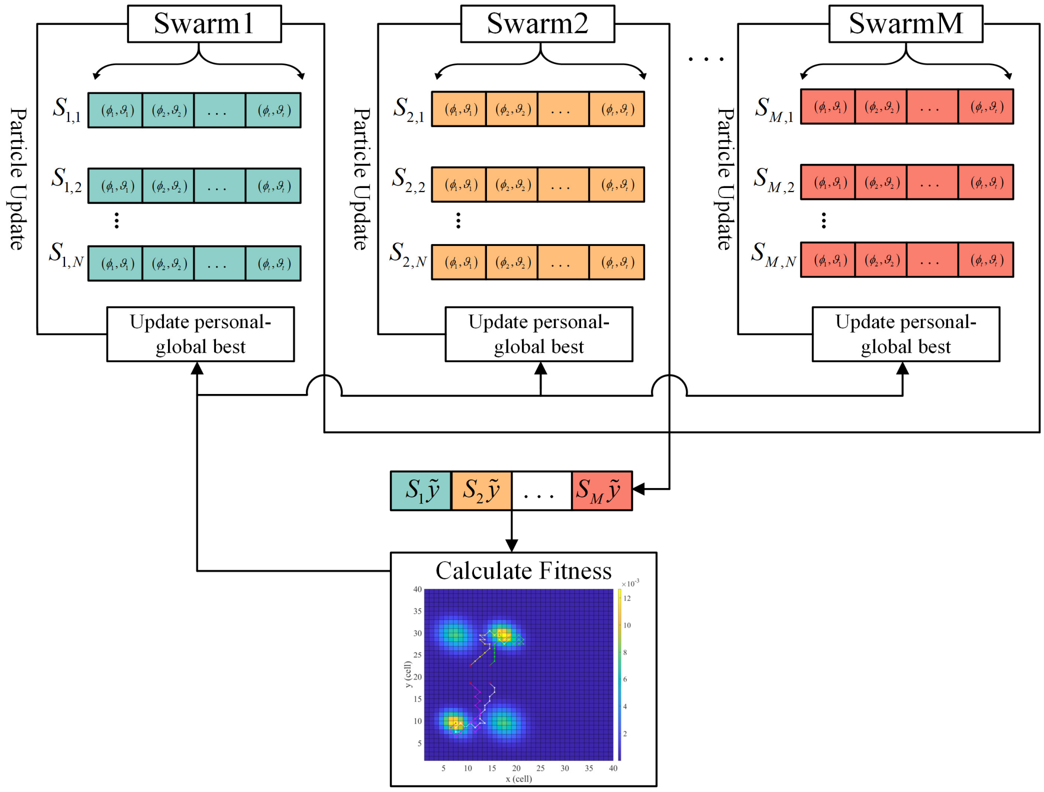


Figure 5. A schematic overview of numerous swarms cooperating to optimize the search paths of multi-UAVs.

Step 1: Problem modeling. The challenge of dynamically searching for targets using multi-UAVs is divided into several sub-problems, with each sub-problem corresponding to a specific UAV. Each sub-problem defines the state representation, search range, and objective function for its own UAV.

Step 2: Population initialization. The particle of each population is initialized for each subproblem and motion-encoded. The position vector of each particle reflects the motion path of each UAV.

Step 3: Subpopulation update. The standard PSO algorithm is used to update the position and velocity of the particles in each subpopulation with the following equations:

$$S_k v_i(t+1) = wS_k v_i(t) + c_1 r_1 [S_k y_i(t) - S_k x_i(t)] + c_2 r_2 [S_k \widetilde{y}_i(t) - S_k x_i(t)] \tag{23}$$

$$S_k x_i(t+1) = S_k x_i(t) + S_k v_i(t+1) \tag{24}$$

where $S_k x_i(t)$ and $S_k v_i(t)$ denote the position and velocity for particle i of swarm k, respectively.

The personal best position $S_k y_i(t)$ for each particle i of each swarm k depicts the best result found for this particle up to iteration t:

$$S_k y_i(t+1) = \begin{cases} S_k y_i(t), \quad if & F\left(S_{\{1:M\}-\{k\}}\widetilde{y}(t), S_k y_i(t)\right) \geq F\left(S_{\{1:M\}-\{k\}}\widetilde{y}(t), S_k x_i(t+1)\right) \\ S_k x_i(t+1), if & F\left(S_{\{1:M\}-\{k\}}\widetilde{y}(t), S_k y_i(t)\right) \leq F\left(S_{\{1:M\}-\{k\}}\widetilde{y}(t), S_k x_i(t+1)\right) \end{cases} \tag{25}$$

The global best position of each particle i for each swarm is updated using:

$$S_k\widetilde{y}(t) = \underset{S_k y_i(t)}{\mathrm{argmax}} F\left(S_{\{1:M\}-\{k\}}\widetilde{y}(t), S_k y_i(t)\right) \tag{26}$$

In addition, the velocity of each particle $S_k x_i$ is controlled by a clamping $S_k v_{\max}$ that regulates the maximum velocity update to a defined range of $[-S_k v_{\max}, S_k v_{\max}]$. The clamping constant controls the exploration–exploitation trade-off by modifying the particle's ability to explore a narrow or large search space.

$$S_k v_i(t+1) \begin{cases} S_k v_i(t+1), if \|S_k v_i(t+1)\| \leq S_k v_{\max} \\ \pm S_k v_{\max}, otherwise \end{cases} \tag{27}$$

Step 4: Particle fitness evaluation. In CC-MPSO, each sub-population corresponds to a specific part of the solution vector in the optimization process. Therefore, to evaluate the value of individual fitness in each population, the particle needs to be solved in association with the globally optimal particles of the remaining populations. The formula is shown as follows:

$$Fitness(S_k x_i) = F\left(S_{\{1:M\}-\{k\}}\widetilde{y}, S_k x_i\right) \tag{28}$$

where $S_{\{1:M\}-\{k\}}\widetilde{y}$ denotes the global best position of each particle in all populations except the swarm k.

Step 5: Update coevolution information. Based on the fitness of each subpopulation, the information on coevolution is updated to guide the algorithm to search towards the global optimal solution. To avoid the algorithm trapping in suboptimal solutions, the swarm particles can be reset into new random positions when one of the swarms converges in a small region of the search space while the global best position of the complementary swarm changes. The determination of whether the swarm is locally convergent or not is based on the computation of the sum of the normalized distances of all particles in the present swarm from the global optimal position, using the following formula:

$$S_k d_{norm}(t) = \frac{S_k d(t)}{S_k d(0)} \tag{29}$$

$$S_k d(t) = \sum_{i=1}^{N} [\|S_k x_i(t) - S_k\widetilde{y}(t)\|] \tag{30}$$

where $S_k d(t)$ denotes the sum of the Manhattan distances between the positions of all particles in the swarm k and the global optimal position. The swarm k is considered to be trapped in stagnation when $S_k d_{norm}(t)$ is smaller than a convergence constant ε.

Step 6: Repeat the above steps until the maximum number of iterations is reached. Based on the output of the algorithm, the optimal solution to the problem is obtained, including the optimal path for each UAV and the final target location probability distribution.

The algorithm presented in this paper offers two benefits for the challenge of dynamically searching for targets using several UAVs. One benefit is that once the update is finished, each set of individual UAV path parameters will be updated with the path parameters of other UAVs. This more detailed updating mechanism circumvents the issue of making progress and then regressing, which is frequently experienced with conventional PSO. An additional benefit is that the algorithm presented in this research effectively integrates many individuals from distinct communities to augment the diversity of global solutions. Pseudo-code for the proposed CC-MPSO algorithm for dynamic target search using multi-UAVs is given in Algorithm 1.

Algorithm 1: The pseudo-code of CC-MPSO.

```
Input: $\omega, c_1, c_2, r_1, r_2$: PSO operational parameters
    $\varepsilon$: swarm convergence parameter
      $M$: number of swarms
      $N$: number of particles in each swarm
      $MaxIt$: The maximum number of iterations per swarm run
Output: Optimal search paths of Multi-UAVs
Initialize $M$ motion-encoded swarms randomly
For $k = 1 : M$
    For $i = 1 : N$
        Compute the fitness of particle $S_k x_i$ according to $F(S_1 x_i, \ldots S_M x_i)$
        Initialize individual optimal position $S_k y(0)$ accordingly
    End for
Initialize the global optimal position $S_k \widetilde{y}(0)$ of each swarm
End for
While $t < MaxIt$
    For $k = 1 : M$
        Select the $swarm_k$
        Compute the normalized sum of distances $S_k d_{norm}(t)$ for $swarm_k$
        If $t > 1$ and $S_{\{1:M\}-\{k\}} \widetilde{y}(t-1) \neq S_{\{1:M\}-\{k\}} \widetilde{y}(t-2)$ and $d_k < \varepsilon$
        Then reset particles of $swarm_k$ to the random position
        For $i = 1 : N$
            Compute fitness of $S_k x_i(t)$
            Update personal optimal position $S_k y_i(t)$ according to Equation (21)
            Calculate velocity $S_k v_i(t+1)$
            Calculate position $S_k x_i(t+1)$
        End for
    Update global optimal position $S_k \widetilde{y}(t)$ according to Equation (22)
    End for
End while
```

3.4. Computational Complexity Analysis

Worst-case time complexity is a vital measure for assessing the efficiency of an algorithm, indicating its ability to effectively solve a task. In order to represent the worst-case temporal complexity, the notation "big-O" is used. The complexity mostly depends on criteria such as the size of the population, the dimension of each individual, the number of iterations, and the number of fitness function evaluations needed. The time complexity for CC-MPSO is given as follows.

In CC-MPSO, it is assumed that N is the swarm scale, and M is the number of swarms. Dim refers to the dimension of the detection space, and the time complexity of the initial stage of the swarm individuals is $O(N \times M \times Dim)$. Therefore, when the total number of iterations of the optimizer is $MaxIt$, the complexity of the CC-MPSO algorithm is $O(N \times M \times Dim \times MaxIt)$.

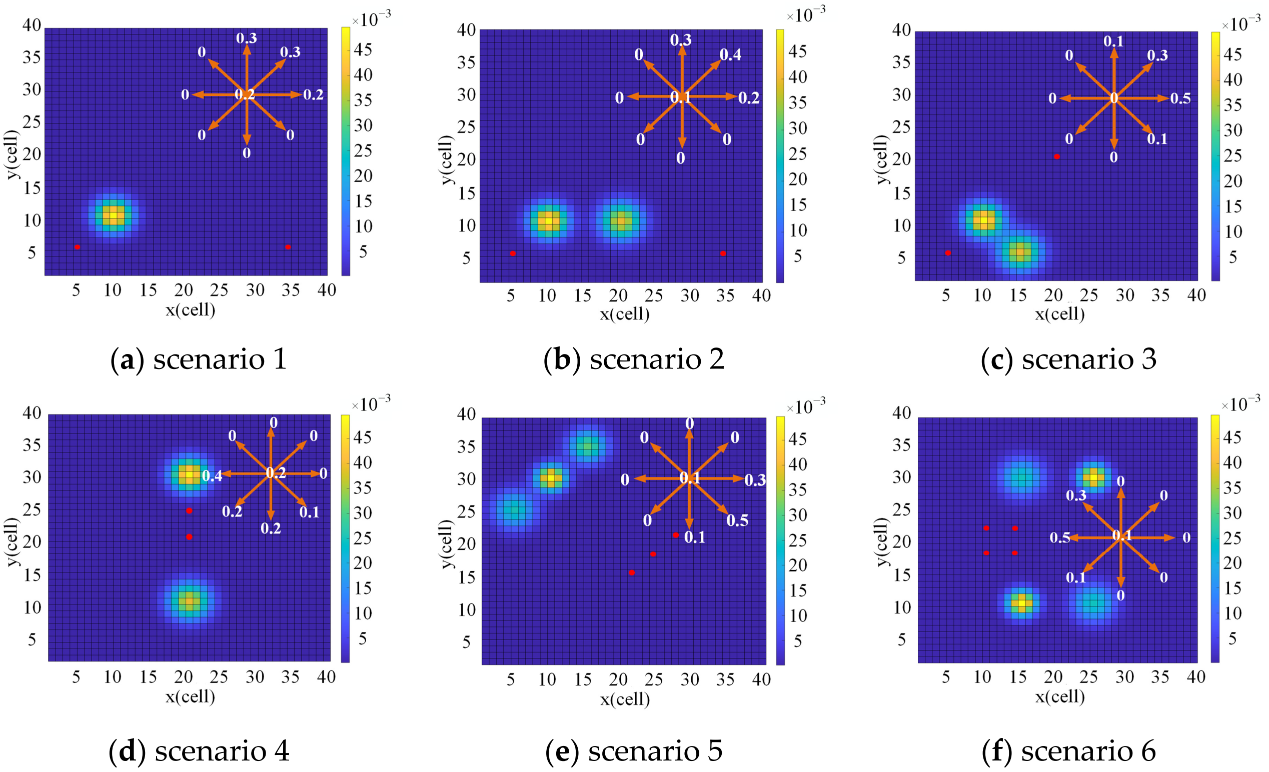
4. Simulation Experiment and Discussion

The algorithm proposed in this paper was fully programmed in the MATLAB2021b environment. The main configuration of the hardware environment is a win10 laptop, Intel Core i9-11900H, 2.50 GHz CPU, and 16 GB RAM.

4.1. Scenarios

To thoroughly assess the performance of the algorithm, six variable search scenarios were employed to evaluate the optimal search performance of CC-MPSO. Scenarios 1 to 4 are drawn from the works of Phung and Ha [13] and Alanezi et al. [17]. Scenarios 5 and 6 are developed based on scenarios 1 to 4. In addition, scenarios 5 and 6 encompass more intricate initial probability distributions of targets, coupled with an increased count of UAVs. According to Markov motion model, this will influence the probability values of the other regions, thus improving the search difficulty of the algorithm.

Figure 6 depicts the tested scenarios whereby the target probability map is color-coded such that warmer color cells represent higher target probabilities. A red dot indicates the initial position of UAVs. A white arrow depicts the dynamics of the moving targets; the values on the arrows correspond to the probability of the target moving in different directions.



(**a**) scenario 1 (**b**) scenario 2 (**c**) scenario 3

(**d**) scenario 4 (**e**) scenario 5 (**f**) scenario 6

Figure 6. Settings for search scenarios.

The scenarios represent different searching situations and are described as follows:

Scenario 1 has one small dense region moving rapidly. The motion state transfer probability of this target indicates that its approximate direction of motion is northeast. Two UAVs are searching at the same time in this scenario; it thus tests the algorithm on its exploration and adaptability capability.

Scenario 2 has two separated high-probability regions with two UAVs on either side of them; their approximate direction of motion is both northeast. The scenario necessitates the algorithm's autonomous selection of a more optimal region for target detection.

Scenario 3 consists of two adjacent regions with a high probability of occurrence. Their position and values differ slightly, which can make it challenging to identify a more suitable place to search for the target.

Scenario 4 includes two separated high-probability regions located opposite to each other over the two UAV locations. The algorithm has to quickly identify the better region to search and track as the target is moving southwest.

Scenario 5 has three neighboring probability regions, all of which show the direction of target movement to be southeast according to the motion state transfer probability. Three UAVs are searching simultaneously on the map. This setting evaluates the algorithm's ability to efficiently search for the correct target region along a diagonal trajectory.

Scenario 6 has two sets of targets, each containing two adjacent high-probability areas. Four UAVs are searching simultaneously; the increased complexity of the initial probability distribution of the targets and the elevated number of UAVs make the solution space of the algorithm much higher and put high requirements on the efficiency and robustness of the algorithm.

4.2. Methods Configuration and Parameter Selection

Parameter selection of the CC-MPSO algorithm is mainly derived from the recommendations in the literature [17], in conjunction with trial and error. Compared to the standard PSO algorithm, CC-MPSO has multiple swarms, and the number of swarms depends on the number of UAVs used for searching. The swarm scale remains consistent with the standard PSO, MPSO algorithm. The inertial weight ω, cognitive coefficient, and social coefficient c_2 of different swarms use the same values. For the velocity clamping constant $S_k v_{\max}$, a problem-dependent approach is used to select the velocity clamping constant of each swarm, whose magnitude is associated with the input space dimension [23]. The size of the swarm convergence constant has a significant impact on the performance of the algorithm. To optimize this parameter, we gradually increment its value in the range of [0, 1] by trial-and-error experiments to determine the value at which optimal performance can be achieved.

For comparison purposes, five widely used swarm intelligence algorithms were also tested in six scenarios. These include the Particle Swarm Algorithm (PSO) [24] and Ant Colony Algorithm (ACO) [25], as well as the improved algorithms MPSO [26], ECPO-ME [10] for PSO, and RI-MAC [15] for ACO. The above algorithms were chosen because they are extremely competitive and have been applied in work related to multi-UAV target search. The list of algorithms along with their abbreviation, main reference and operational parameters are given in Table 1. The population sizes for ACO, ECPO-ME, RI-MAC, and CC-MPSO are all set to 100. However, the PSO and MPSO is set to 500 due to poor performance in simulation and difficulties in convergence when the population size is set too low. The other crucial parameters of the algorithm are set from its original references. To mitigate the inherent instability and uncertainty of a single experiment, we conducted 100 experiments for each scenario using Monte Carlo simulation. In each experiment, the algorithms were run for a maximum of 200 iterations. The results of these experiments were then subjected to statistical analysis. The specific discussions are in Section 4.3.

Table 1. Operational parameters of comparing algorithms.

Algorithm	Abbreviation	Main Reference	Parameters
Particle Swarm Optimization	PSO	[28]	Population size = 500, Inertia weight = 0.8 cognitive factor = 2.5, social factor = 2.5.
Ant colony optimization	ACO	[15]	Colony size = 100, Pheromone importance factor = 1, Pheromone Evaporation Rate =0.85.
Motion-encoded Particle Swarm Optimization	MPSO	[13]	Population size = 500, Inertia weight = 1 at the damping rate of 0.98 cognitive factor = 2.5, social factor = 2.5.
Motion-encoded electric charged particles optimization	ECPO-ME	[14]	Number of ECPs = 100, number of ECPs in interaction = 3, size of the archive pool = 100.
Rule-inspired multi ant colony optimization	RI-MAC	[19]	Population size = 100, the number of ant populations corresponding to the UAV swarm, Initialization of pheromone based on target prior probability.
Cooperative Coevolution Motion-encoded Particle Swarm Optimization	CC-MPSO	This paper	Population size = 100, inertia factor = 0.6, cognitive factor = 1.8, social factor = 1.8, velocity clamping factor = 0.2, Convergence constant = 0.2.

4.3. Experimental Results

In this section, we will statistically analyze the experimental results. The "BEST", the "WORST", the "MEAN", the "MEDIAN", and the "Standard Deviation (SD)" for all the compared algorithms and scenarios are tabulated and discussed, as given in Tables 2–7. It should be noted that the global objective function constructed in this paper represents the joint detection probability of multi-UAVs. To carry out a more fine-grained analysis, the detection probability of each UAV in the same scenario is counted separately.

(1) Scenario 1

The results for scenario 1 are tabulated in Table 2. The acquired results indicate the following conclusions:

- For the first UAV, the proposed CC-MPSO obtained the best results in terms of the statistical parameters "BEST", "MEAN", and "SD". Meanwhile, RI-MAC has obtained the best results for the remaining statistical parameters. For the second UAV, CC-MPSO has achieved the best results for all the considered statistical parameters.
- Except for the standard PSO, the other algorithms perform well in scenario 1. This suggests their capability to formulate effective search paths within uncomplicated settings. However, further examination is warranted to evaluate their efficacy in more intricate scenarios.

Table 2. Statistical analysis of the outcome of all algorithms for scenario 1. (The bold data indicates the optimal value for each row and * sign indicates the serial number of UAVs used in the scenario, which is the same for Tables 2–7.)

UAV*	Statistical Analysis	CC-MPSO	PSO	MPSO	ECPO-ME	ACO	RI-MAC
UAV1	BEST	**0.44171**	0.35157	0.39275	0.42179	0.40926	0.43926
	WORST	0.41095	0.29137	0.31151	0.35953	0.31654	**0.41153**
	MEAN	**0.43966**	0.28414	0.34991	0.38946	0.35962	0.42027
	MEDIAN	0.42153	0.27459	0.35107	0.37881	0.34717	**0.42218**
	SD	**0.00591**	0.01454	0.00782	0.06110	0.01296	0.00432
UAV2	BEST	**0.18294**	0.15926	0.17926	0.18153	0.17895	0.18015
	WORST	**0.17151**	0.13575	0.15784	0.17045	0.16796	0.16958
	MEAN	**0.17745**	0.14154	0.16015	0.17456	0.17014	0.17414
	MEDIAN	**0.18001**	0.13975	0.16792	0.17551	0.17442	0.17523
	SD	**0.00414**	0.01879	0.01058	0.00793	0.01031	0.01294

Table 3. Statistical analysis of the outcome of all algorithms for scenario 2.

UAV*	Statistical Analysis	CC-MPSO	PSO	MPSO	ECPO-ME	ACO	RI-MAC
UAV1	BEST	**0.21464**	0.18324	0.19516	0.20153	0.19201	0.21123
	WORST	0.18575	0.15421	0.16198	0.17559	0.17015	**0.19091**
	MEAN	**0.19911**	0.16545	0.17759	0.18652	0.18552	0.19821
	MEDIAN	0.19512	0.16137	0.17181	0.19033	0.18157	**0.19541**
	SD	**0.00458**	0.01517	0.00845	0.01021	0.09258	0.00584
UAV2	BEST	**0.16044**	0.14926	0.15926	0.15975	0.15021	0.15886
	WORST	**0.14351**	0.11926	0.13857	0.13211	0.12151	0.13952
	MEAN	**0.15281**	0.12546	0.14511	0.13998	0.13095	0.14051
	MEDIAN	**0.15011**	0.12617	0.14331	0.14014	0.13588	0.14541
	SD	**0.00686**	0.01253	0.00984	0.00857	0.01102	0.00843

Table 4. Statistical analysis of the outcome of all algorithms for scenario 3.

UAV*	Statistical Analysis	CC-MPSO	PSO	MPSO	ECPO-ME	ACO	RI-MAC
UAV1	BEST	**0.21599**	0.18798	0.19846	0.19846	0.17651	0.20151
	WORST	0.18684	0.14321	0.15138	0.16846	0.15926	**0.18754**
	MEAN	**0.19021**	0.15952	0.16315	0.17021	0.16532	0.19035
	MEDIAN	**0.19115**	0.16123	0.17015	0.16941	0.16746	0.19151
	SD	0.00541	0.02151	0.01218	0.00836	0.01151	**0.00431**
UAV2	BEST	**0.21238**	0.17465	0.18846	0.19941	0.17065	0.20185
	WORST	**0.18654**	0.14135	0.16546	0.17056	0.15489	0.17651
	MEAN	**0.19165**	0.15454	0.17035	0.18541	0.16035	0.18954
	MEDIAN	0.19084	0.15135	0.17156	0.17954	0.16846	**0.19135**
	SD	**0.00451**	0.01546	0.00515	0.01051	0.00984	0.00549

Table 5. Statistical analysis of the outcome of all algorithms for scenario 4.

UAV*	Statistical Analysis	CC-MPSO	PSO	MPSO	ECPO-ME	ACO	RI-MAC
UAV1	BEST	**0.24595**	0.20165	0.21151	0.21254	0.19456	0.23458
	WORST	0.20235	0.15498	0.15687	0.18456	0.14354	**0.21045**
	MEAN	**0.22546**	0.17416	0.17945	0.19453	0.15235	0.22054
	MEDIAN	**0.23015**	0.16941	0.17456	0.19154	0.16020	0.21945
	SD	0.00921	0.02481	0.02358	0.00985	0.02157	**0.00845**
UAV2	BEST	**0.28209**	0.25138	0.26165	0.26786	0.24198	0.27954
	WORST	0.26135	0.19465	0.21546	0.20589	0.18454	**0.26735**
	MEAN	0.27135	0.22054	0.23024	0.23498	0.21548	**0.27218**
	MEDIAN	**0.27984**	0.22157	0.22098	0.23759	0.22085	0.26549
	SD	0.00584	0.02565	0.01546	0.00984	0.01489	**0.00448**

Table 6. Statistical analysis of the outcome of all algorithms for scenario 5.

UAV*	Statistical Analysis	CC-MPSO	PSO	MPSO	ECPO-ME	ACO	RI-MAC
UAV1	BEST	**0.16603**	0.14135	0.15926	0.15055	0.13548	0.16015
	WORST	**0.15118**	0.12125	0.12515	0.12835	0.10546	0.13456
	MEAN	**0.15756**	0.13456	0.13548	0.13456	0.11548	0.14568
	MEDIAN	**0.15486**	0.13875	0.13058	0.13816	0.12132	0.14125
	SD	**0.00486**	0.00587	0.01289	0.00898	0.00789	0.00985
UAV2	BEST	**0.18433**	0.15165	0.14158	0.16489	0.14321	0.17189
	WORST	0.15984	0.12346	0.11356	0.12151	0.11285	**0.16154**
	MEAN	**0.16865**	0.13786	0.13156	0.14321	0.12486	0.13465
	MEDIAN	**0.16423**	0.14032	0.13486	0.14154	0.12154	0.13515
	SD	**0.00358**	0.00586	0.00786	0.00654	0.00145	0.00865
UAV3	BEST	**0.17945**	0.15654	0.16153	0.16479	0.15135	0.17156
	WORST	0.15354	0.13546	0.14321	0.14315	0.13154	**0.15868**
	MEAN	0.16321	0.14535	0.15135	0.15465	0.14535	**0.16324**
	MEDIAN	**0.16153**	0.15154	0.12535	0.12454	0.14684	0.16054
	SD	**0.00518**	0.00486	0.00589	0.01255	0.01086	0.00846

Table 7. Statistical analysis of the outcome of all algorithms for scenario 6.

UAV*	Statistical Analysis	CC-MPSO	PSO	MPSO	ECPO-ME	ACO	RI-MAC
UAV1	BEST	**0.15581**	0.11235	0.12135	0.13153	0.12654	0.14985
	WORST	**0.13516**	0.08486	0.10135	0.11235	0.09846	0.13158
	MEAN	**0.14155**	0.10211	0.11232	0.12021	0.11205	0.14029
	MEDIAN	0.14551	0.11213	0.10756	0.11954	0.10984	**0.14689**
	SD	0.00483	0.01321	0.00865	**0.00315**	0.00846	0.00532
UAV2	BEST	**0.15305**	0.13155	0.12895	0.13158	0.12156	0.15123
	WORST	0.13215	0.11215	0.10184	0.11218	0.09154	**0.13847**
	MEAN	**0.14516**	0.12021	0.11055	0.12157	0.11484	0.14084
	MEDIAN	**0.14984**	0.12566	0.11846	0.12786	0.11846	0.14235
	SD	**0.00485**	0.00858	0.00684	0.01325	0.00987	0.00865
UAV3	BEST	0.14041	0.10513	0.11235	0.12154	0.11235	**0.14335**
	WORST	0.12015	0.08356	0.09021	0.10213	0.08456	**0.12154**
	MEAN	**0.13031**	0.09128	0.10212	0.11235	0.10215	0.12485
	MEDIAN	0.12832	0.09513	0.10514	0.10984	0.10465	**0.12984**
	SD	**0.00516**	0.00684	0.00846	0.01032	0.00846	0.00784
UAV4	BEST	**0.15118**	0.12032	0.13151	0.14651	0.12123	0.14987
	WORST	0.12021	0.09125	0.10213	0.11238	0.09865	**0.12346**
	MEAN	**0.13456**	0.10218	0.11215	0.12354	0.10125	0.12354
	MEDIAN	**0.13516**	0.11098	0.11568	0.11984	0.11523	0.12865
	SD	0.00746	0.00648	0.01028	0.00788	0.00589	**0.00424**

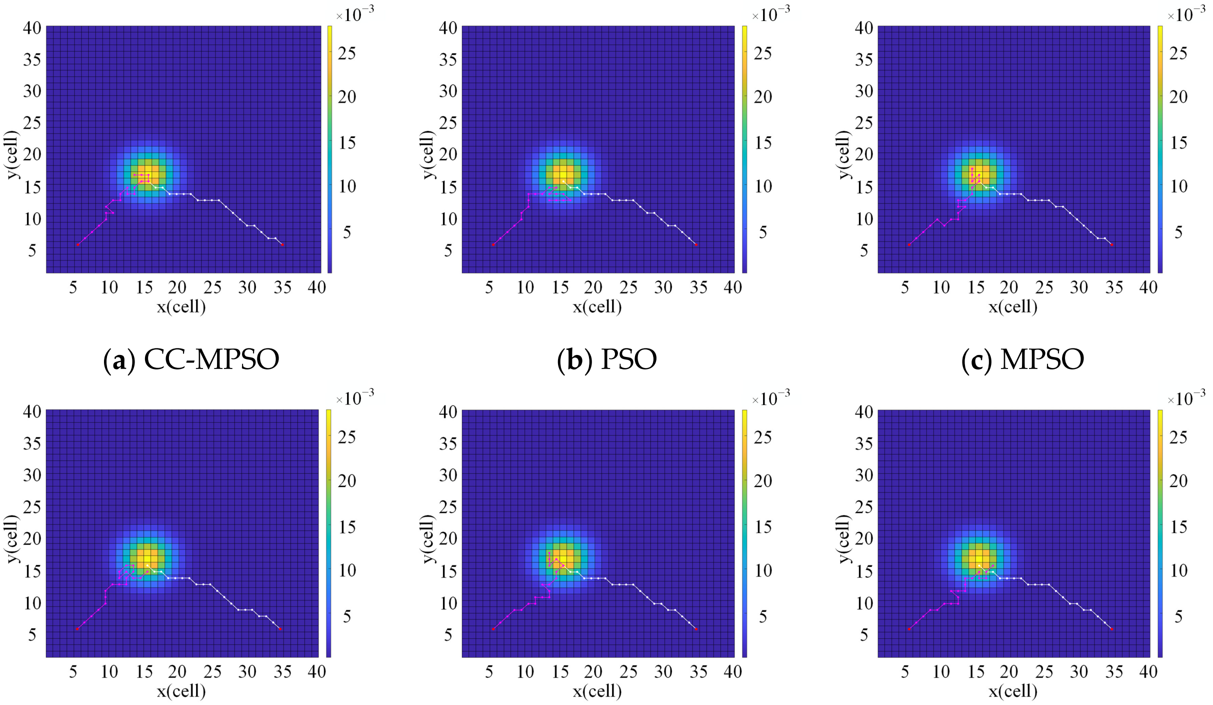
Figure 7 shows the optimal search paths for each UAV for all tested algorithms. In the figure, the purple path indicates the path generated by UAV1, the white path indicates the path generated by UAV2, and the red dot indicates the initial position of UAVs. It is worth mentioning that the presented target probability map depicts the target's final step. By comparing the present map with the one shown in Figure 7, it is feasible to examine the evolution of targets and track their progression. The rest of the scenarios are represented in the same way as this figure. It can be seen that in the path planned by CC-MPSO, UAV1 can search along the direction of the target's motion trajectory, and UAV2 can quickly approach the high-probability target area at an initial position farther away from the target.

(2) Scenario 2

The results for scenario 2 are tabulated in Table 3. The acquired results indicate the following conclusions:

- For the first UAV, the CC-MPSO algorithm achieves the best results among the three statistical parameters "BEST", "MEAN", and "SD", and "RI-MAC" achieves the best results among the two statistical parameters "WORST" and "MEDIAN".
- For the second UAV, the CC-MPSO algorithm achieves optimal results in all statistical terms.

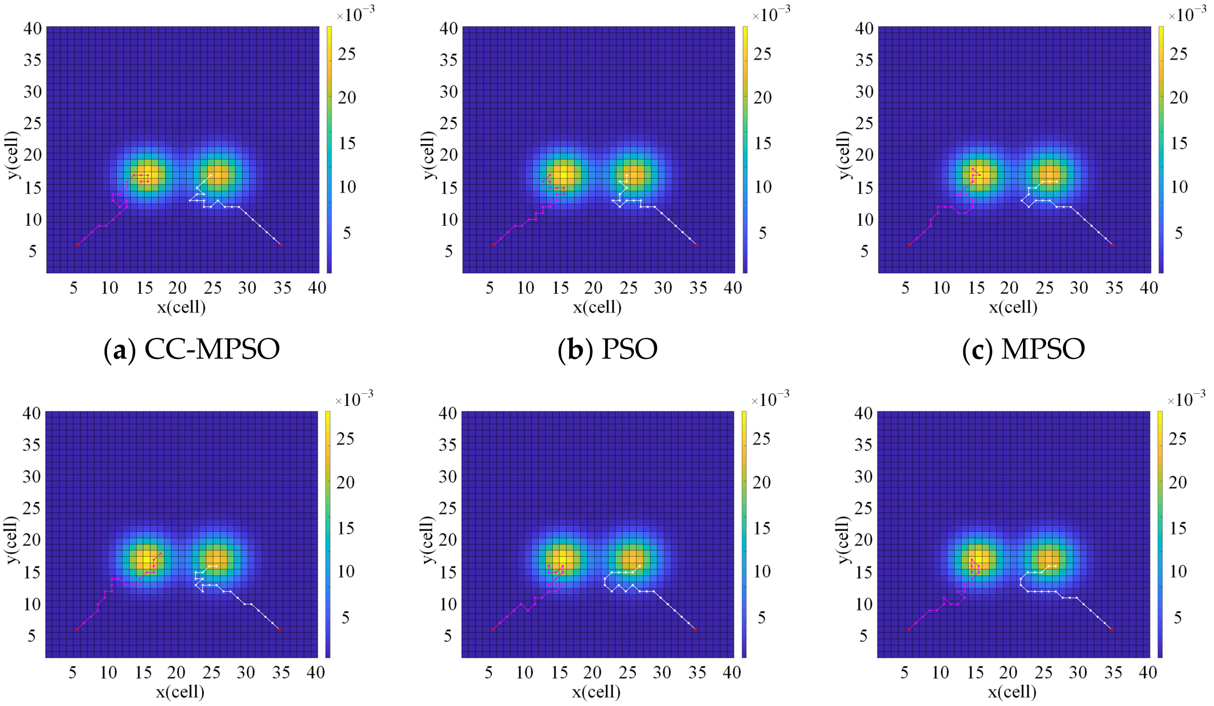
Figure 8 shows the optimal search paths for each UAV for all tested algorithms. In the figure, the purple path indicates the path generated by UAV1, and the white path indicates the path generated by UAV2. The path generated by the CC-MPSO algorithm closely approximates the target's motion trajectory. Both UAVs successfully locate the nearest high-probability grid to their initial positions at the end of the path, demonstrating the strong predictive capability of the CC-MPSO algorithm.



(**a**) CC-MPSO (**b**) PSO (**c**) MPSO

(**d**) ECPO-ME (**e**) ACO (**f**) RI-MAC

Figure 7. Search path for scenario 1 generated by the tested algorithms.



(**a**) CC-MPSO (**b**) PSO (**c**) MPSO

(**d**) ECPO-ME (**e**) ACO (**f**) RI-MAC

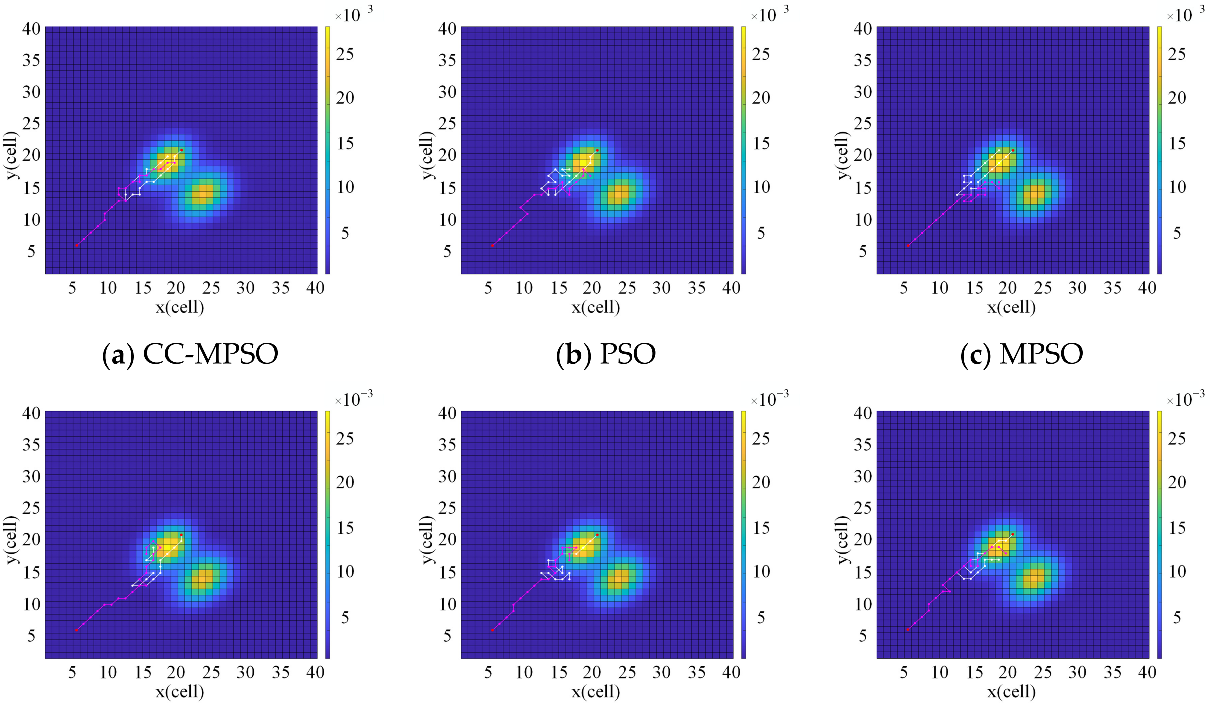
Figure 8. Search path for scenario 2 generated by the tested algorithms.

(3) Scenario 3

The results for scenario 3 are tabulated in Table 4. The acquired results indicate the following conclusions:

- For UAV1, the CC-MPSO algorithm achieves optimal results in the three statistical parameters “BEST”, “MEAN”, and “MEDIAN”. “RI-MAC” achieves the best results in “WORST” and “SD”.
- For UAV2, the CC-MPSO algorithm achieves the best results in “ BEST”, “WORST”, “MEAN”, and “SD”; RI-MAC achieves the best results in “MEDIAN”; and RI-MAC achieves optimal results in “BEST”, “WORST”, “MEAN”, and “SD”.

Figure 9 displays the most efficient search routes produced by all the algorithms that were tested in scenario 3. The purple trajectory represents the flight route of UAV1, whereas the white trajectory represents the flight path of UAV2. UAV1 starts at the bottom left corner of the map, which is near the high probability target area. The CC-MPSO algorithm enables UAV1 to search in the direction of the target’s path. UAV2 initially moves in the opposite direction of the target during its search. As the target moves further away, UAV2 can dynamically change its search direction to the opposite side. This demonstrates the strong adaptive capability of the CC-MPSO algorithm when searching along the diagonal.



(**a**) CC-MPSO (**b**) PSO (**c**) MPSO

(**d**) ECPO-ME (**e**) ACO (**f**) RI-MAC

Figure 9. Search path for scenario 3 generated by the tested algorithms.

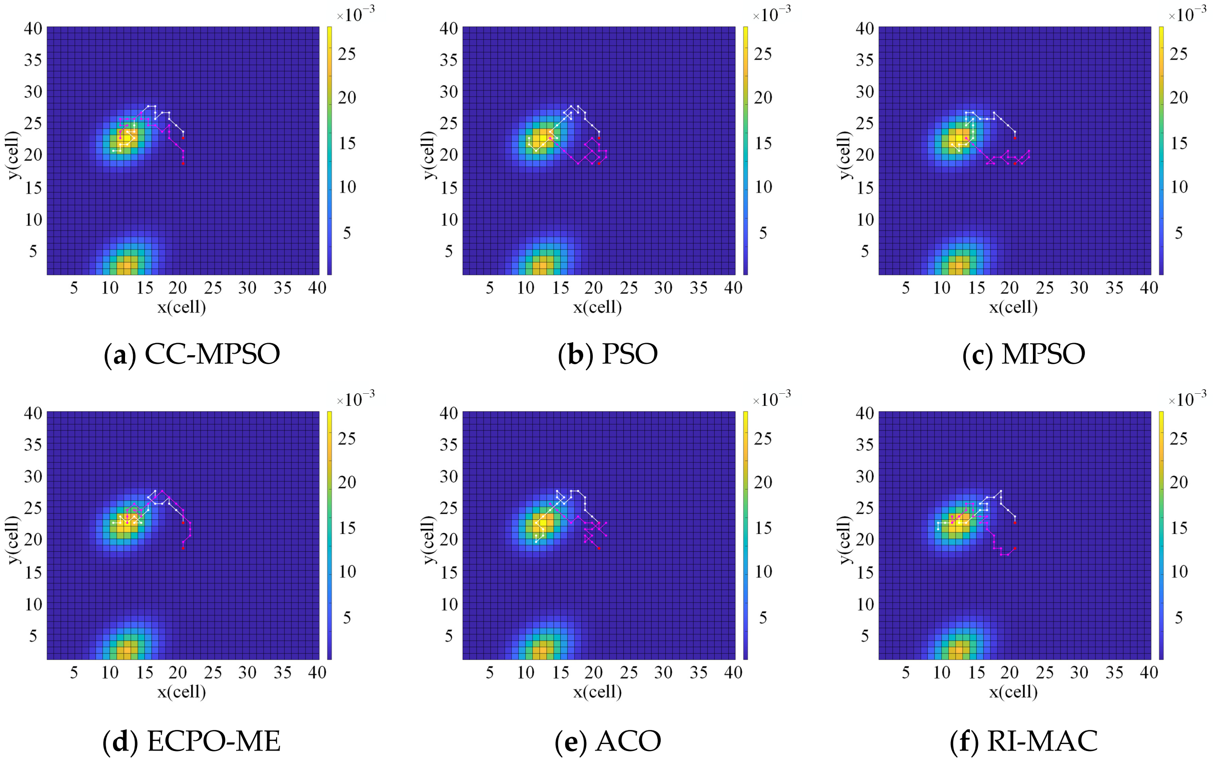
(4) Scenario 4

The results for scenario 4 are tabulated in Table 5. The acquired results indicate the following conclusions:

- For UAV1, the CC-MPSO algorithm achieves optimal results in “BEST”, “MEAN”, and “MEDIAN”, while the RI-MAC algorithm achieves optimal results in the rest of the statistical parameters.
- For UAV2, CC-MPSO achieves optimal results in “BEST” and “MEDIAN”, while RI-MAC achieves optimal results in the rest of the statistical parameters.

- As the scenario becomes more complicated, the "SD" value of PSO, MPSO, and ACO gradually becomes larger. However, CC-MPSO can maintain a smaller value, indicating its superior robustness.

Figure 10 displays the most efficient search routes produced by all examined algorithms in scenario 4. As the scenario becomes more complex, the paths produced by other algorithms exhibit increasing bias and yield a small percentage of erroneous paths. However, the search path nodes produced by the CC-MPSO method maintain a strong alignment with regions of high probability. The results suggest that the CC-MPSO algorithm may successfully find the target's location with high probability within a limited search period



(**a**) CC-MPSO (**b**) PSO (**c**) MPSO

(**d**) ECPO-ME (**e**) ACO (**f**) RI-MAC

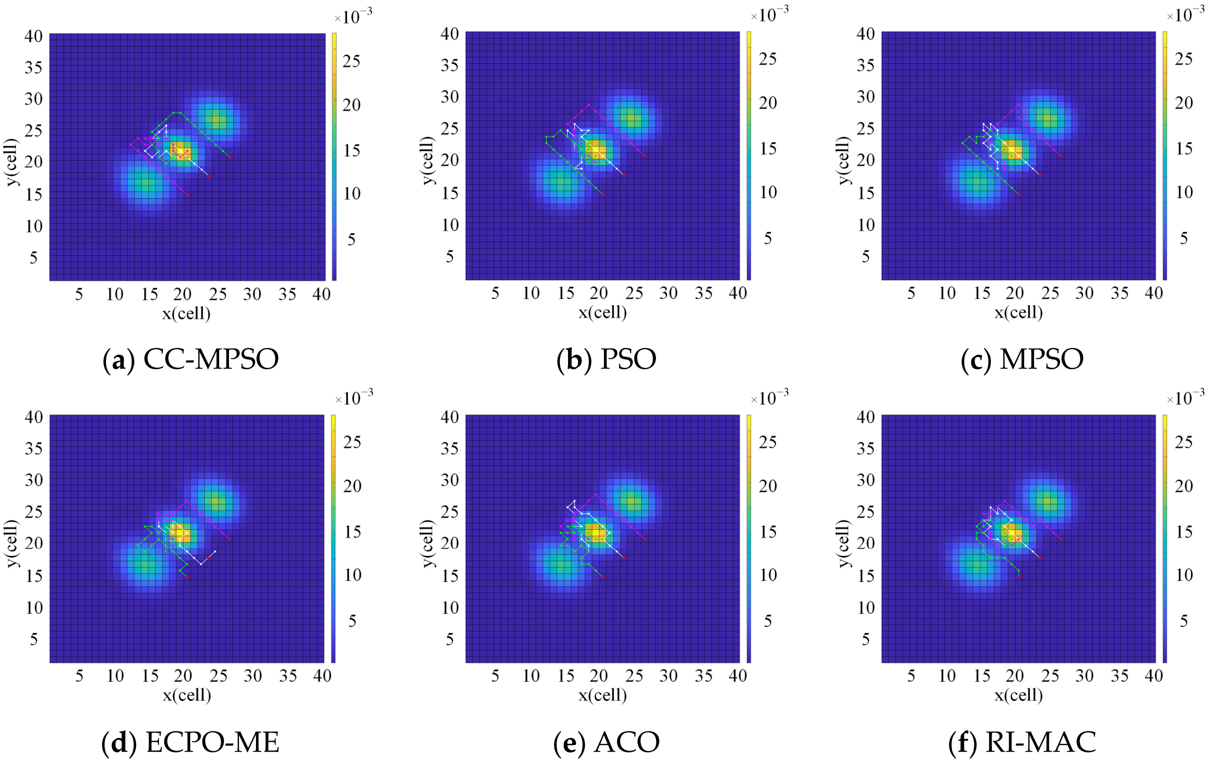
Figure 10. Search path for scenario 4 generated by the tested algorithms.

(5) Scenario 5

The results for scenario 5 are tabulated in Table 6. The acquired results indicate the following conclusions:

- For UAV1, CC-MPSO achieves optimal results in all statistical parameters. In comparison to other algorithms, the results are far more advanced.
- For UAV2, CC-MPSO achieves optimal results in "BEST", "MEAN", "MEDIAN", and "SD", and "RI-MAC" achieves optimal results in "WORST".
- For UAV3, CC-MPSO achieves optimal results in "BEST", "MEDIAN", and "SD", and RI-MAC achieves optimal results in the rest.

Figure 11 depicts the most efficient search paths produced by all the algorithms that were tested in scenario 5. The purple line represents the trajectory of UAV1, the white path represents the trajectory of UAV2, and the green path represents the trajectory of UAV3. There are three adjacent regions with a high probability of occurrence simultaneously, and they are located distant from the initial position of the UAV. The picture demonstrates that the CC-MPSO algorithm efficiently identifies the area with the highest probability and surrounds it in a circular situation. The remaining algorithms' generated pathways exhibit some divergence from the high-probability target region.



(**a**) CC-MPSO (**b**) PSO (**c**) MPSO

(**d**) ECPO-ME (**e**) ACO (**f**) RI-MAC

Figure 11. Search path for scenario 5 generated by the tested algorithms.

(6) Scenario 6

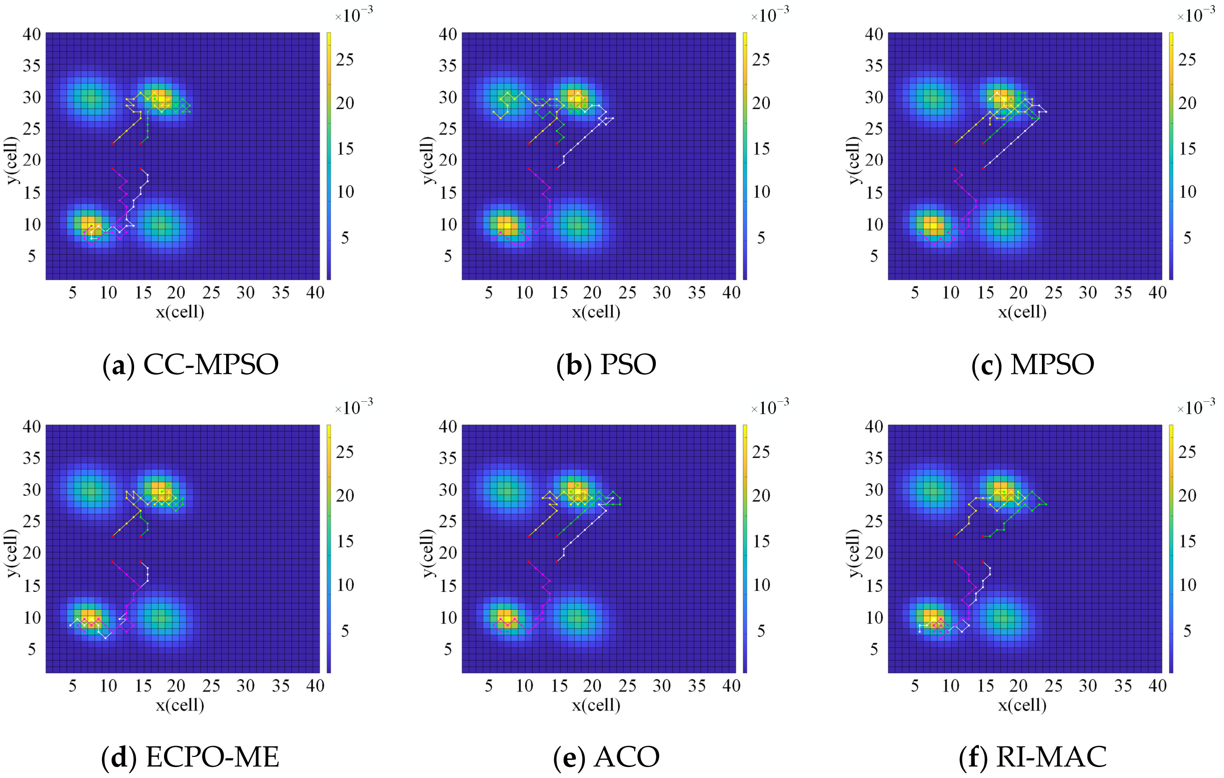
The results for scenario 6 are tabulated in Table 7. The acquired results indicate the following conclusions:

- For UAV1, CC-MPSO achieves optimal results in "BEST", "WORST", and "MEAN"; RI-MAC is optimal in "MEDIAN"; and ECPO-ME is optimal in "SD".
- For UAV2, CC-MPSO is optimal in "BEST", "MEAN", "MEDIAN", and "SD", and RI-MAC is optimal in "WORST".
- For UAV3, CC-MPSO is optimized in "MEAN" and "SD", and RI-MAC is optimized in "BEST", "WORST", and "MEDIAN".
- For UAV4, CC-MPSO achieves the best results in "BEST", "MEAN", and "MEDIAN", and RI-MAC achieves the best results in "WORST" and "SD".

Figure 12 depicts the most efficient search routes produced by all the algorithms examined in scenario 6. In scenario 6, there are two sets of regions with high likelihood, each consisting of two adjacent regions. One set is located above the map, while the other is located below the map. The four UAVs are positioned centrally between the two clusters of regions. The pathways produced by the PSO, MPSO, and ACO algorithms do not immediately lead to the optimal region. In addition, the paths generated by the ECPO-ME and RI-MAC algorithms can reach the high-probability target area, but their paths produce some invalid nodes. Compared to other algorithms, the CC-MPSO algorithm shows excellent performance in generating search paths that are not only unattracted by interfering target regions but also find the optimal region quickly and directly.

4.4. Analysis of Convergence Behavior

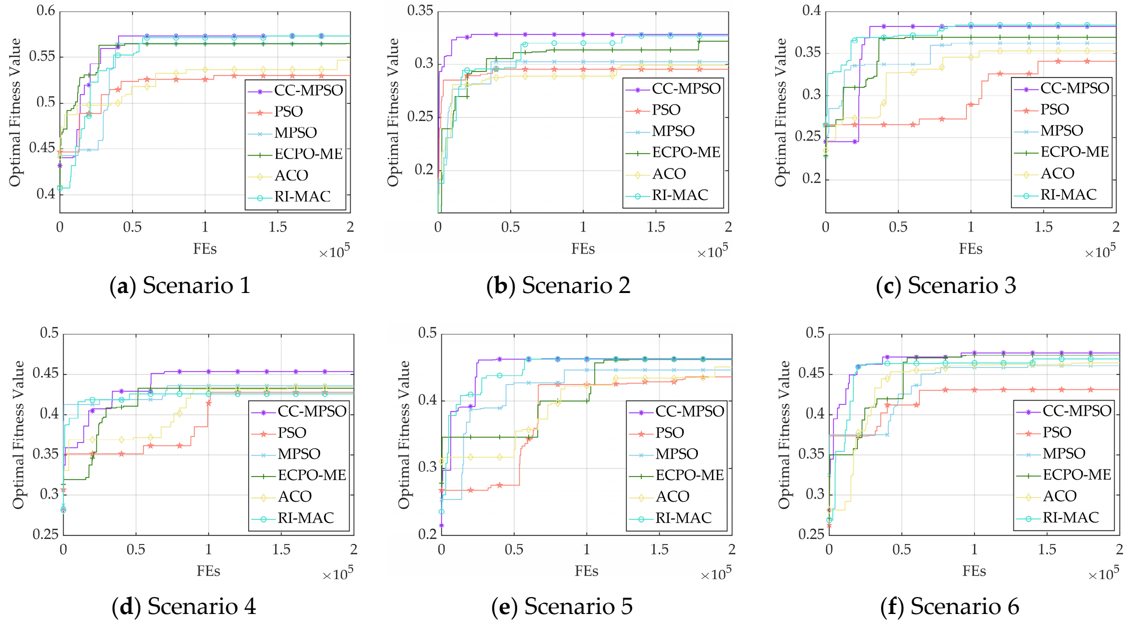
To investigate the convergence behavior of the CC-MPSO algorithm in the above six scenarios, a convergence analysis was performed based on FEs. FEs denotes the maximum number of fitness evaluations of each algorithm, where $FEs = NP \times MaxIt$. NP is the total population size of all swarms and $MaxIt$ is the maximum number of iterations. To conduct a fair comparison, the FEs of each algorithm was set to 2×10^5. The convergence curves are summarized in Figure 13. The optimal fitness value in Figure 13 corresponds to the combined probability of multi-UAVs detecting a target. It can be seen from Figure 13 that the convergence rate of CC-MPSO does not slow down, although we use the multi-swarm cooperative evolutionary framework to improve the diversity of the population. In all scenarios, the convergence rate of CC-MPSO is fastest, and it obtains the highest fitness value in scenarios 2, 4, 5, and 6. To summarize, CC-MPSO effectively maintains a balance between population diversity and convergence speed in order to determine the optimal fitness value.



(**a**) CC-MPSO (**b**) PSO (**c**) MPSO

(**d**) ECPO-ME (**e**) ACO (**f**) RI-MAC

Figure 12. Search path for scenario 6 generated by the tested algorithms.



(**a**) Scenario 1 (**b**) Scenario 2 (**c**) Scenario 3

(**d**) Scenario 4 (**e**) Scenario 5 (**f**) Scenario 6

Figure 13. Convergence behavior of the tested algorithms in six different scenarios.

4.5. Discussion

The obtained results in this study demonstrate the superiority of CC-MPSO over the well-known and widely used algorithms considered in this paper. The primary factor contributing to this effectiveness stems from the following aspects:

Firstly, the algorithm is directed to explore the motion space rather than the Cartesian space by the innovative motion encoding mechanism. This method effectively stops the algorithm from generating useless paths throughout the search phase, resulting in fewer search failures. Meanwhile, it can significantly increase algorithm search performance and accelerate convergence speed, making it more suitable for dealing with dynamic target search problems.

Secondly, the algorithm employs a problem decomposition strategy to break down the multi-UAV target search problem into several subproblems, with each subproblem being addressed by a solitary particle. Decomposition is beneficial in handling the intricacy of the problem and enhancing the scalability of the method.

Thirdly, the implementation of the co-evolution mechanism enables various subproblems to collaborate and achieve global search by exchanging knowledge. This aids in circumventing the occurrence of local optima and enhancing the algorithm's ability to search globally.

Based on the statistical analysis, the CC-MPSO algorithm successfully optimizes 52 out of 75 statistical elements. It also shows suboptimal performance in the remaining statistical items, with the majority of suboptimal outputs having a difference of barely one to two percent compared to the optimal ones. These differences are almost insignificant in actual applications. Furthermore, the simulation results have shown that the target search method, which is based on RI-MAC, also has fine performance. The reason for this can be linked to the algorithm's implementation of an association mechanism for prior information. In this mechanism, the target probability graph is continuously modified based on the association rules matched by various sorts of targets. It can be utilized to augment the performance of the CC-MPSO algorithm in the same way. Nevertheless, the pressing task at hand is to improve the target association rule base in order to align it more successfully with real-world application circumstances.

5. Conclusions

This paper uses a novel approach based on a variant of the famous particle swarm optimization algorithm using a cooperative coevolution framework and a motion-encoded mechanism to search for dynamic targets using multiple UAVs. On the one hand, the target search problem was converted from a Cartesian problem to a motion-based one via space transformation using the motion encoding feature. This transformation effectively prevents the algorithm from generating invalid search paths in the search process. On the other hand, the introduction of the co-evolution mechanism allows individuals to be guided by dynamic elite individuals in multiple populations during the updating process, thus preventing the algorithm from falling into premature convergence. The created algorithm was tested on six distinct scenarios with increasing levels of complexity, using different quantities of UAVs. The algorithm's performance was assessed by comparing it with five well-regarded swarm-based intelligent algorithms. The analysis of the results showcased the efficacy and dependability of the suggested approach when employed for dynamic target search with numerous UAVs.

This paper solely focused on cumulative probability in the objective function. Additional objectives, such as minimizing fuel usage and avoiding restricted regions, barriers, and other limitations, may be explored in future research. These factors would transform the problem into a multi-objective one, which would require future extensions. Meanwhile, the primary focus of this paper is algorithmic investigation and subsequent validation through simulation. Nevertheless, real experiments are crucial and indispensable. Exploring the practical implementation of the algorithms proposed in this study for real search situations is a crucial area of future study.

Author Contributions: Conceptualization, Y.L. and B.F.; methodology, Y.L.; software, Y.L.; validation, Y.L., W.C. and Z.W.; formal analysis, Y.L.; investigation, Y.L.; resources, L.H.; data curation, G.Y.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L.; visualization, L.H.; supervision, G.Y.; project administration, G.Y.; funding acquisition, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Nature Science Foundation of China, grant number 11774432.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to confidentiality agreement.

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

References

1. Nagasawa, R.; Mas, E.; Moya, L.; Koshimura, S. Model-Based Analysis of Multi-UAV Path Planning for Surveying Postdisaster Building Damage. *Sci. Rep.* **2021**, *11*, 18588. [CrossRef]
2. Shakeri, R.; Al-Garadi, M.A.; Badawy, A.; Mohamed, A.; Khattab, T.; Al-Ali, A.K.; Harras, K.A.; Guizani, M. Design Challenges of Multi-UAV Systems in Cyber-Physical Applications: A Comprehensive Survey and Future Directions. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3340–3385. [CrossRef]
3. Sun, Q.; Chen, L.; Xu, X.; Gu, X.; Hu, X.; Yang, F.; Pan, Y. A New Comprehensive Index for Monitoring Maize Lodging Severity Using UAV-Based Multi-Spectral Imagery. *Comput. Electron. Agric.* **2022**, *202*, 107362. [CrossRef]
4. Guerrero-Sánchez, M.E.; Hernández-González, O.; Valencia-Palomo, G.; Mercado-Ravell, D.A.; López-Estrada, F.R.; Hoyo-Montaño, J.A. Robust IDA-PBC for under-Actuated Systems with Inertia Matrix Dependent of the Unactuated Coordinates: Application to a UAV Carrying a Load. *Nonlinear Dyn.* **2021**, *105*, 3225–3238. [CrossRef]
5. George, J.; Sujit, P.B.; Sousa, J.B. Search Strategies for Multiple UAV Search and Destroy Missions. *J. Intell. Robot Syst.* **2011**, *61*, 355–367. [CrossRef]
6. Husain, Z.; Al Zaabi, A.; Hildmann, H.; Saffre, F.; Ruta, D.; Isakovic, A.F. Search and Rescue in a Maze-like Environment with Ant and Dijkstra Algorithms. *Drones* **2022**, *6*, 273. [CrossRef]
7. Duchoň, F.; Babinec, A.; Kajan, M.; Beňo, P.; Florek, M.; Fico, T.; Jurišica, L. Path Planning with Modified a Star Algorithm for a Mobile Robot. *Procedia Eng.* **2014**, *96*, 59–69. [CrossRef]
8. Wirth, L.; Oettershagen, P.; Ambühl, J.; Siegwart, R. Meteorological Path Planning Using Dynamic Programming for a Solar-Powered UAV. In Proceedings of the 2015 IEEE Aerospace Conference, Big Sky, MT, USA, 7–14 March 2015; pp. 1–11.
9. Zhao, Y.; Zheng, Z.; Liu, Y. Survey on Computational-Intelligence-Based UAV Path Planning. *Knowl.-Based Syst.* **2018**, *158*, 54–64. [CrossRef]
10. Mavrovouniotis, M.; Li, C.; Yang, S. A Survey of Swarm Intelligence for Dynamic Optimization: Algorithms and Applications. *Swarm Evol. Comput.* **2017**, *33*, 1–17. [CrossRef]
11. Slowik, A.; Kwasnicka, H. Nature Inspired Methods and Their Industry Applications—Swarm Intelligence Algorithms. *IEEE Trans. Ind. Inf.* **2018**, *14*, 1004–1015. [CrossRef]
12. Tang, J.; Liu, G.; Pan, Q. A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems: Applications and Trends. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 1627–1643. [CrossRef]
13. Phung, M.D.; Ha, Q.P. Motion-Encoded Particle Swarm Optimization for Moving Target Search Using UAVs. *Appl. Soft Comput.* **2020**, *97*, 106705. [CrossRef]
14. Alanezi, M.A.; Bouchekara, H.R.E.H.; Shahriar, M.S.; Sha'aban, Y.A.; Javaid, M.S.; Khodja, M. Motion-Encoded Electric Charged Particles Optimization for Moving Target Search Using Unmanned Aerial Vehicles. *Sensors* **2021**, *21*, 6568. [CrossRef] [PubMed]
15. Morin, M.; Abi-Zeid, I.; Quimper, C.-G. Ant Colony Optimization for Path Planning in Search and Rescue Operations. *Eur. J. Oper. Res.* **2023**, *305*, 53–63. [CrossRef]
16. Chen, T.; Chen, S.; Zhang, K.; Qiu, G.; Li, Q.; Chen, X. A Jump Point Search Improved Ant Colony Hybrid Optimization Algorithm for Path Planning of Mobile Robot. *Int. J. Adv. Robot. Syst.* **2022**, *19*, 172988062211279. [CrossRef]
17. Alanezi, M.A.; Bouchekara, H.R.E.H.; Apalara, T.A.-A.; Shahriar, M.S.; Sha'aban, Y.A.; Javaid, M.S.; Khodja, M.A. Dynamic Target Search Using Multi-UAVs Based on Motion-Encoded Genetic Algorithm with Multiple Parents. *IEEE Access* **2022**, *10*, 77922–77939. [CrossRef]
18. Garg, V.; Tiwari, R.; Shukla, A.; Dhar, J. A Distributed Cooperative Approach for Dynamic Target Search Using Particle Swarm Optimization with Limited Intercommunication. *Arab J. Sci. Eng.* **2022**, *47*, 10623–10637. [CrossRef]
19. Yue, W.; Tang, W.; Wang, L. Multi-UAV Cooperative Anti-Submarine Search Based on a Rule-Driven MAC Scheme. *Appl. Sci.* **2022**, *12*, 5707. [CrossRef]

20. Duan, H.; Zhao, J.; Deng, Y.; Shi, Y.; Ding, X. Dynamic Discrete Pigeon-Inspired Optimization for Multi-UAV Cooperative Search-Attack Mission Planning. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 706–720. [CrossRef]
21. Ma, X.; Li, X.; Zhang, Q.; Tang, K.; Liang, Z.; Xie, W.; Zhu, Z. A Survey on Cooperative Co-Evolutionary Algorithms. *IEEE Trans. Evol. Comput.* **2019**, *23*, 421–441. [CrossRef]
22. Kapnopoulos, A.; Alexandridis, A. A Cooperative Particle Swarm Optimization Approach for Tuning an MPC-Based Quadrotor Trajectory Tracking Scheme. *Aerosp. Sci. Technol.* **2022**, *127*, 107725. [CrossRef]
23. Alexandridis, A.; Chondrodima, E.; Sarimveis, H. Cooperative Learning for Radial Basis Function Networks Using Particle Swarm Optimization. *Appl. Soft Comput.* **2016**, *49*, 485–497. [CrossRef]
24. Singh, S.; Krishnamurthy, V. The Optimal Search for a Markovian Target When the Search Path Is Constrained: The Infinite-Horizon Case. *IEEE Trans. Automat. Contr.* **2003**, *48*, 493–497. [CrossRef]
25. Fu, Y.; Ding, M.; Zhou, C. Phase Angle-Encoded and Quantum-Behaved Particle Swarm Optimization Applied to Three-Dimensional Route Planning for UAV. *IEEE Trans. Syst. Man Cybern. A* **2012**, *42*, 511–526. [CrossRef]
26. Potter, M.A.; De Jong, K.A. A Cooperative Coevolutionary Approach to Function Optimization. In *Parallel Problem Solving from Nature—PPSN III*; Springer: Berlin/Heidelberg, Germany, 1994; pp. 249–257.
27. van den Bergh, F.; Engelbrecht, A.P. A Cooperative Approach to Particle Swarm Optimization. *IEEE Trans. Evol. Computat.* **2004**, *8*, 225–239. [CrossRef]
28. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; IEEE: Piscateville, NJ, USA, 1995; Volume 4, pp. 1942–1948.

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.