Multi-UAV Collaborative Search and Attack Mission Decision-Making in Unknown Environments

Zibin Liang 1,2,3, Qing Li 1,2,3,* and Guodong Fu 1,2,3

1 Beijing Key Laboratory of High Dynamic Navigation Technology, Beijing 100192, China; zibinl@163.com (Z.L.); fuguodd@163.com (G.F.)
2 Ministry of Education Key Laboratory of Modern Measurement & Control Technology, Beijing 100101, China
3 School of Automation, Beijing Information Science & Technology University, Beijing 100192, China
* Correspondence: liqing@bistu.edu.cn

Abstract: To address the challenge of coordinated combat involving multiple UAVs in reconnaissance and search attacks, we propose the Multi-UAV Distributed Self-Organizing Cooperative Intelligence Surveillance and Combat (CISCs) strategy. This strategy employs distributed control to overcome issues associated with centralized control and communication difficulties. Additionally, it introduces a time-constrained formation controller to address the problem of unstable multi-UAV formations and lengthy formation times. Furthermore, a multi-task allocation algorithm is designed to tackle the issue of allocating multiple tasks to individual UAVs, enabling autonomous decision-making at the local level. The distributed self-organized multi-UAV cooperative reconnaissance and combat strategy consists of three main components. Firstly, a multi-UAV finite time formation controller allows for the rapid formation of a mission-specific formation in a finite period. Secondly, a multi-task goal assignment module generates a task sequence for each UAV, utilizing an improved distributed Ant Colony Optimization (ACO) algorithm based on Q-Learning. This module also incorporates a colony disorientation strategy to expand the search range and a search transition strategy to prevent premature convergence of the algorithm. Lastly, a UAV obstacle avoidance module considers internal collisions and provides real-time obstacle avoidance paths for multiple UAVs. In the first part, we propose a formation algorithm in finite time to enable the quick formation of multiple UAVs in a three-dimensional space. In the second part, an improved distributed ACO algorithm based on Q-Learning is introduced for task allocation and generation of task sequences. This module includes a colony disorientation strategy to expand the search range and a search transition strategy to avoid premature convergence. In the third part, a multi-task target assignment module is presented to generate task sequences for each UAV, considering internal collisions. This module provides real-time obstacle avoidance paths for multiple UAVs, preventing premature convergence of the algorithm. Finally, we verify the practicality and reliability of the strategy through simulations.

Keywords: multi-UAV; coordinated operations; Q-learning; ant colony algorithm; artificial potential field

1. Introduction

In recent years, with the increase in mission complexity and the enhancement of UAV autonomy, unmanned aircraft swarm systems (UAS) have gradually received attention from researchers. Drone swarms have been used for target search and rescue [1,2], cluster operation [3,4], and target tracking [5,6]. Within the military domain, coordinated reconnaissance search attacks by UAV swarms in unknown environments is an important application. Drone swarm reconnaissance operations can provide more information, faster response times, greater accuracy, and lower costs. UAV swarm missions are characterized...
by complexity and dynamics, making multi-objective tasking of multiple UAVs a challenge [7,8].

Task allocation is the assignment of a task to an intelligence or a group of intelligences, with the goal of finding an optimal or near-optimal mapping between an intelligence and a task [9]. Auction or market-based approaches are commonly used in multi-robot task allocation problems [10,11], including AsyMTRE-D [12], Murdoch [13], M+ [14], and TraderBots [15]. Graph-theoretic particle swarm optimization for military applications [16], Sandholm’s algorithm with K-means clustering [17], Efficiency-based tasking [18], and many other approaches focus on minimizing the distance carried out by the robot to complete the task while satisfying the robot’s own resource allocation. Borja Fernandez-Gauna et al. developed a distributed round-robin Q-Learning for the transportation of hoses using UGVs [19]. Mathew et al. developed path planning algorithms for miniature UAVs and UGVs performing parcel delivery tasks [20]; the problem is modeled as a multi-warehouse delivery problem, transformed into a generalized traveler’s problem, and solved in polynomial time using a kernel sequence enumeration algorithm. Bayesian Reinforcement Learning (RL) allows the agents to learn the capabilities of other agents through interactions and transform repetitive coalition formation problems into sequential decision problems [21]. This approach was validated in the soccer team formation problem [22]. Modeling Dynamic Robot Formation for the Area Coverage Problem Using Weighted Voting Games and Q-Learning, and Extended to Formation-Based Navigation Problems [23,24]. The formation structure is pruned using Shapley values and marginal contributions, and the transition of robots from one formation to another is represented by the Markov process, which searches for the optimal structure of the formation space using a Markov probability distribution. Sayan D. et al. proposed a multi-criteria decision-making algorithm based on influence diagrams to select the best formation algorithm for a given realistic scenario [25]. Pengxing Zhu et al. constructed a heterogeneous UAV multi-task assignment model based on multiple realistic constraints and proposed an improved semi-randomized Q-Learning algorithm, which was demonstrated through simulation to be able to improve the performance of task execution compared to heuristic algorithms such as the Q-Learning algorithm, including improving the reasonableness of the task assignment, with a gain value of improvement of 12.12% [26]. Han Qi et al. solved the target assignment and trajectory planning separately and proposed a multi-intelligence target assignment and trajectory planning method based on a multi-intelligence reinforcement learning algorithm [27]. Minchan Li et al. proposed an effective stability quantum particle swarm optimization algorithm (SQPSO) and designed an efficient coalition-building quantum particle swarm optimization algorithm (EQPSO), which incorporates coalition similarity judgments and reduces coalition formation time overheads [28]. Heba Kurdi et al. proposed a bacterial heuristic algorithm for UAV task assignment [29]. Xiang Ma et al. proposed an improved virtual force method to obtain the optimal damage effectiveness control strategy, which solves the numerical solution problem caused by the high complexity of the damage function [30]. Xia Chen et al. proposed a systematic framework for solving the combinatorial optimization model with a new particle swarm optimization algorithm based on a bootstrapping mechanism, and for the trajectory planning problem, an ant colony optimization algorithm based on adaptive parameter tuning and bidirectional search was proposed [31]. Taehoon Yoo et al. proposed a reinforcement learning-based intelligent formation controller that enables each UAV to communicate with other UAVs [32]. Yuchong Gao et al. proposed a scalable, purely azimuthal passive UAV formation keeping distributed control algorithm to maintain multi-UAV formation using pure angular information through only a small amount of inter-UAV communication [33].

In this paper, a multi-UAV-distributed self-organized collaborative intelligent reconnaissance operation strategy is proposed for the problem of collaborative reconnaissance operation of multi-UAV multi-mission targets. We have performed the following main tasks:
• A distributed self-organized multi-UAV cooperative intelligent reconnaissance combat strategy (CISCS) is proposed for the multi-UAV cooperative reconnaissance combat problem;
• A formation controller with time constraints is proposed to extend the multi-layer artificial potential field to avoid the internal collision of multiple UAVs in fast formation and to solve the problem where the traditional artificial potential field method tends to fall into the optimal solution;
• Reinforcement learning is utilized to accelerate the convergence speed of the ACO algorithm and expand the search capability, and a colony disorientation strategy and a search-switching strategy are incorporated to prevent its premature maturity.

We will discuss the above according to the following structure: Section 2 is the multi UAV multi-task target feature scene and task allocation decision model; Section 3 designs the overall structure of CISCS, including a formation controller with time constraints for fast formation, a distributed and improved ant colony algorithm based on Q-Learning for generating multi-UAV task sequences, and an extended artificial-potential field algorithm for drone avoidance within the drones formation; in Section 4, we conducted simulation and demonstration of this algorithm; and the conclusions are presented in Section 5.

2. Scene Description and Modeling

In this section, the search attack task in an uncertain environment is modeled; the search attack task planning problem is defined, and the constraints of the search attack task planning model are given.

Assumption 1. Each UAV is capable of performing reconnaissance search and attack missions.

\[ V = \{V_1, V_2, ..., V_{N_v}\} \] indicates \( N_v \) multi-drone formation. \( T = \{T_1, T_2, ..., T_{N_l}\} \) is the set of \( N_l \) unknown targets distributed within the task area.

2.1. Collaborative Search Attack Constraints

Multi-UAV coordinated search–attack mission is an online planning problem whose goal is to discover and destroy as many targets as possible in a dynamic and complex environment. Therefore, the primary problem is to develop a distributed optimization model for search–attack missions. A detailed description of these constraints is given below [34]:
1. Maximum range constraint;
2. Single UAV mission with maximum range \( D_{\text{max}} \):
   \[ D_{\text{max}} - D_i(t) \geq 0 (i = 1, 2, 3, ..., N_v) \] (1)
3. where \( D_i(t) \) denotes the cumulative distance traveled by the \( i \)th UAV before moment \( t \); \( D_{\text{max}}(t) \) denotes the maximum distance traveled;
4. Maximum turning angle constraint;
5. The drone has a maximum turning angle \( \theta_{\text{max}} \):
   \[ \theta_{\text{max}} - \theta_i(t) \geq 0 (i = 1, 2, 3, ..., N_v) \] (2)
6. where \( \theta_i(t) \) denotes the turning angle of the \( i \)th UAV at moment \( t \). The maximum turn angle limits the UAV turn radius and turn rate, \( \theta_{\text{max}} \in (0, \pi) \). When the desired turn angle is 0, it means that the drone does not need to turn; when the desired turn angle is \( \pi \), it means that the drone is expected to fly in the reverse direction;
7. Minimum flight distance between drones;
8. In order to ensure the flight safety of multi-UAV formations, there exists a safety distance for multi-UAVs to perform their tasks:
where \(d_{ij}(t)\) denotes the distance between the \(i\)th UAV and the \(j\)th UAV at time \(t\); \(d_{\text{min}}\) denotes the minimum flight distance between any two drones.

9. Risk area avoidance constraints:

\[
d_{il}(t) - R_{\text{min}} \geq 0 (i = 1, 2, 3, ..., N_v, l = 1, 2, 3, ..., N_i)
\]

where \(d_{il}(t)\) denotes the distance between the \(i\)th UAV and the \(l\)th risk zone at time \(t\); \(R_{\text{min}}\) denotes the minimum risk radius.

2.2. Multi-UAV Collaborative Mission Description

(1) Description of reconnaissance coverage

Ground reconnaissance coverage radius exists for UAVs during reconnaissance missions.

Reconnaissance coverage:

\[
P_r = \frac{\sum_{i=1}^{N_v} S_i}{S_{\text{area}}}
\]

where \(S_i\) denotes the reconnaissance area of the \(i\)th UAV; \(S_{\text{area}}\) denotes the total area of the combat zone.

Reconnaissance search benefit function:

\[
J_r = P_r
\]

(2) Description of the attack mission

Attack gain function:

\[
J_a = \sum_{i=1}^{N_v} \sum_{j=1}^{T} a_{ij} \text{score}_j
\]

where \(a_{ij}\) denotes that the \(i\)th UAV attacked the \(j\)th target; when \(a_{ij} = 0\), it means that the attack mission on the \(j\)th target was not accomplished; when \(a_{ij} = 1\), it means that the attack mission on the \(j\)th target was accomplished. \(\text{score}_j\) denotes the gain of the attack on the \(j\)th target.

2.3. Mathematical Model for Tasking

The coordinated multi-UAV search attack needs to maximize the benefit function under the above constraints, so the following centralized multi-UAV task allocation constraint model can be established:

\[
U^* = \arg \max_U (\eta J_r + (1 - \eta) J_a)
\]

subject to

\[
\begin{align*}
D_{\text{max}} - D_i(t) & \geq 0 \\
\theta_{\text{max}} - \theta_i(t) & \geq 0 \\
d_{ij}(t) - d_{\text{min}} & \geq 0 \quad (i, j = 1, 2, ..., N_v, i \neq j) \\
d_{il}(t) - R_{\text{min}} & \geq 0
\end{align*}
\]

where \(\eta \in (0, 1)\); when \(\eta = 0\), it denotes that the UAV performs an attack mission; when \(\eta = 1\), it denotes that the UAV performs a reconnaissance mission. \(U\) denotes the UAV decision-making. Assuming that each UAV is an independent individual and all have the ability to make independent decisions, the centralized optimization model (8) can be decomposed into a distributed optimization model:

\[
U^*_i = \arg \max_{U_i} (\eta J_{rl}(X_i, \tilde{x}_j) + (1 - \eta) J_{al}(X_i, \tilde{x}_j))
\]
where \( U' \) denotes the decision of the \( i \)th drone, \( f_{ri} \) denotes the reconnaissance gain of the \( i \)th UAV, \( f_{ai} \) denotes the gain from the \( i \)th drone attack; \( x_i \) denotes the state of the \( i \)th drone; \( \tilde{X}_f = \{x_j | j \in T^i_{\text{adjacent}} \} \), where \( T^i_{\text{adjacent}} \) denotes the set of UAVs that can communicate with the \( i \)th UAV.

### 3. Algorithms for Multi-Drone Collaborative Intelligent Warfare

Based on the above-distributed optimization model (9), multi-UAVs interact with the battlefield for information, real-time formation control of multi-UAVs, task assignment, and path planning. In this section, we will design a multi-UAV cooperative intelligent combat algorithm for the multi-UAV real-time target search and attack problem.

#### 3.1. Formation Controller with Time Constraints

When multiple UAVs fly in formation, the Leader sends position information to the follower, and the follower adjusts its desired position coordinates in real time according to the Leader’s position information and, eventually, forms into a stable formation [35]. In order to improve the rapidity of multi-UAV formation, we propose a finite time-based multi-UAV formation control algorithm, as shown in Equation (10).

\[
 u'_f(t) = k_f [P(\theta_i(t), \varphi_i(t))(d_f - d_i(t)) + k_i \int_{t_0}^{t_1} (d_f - d_i(t)) dt + k_d \frac{d}{dt} (d_f - d_i(t))] 
\]

\[ k_f = \frac{w}{t_w - t}, \quad t \in [0, t_w) \]  

In Equation (10), \( u'_f(t) \) is the input of UAV \( i \); \( k_f \) is the finite time gain variable; \( t_w \) indicates limited time; \( d_f \) is the distance between the drone and the formation target; \( d_i \) is the desired distance; \( d_i(t) \) is the distance between UAV \( i \) and the formation target at the moment \( t \); \( \theta_i(t) \) is the angle between the line of the leader and the follower and the \( z \)-axis positive direction, and \( \varphi_i(t) \) is the angle between the projection of this line on the XOY plane and the x-axis positive direction. \( P(\theta, \varphi) \) is \( \begin{bmatrix} \sin \theta \cos \varphi \\ \sin \varphi \cos \theta \\ \cos \theta \end{bmatrix} \), which is used to convert \( u'_f(t) \) to an output quantity in the \( x, y \), and \( z \) directions.

In Equation (11), \( w \) is the finite time gain normal, and \( t_w \) is the finite time parameter.

**Theorem 1.** For a multi-UAV system, the finite time gain \( k_f \) in Equation (11) satisfies that \( w \) is a positive constant and \( t_w \) is an arbitrarily given finite time; then, it is guaranteed that the multi-UAVs form a leader–follower flight formation at any time \( t_w \).

**Proof.** The Lyapunov function is now chosen as

\[
 V(d_i(t)) = \frac{1}{2} (d_f - d_i(t))^2 
\]

Derivation leads to

\[
 \dot{V}(d_i(t)) = -(d_f - d_i(t)) \dot{d}_i(t) 
\]

From the controller Equation (10)

\[
 \dot{V}(d_i(t)) = -(d_f - d_i(t)) \cdot \frac{w}{t_w - t} \left[ P + k_i + k_d \cdot d_f - P \cdot d_i(t) - k_i \int_0^{t_1} d_i(t) dt \right] 
\]

This proves the theorem.
\[-k_d \cdot \frac{d}{dt} d(t) \leq 0\]

Therefore, according to Lyapunov stability analysis, the given UAV distance controller is stable and the actual flight distance \(d(t)\) converges to the desired flight distance \(d_f\). This proves the convergence and stability of the controller. □

3.2. Reconnaissance Search Attack Tasking Algorithm

We propose an improved ant colony multi-task multi-objective solving method based on Q-Learning with constraints by utilizing the powerful feedback memory capability of Q-Learning [36], and the process of using this algorithm for UAV cooperation is described in detail below.

3.2.1. Q-Learning State Space

The ant colony position information is used as the state space of Q-Learning, which guides the later ant colonies to find the best solution through the historical optimal path and accelerates the convergence of the ant colony algorithm. The Euclidean distance between the current state \(x_t\) of the ant colony and the optimal solution of the previous state is taken as the current state \(S_t\) of Q-Learning, i.e.,

\[S_t = \|x_t - x_{t-1}\|\] (15)

where \(x_{t-1}\) denotes the optimal position at moment \(t - 1\).

Normalize \(S_t\)

\[
S_t = \begin{cases} 
HH, & 0.8 \leq S_t < 1.0 \\
HM, & 0.6 \leq S_t < 0.8 \\
MM, & 0.4 \leq S_t < 0.6 \\
LM, & 0.2 \leq S_t < 0.4 \\
LL, & 0 \leq S_t < 0.2
\end{cases}
\] (16)

where HH, HM, MM, LM, LL denote five different ant colony states, respectively.

3.2.2. Q-Learning Action Space

Since ant colony search mainly involves three different parameters [37]: pheromone-inspired factor \(\alpha\); expectation-inspired factor \(\beta\); and pheromone evaporation factor \(\rho\). In this paper, we adaptively adjust the size of the three parameters according to the environment and regulate the search capability into five grades, namely, full-domain search, local search, hold search, slow-step convergence, and fast convergence, which are used as the action space of Q-Learning. These five search modes correspond to preset parameter combinations, as shown in Table 1.

**Table 1. Ant colony update parameters.**

<table>
<thead>
<tr>
<th>Ant Colony Update Parameters</th>
<th>(\alpha)</th>
<th>(\beta)</th>
<th>(\rho)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Search</td>
<td>1.0</td>
<td>3.0</td>
<td>0.15</td>
</tr>
<tr>
<td>Local Search</td>
<td>1.25</td>
<td>3.50</td>
<td>0.25</td>
</tr>
<tr>
<td>Maintenance Search</td>
<td>1.50</td>
<td>4.0</td>
<td>0.45</td>
</tr>
<tr>
<td>Slow convergence</td>
<td>1.75</td>
<td>4.50</td>
<td>0.65</td>
</tr>
<tr>
<td>Fast Convergence</td>
<td>2.0</td>
<td>5.0</td>
<td>0.75</td>
</tr>
</tbody>
</table>

3.2.3. Reward Function

In order to speed up the process of ant colony optimization, setting reasonable rewards for the ant colony can guide the ant colony to search in a more optimal direction. Considering multiple objectives in a multi-objective optimization problem, the sum of the fitness differences before and after the ant colony update is taken as the reward value, i.e.,
where $r_{t,t+1}$ denotes the reward value of the ant colony after updating; $x_t$ and $x_{t+1}$ denote the positions of the colony before and after updating; $n$ denotes the number of objective functions; and $Fit_i(x_i)$ denotes the fitness of the $i$th objective function at $x_i$.

### 3.2.4. Q-Value Update

Learn the design of states, actions, and rewards based on the above Q. By judging the current state the colony is in, the action corresponding to the maximum Q value of that state in the Q table is selected [38]. The combination of parameters is selected based on the optimization of this action; the state information of the ant colony is updated by the new parameters; the rewards obtained from the updating process are calculated, and the Q-table is updated by combining the new states and actions of the colony in order to track, learn, and dynamically adjust the optimization process [39].

The update function for the Q-value is:

$$Q(S_t,a_t) = (1-\alpha)Q(S_t,a_t) + \alpha[r_t + \gamma \max_a Q(S_{t+1},a)]$$  \hspace{1cm} (18)

where $\alpha$ is the learning rate; $\gamma$ is the discount; $a_t$ denotes the taken action at moment $t$, i.e., the update parameter at moment $t$; $r_{t,t+1}$ denotes the reward value after the update of the ant colony, and $Q(S_{t+1},a_{t+1})$ denotes the action value of state $S_{t+1}$ taking action.

The algorithm flow of Q-learning is shown in Figure 1.

![Figure 1. Q-Learning Algorithmic.](image)

### 3.2.5. Ant Colony Lost

In order to prevent the ant colony algorithm from maturing prematurely and entering the local optimal solution, this paper proposes an ant colony lost strategy to increase the local search ability of the ant colony. As shown in Figure 2. When the ant colony selects the next node according to the transfer probability, it mimics the situation of ants becoming lost and sets the lost rate $\tau$, so that the ants do not arrive at the next node according to the agreement but are lost to another nearby node, thus increasing the local search ability of the ant colony.
Figure 2. Ant colony lost strategy.

It is now assumed that when the ant colony has entered into a locally optimal solution, but the colony is still in motion, there is a certain probability of the ant colony in the process of motion, i.e., the rate of disorientation; the ant colony does not follow the originally established route but is disoriented into another path, such as that of the ants in the blue circle in Figure 2, and enters into the other path, and if this disorientation obtains a new solution that is more superior to the previous one, then it means that the ant colony jumps out of the locally optimal solution.

3.2.6. Search Conversion Strategy

In the multi-task objective search optimization algorithm, the ant colony is prone to premature maturity and local optimization, which makes the ant colony algorithm unable to obtain the optimal solution and lose some of the optimization search samples in order to increase the diversity of sample search; this paper designs a multi-objective search transformation strategy, which is used to enhance the algorithm’s optimization searching ability, and balances the algorithm’s ability to explore globally and develop locally. On the basis of multi-task objective ant colony optimization, the variation strategy of jumping out of the local optimum is set for the ant colony. Firstly, the ant colony enters the global optimization session to obtain the short-lived optimal solution set G. Then, it enters the local search stage and performs random mutation based on Cauchy distribution on all the solutions in G and obtains the position of the ant colony after mutation. Finally, the newly generated ant colony is evaluated for fitness and ranked in comparison with the original ant colony in G, thus updating the G table. The two search methods are used to switch continuously during the optimization process to update the optimal set G [30].

In order to fully increase the diversity of search samples, this paper adopts a variant based on the Cauchy distribution, which is used to increase the local search capability of the ant colony. Cauchy distribution is suitable for exploring randomness.

Therefore, using the nature of the Cauchy distribution to combine it with the ant colony transfer probability function, the Cauchy operator is introduced into the ant colony transfer so that the optimal solution set is updated through the Cauchy distribution-based variation to improve the local exploration ability and increase the diversity of samples. The search transformation strategy improves the searchability of the ant colony and leads the ant colony to reject the local optimal solution with a certain probability, and its formula for the ant colony variation based on the Cauchy distribution is as follows:

\[ \text{Pos}_{\text{next}} = \begin{cases} \text{Pos}_{\text{rand}} & \text{if } \text{Pos}_{\text{rand}} \leq \tau \\ \text{Pos}_{\text{else}} & \text{otherwise} \end{cases} \]  

(19)
where \( i, j \) are the start and end points, respectively; \( \alpha, \beta \) are the weighting factors; \( \tau \) is the pheromone concentration; \( \eta \) is the visibility, which is the inverse of the distance between points \( i, j \); and \( C \) is the set of unvisited nodes.

3.2.7. Improved Ant Colony Multi-Task Objective Optimization Algorithm Based on Q-Learning

The ant colony is used as the intelligent body of Q-Learning, and the ant colony updates its own action parameters through its behavioral state, which realizes the expansion of the search range, accelerates the convergence speed of the algorithm, completes the problem of solving the multi-task objective, and overcomes the problems of the algorithm being easy to enter into the locally optimal solution and precocious maturity. The algorithm flow is shown in Algorithm 1.

\[
p_{ij}^k(t) = \left( \sum_{k \in C_i} \tau_{ij}^a \eta_{ij}^b \right) + \frac{1}{2} \left( \frac{t - \tau_{ij}}{p_{ij}} \right), \quad j \in C_i
\]

Algorithm 1: Improved Ant Colony Multi-Task Objective Optimization Algorithm Based on Q-Learning

<table>
<thead>
<tr>
<th>Step 1: Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the maximum number of iterations ( t_{\text{max}} ), maximum number of ants ( m_{\text{max}} ), ( \alpha, \beta, \eta ) parameters</td>
</tr>
<tr>
<td>Initialize the total amount of pheromones, Q table, and global optimal solution set ( G )</td>
</tr>
<tr>
<td>Initialize the ant colony and randomly allocate starting points for the ants</td>
</tr>
<tr>
<td>Set initial pheromone concentration</td>
</tr>
<tr>
<td>Set evaluation function and multi-task goal planning task</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2: Ant colony search</th>
</tr>
</thead>
<tbody>
<tr>
<td>For ( t = 1 ) to ( t_{\text{max}} ) do</td>
</tr>
<tr>
<td>For ( m = 1 ) to ( m_{\text{max}} ) do</td>
</tr>
<tr>
<td>Calculate the probability of movement for each ant</td>
</tr>
<tr>
<td>Choose the next moving node</td>
</tr>
<tr>
<td>Ant loss strategy, choose whether to be lost in the path with a certain probability</td>
</tr>
<tr>
<td>Update the pheromone concentration</td>
</tr>
<tr>
<td>Evaluate the ant colony, update the optimal solution set ( G )</td>
</tr>
<tr>
<td>Calculate ant reward, update the action value function</td>
</tr>
<tr>
<td>End For</td>
</tr>
<tr>
<td>Generate a transient optimal solution set ( G )</td>
</tr>
<tr>
<td>Perform search transformation strategy, update ( G )</td>
</tr>
<tr>
<td>End For</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3: Output the results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output the global optimal solution set ( G )</td>
</tr>
</tbody>
</table>

3.2.8. Extended Artificial Potential Field Algorithm (EAPF)

To ensure that the flight safety of multi-UAVs in the mission avoids internal collision [40], we designed a multi-UAV drone avoidance control algorithm based on the traditional artificial potential field method. The algorithm extends the multilayer potential field function to improve the UAV avoidance performance based on the traditional artificial potential field algorithm with a gravitational potential field of

\[
U_{\text{att}}(p) = \begin{cases} 
\alpha_1 (p - p_0)^2, & d \in (0, d_0) \\
2d_0 \alpha_1 (p - p_0), & d \in (d_0, +\infty) 
\end{cases}
\]

where \( U_{\text{att}}(p) \) is the potential field function; \( \alpha_1 \) denotes the gravitational coefficient; \( p \) denotes the current position of the UAV; \( p_0 \) is danger zone; \( d \) is the Euclidean distance
between the UAV and the target, and $d_o$ is the distance threshold; when the distance between the position of the UAV and the location of the target is greater than the set distance threshold $d_o$, the gravitational force is constant. When the distance between the UAV and the target is smaller than the set distance threshold $d_o$, the UAV can be considered to be moving within the influence distance of the target point.

The repulsive potential field function is:

$$U_{rep}(p) = \begin{cases} 
0, & d \in (d_1, +\infty) \\
\beta_1 \left(1 - \frac{1}{d_0}\right)^2 (p - p_0)^{\gamma_1}, & d \in (d_0, d_1) \\
2\beta_1 \left(1 - \frac{1}{d_0}\right)^2 (p - p_0)^2, & d \in (0, d_0) 
\end{cases}$$  \hspace{1cm} (22)

where $U_{rep}(p)$ denotes the repulsive potential field function; $\beta_1$ denotes the repulsive coefficient; $\gamma_1$ denotes the repulsive factor; and $d_1$ denotes the distance threshold.

The gravitational function is

$$F_{att}(p) = -\nabla U_{att}(p) = \begin{cases} 
-2\alpha_1 (p - p_0), & d \in (0, d_0) \\
-\frac{\alpha_1 d_0 (p - p_0)}{\|p - p_0\|}, & d \in (d_0, +\infty) 
\end{cases}$$  \hspace{1cm} (23)

The repulsion function is

$$F_{rep}(p) = -\nabla U_{rep}(p) = \begin{cases} 
0, & d \in (d_1, +\infty) \\
F_1 + F_2, & d \in (d_0, d_1) \\
F_1 + F_2, & d \in (0, d_0) 
\end{cases}$$  \hspace{1cm} (24)

where $F_1$ and $F_2$ are

$$F_1(p) = \begin{cases} 
2\beta_1 \left(1 - \frac{1}{d_0}\right)^2 \left(\frac{p - p_0}{d_0}\right)^{\gamma_1}, & d \in (d_0, d_1) \\
2\beta_1 \left(1 - \frac{1}{d_0}\right)^2 \left(\frac{p - p_0}{d_0}\right)^2, & d \in (0, d_0) 
\end{cases}$$  \hspace{1cm} (25)

$$F_2(p) = \begin{cases} 
2\beta_1 \left(1 - \frac{1}{d_0}\right) (p - p_0)^{\gamma_1 - 1}, & d \in (d_0, d_1) \\
2\beta_1 \left(1 - \frac{1}{d_0}\right) (p - p_0), & d \in (0, d_0) 
\end{cases}$$  \hspace{1cm} (26)

where $F_1$ and $F_2$ denote the two forces of repulsion; here is a segmented design of the repulsion.

The schematic diagram of the extended artificial potential field is shown in Figure 3. “!” denotes the danger zone; “T” denotes the target point.

![Figure 3. Schematic diagram of EAPF.](image-url)
4. Experimentation and Simulation

We utilized MATLAB/Simulink to test the performance of the CISCS algorithm from different angles.

4.1. Formation Controller Simulation

In the following, the effectiveness and reliability of the multi-UAV distributed self-organized collaborative intelligent reconnaissance combat strategy (CISCS) proposed are verified through example solutions. The simulation platform is a kind of multi-UAV simulation platform running in MATLAB R2019a/Simulink, as shown in Figures 4–6.

![Figure 4. Multi-UAV simulation module.](image)

![Figure 5. Stand-alone control model.](image)
In this simulation platform, the UAV body model consists of a UAV power model, a control efficiency model, and a rigid body model, and the UAV controller consists of a position controller and an attitude controller, both of which are PID controllers, and the input of the controller is the desired position, and the output is the PWM control command of the UAV motor, which is inputted into the UAV body model to control the UAV position attitude, and the UAV position attitude is fed back to the controller.

Formation control analysis: in order to conduct formation control performance analysis, we compare the two control methods of adding a formation controller and not adding a formation controller and set up comparison experiments of dynamic flight formation and static flight formation to verify the rapidity and stability of the formation controller. It can be seen that the formation controller can help multi-UAVs form stable flight formations faster.

In Table 2, we give the list of parameters, the selection of which relies on several experiments to obtain the best possible performance optimization.

Table 2. Parameter Setting.

<table>
<thead>
<tr>
<th>Parameter Meaning</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finite time parameter</td>
<td>$\tau_w$</td>
<td>20</td>
</tr>
<tr>
<td>Finite time gain</td>
<td>$w$</td>
<td>1.0</td>
</tr>
<tr>
<td>Learning rate</td>
<td>$\alpha$</td>
<td>0.2</td>
</tr>
<tr>
<td>Discount</td>
<td>$\gamma$</td>
<td>0.9</td>
</tr>
<tr>
<td>Lost rate</td>
<td>$\tau$</td>
<td>0.4</td>
</tr>
<tr>
<td>Gravitational coefficient</td>
<td>$a_1$</td>
<td>9</td>
</tr>
<tr>
<td>Repulsion coefficient</td>
<td>$\beta_1$</td>
<td>13</td>
</tr>
<tr>
<td>Arrival distance</td>
<td>$d_a$</td>
<td>0.1</td>
</tr>
<tr>
<td>Distance threshold</td>
<td>$d_1$</td>
<td>0.5</td>
</tr>
<tr>
<td>repulsive factor</td>
<td>$\gamma_1$</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Firstly, a formation simulation experiment is carried out using one Leader and two Followers; the position of each UAV in 3D space is shown in Table 3; multiple drones are placed in horizontal rows on the ground with a spacing of 1 m. The triangular formation Follower forms a 45° angle with the Leader at a distance of $\sqrt{2}$ m. The circular formation Follower forms an enclosing circle around the Leader with a radius of 2 m. The UAVs in the triangular shape are then simulated in a circular shape. After simulation, it is obtained, as shown in Figures 7–10.
Table 3. Initial position of the drone.

<table>
<thead>
<tr>
<th>UAV</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAV1</td>
<td>(0, 0, 0)</td>
</tr>
<tr>
<td>UAV2</td>
<td>(0, 1, 0)</td>
</tr>
<tr>
<td>UAV3</td>
<td>(0, −1, 0)</td>
</tr>
<tr>
<td>UAV4</td>
<td>(0, 2, 0)</td>
</tr>
<tr>
<td>UAV5</td>
<td>(0, −2, 0)</td>
</tr>
</tbody>
</table>

Figure 7. (a) A 3D view of the non-formation controller. (b) Distance between UAVs without formation controller. Formation time = 15.84 s.

Figure 8. (a) A 3D view with formation controller. (b) Distance between UAVs with formation controllers. Formation time = 12.3320 s.
Figure 9. (a) A 3D view of the formation-free controller. (b) Distance between UAVs without formation controller.

Figure 10. (a) A 3D view with formation controller. (b) Distance between UAVs with formation controllers. Formation time = 16.3720 s.

Extracting the simulation data, it can be seen that the formation time with the formation controller is significantly shorter than the formation time without the formation controller. Without changing the UAV's own controller parameter, the formation controller can help the UAV form a triangular static formation faster, compared with the formation without a formation controller, to accelerate the formation by 3.5080 s. In Figure 8b, the UAV spacing is jittered due to the parameter adjustment problem of the formation controller, but the maximum range of the jittering is ±18.79% of the control amount, and the maximum value of the UAV spacing without the formation controller is 2.7663 m, which is 95.64% of over-regulation. It can be concluded that in static three-aircraft formation, the formation controller can form a faster and more stable formation.

In Figure 9a, without adding the formation controller, the two Follower-drones’ paths cannot be overlapped, and in Figure 9b, it can be seen that the distance control of multiple UAVs is unstable when they are in a circular formation around the Leader; after adding the UAV controller, the Follower-drones’ paths in Figure 10a are basically overlapped, and the UAV distance in Figure 10b can be stabilized around 2 m, and the overshooting is 95.64%. Stabilization can be stabilized near 2 m, and the formation time is 16.3720 s, which is larger than the formation time of 15.84 s for static formation, indicating that the formation of dynamic formation is more difficult. Under the condition of not changing the controller parameter of the UAV itself, in the three-aircraft dynamic formation, the formation effect without a formation controller is poor, and the UAV spacing...
jerks violently; after the addition of the formation controller, there is a small fluctuation in the UAV spacing, but the overall tendency is stable, and the overshooting amount is 27.71%. Therefore, in the three-aircraft dynamic formation, the formation control can help the UAV form a more stable formation.

To verify the scalable performance of this formation controller, formation simulation experiments with five machines are added.

In the five-aircraft static formation control experiment, the comparison between Figures 11a and 12a is obvious; in the presence of the formation controller, the UAV formation is faster and with less error, which accelerates the formation by 3.992 s compared to the formation without the formation controller. In Figure 11b, the formation can be stabilized, but it is obvious that the overshoot is too large compared to Figure 12b, and the formation time is much longer. Although there is some jitter in Figure 12b, it is due to the UAV airframe model, and the range of jitter is small.

![Figure 11](image1.png)  
**Figure 11.** (a) A 3D view without formation controller. (b) Distance between UAVs without formation controller. Formation time = 17.5360 s.

![Figure 12](image2.png)  
**Figure 12.** (a) A 3D view with formation controller. (b) Distance between UAVs with formation controllers. Formation time = 13.5440 s.

In the five-aircraft dynamic formation experiment, there is a large formation error in the Follower-drones in Figure 13a, which may be due to the difficulty of stabilizing the spacing of multiple UAVs during the formation process, which can also be obtained from Figure 13b; after adding the UAV controller, the Follower-drones in Figure 14a are able to form more stable circular formations around Leader, Figure 14b. A stable circular formation and the drone spacing in Figure 14 are obviously much more stable than those in
Figure 13, indicating that the formation controller can help the multi-drones form a better formation.

![Figure 13](image)

**Figure 13.** (a) A 3D view without formation controller. (b) Distance between UAVs without formation controller.

![Figure 14](image)

**Figure 14.** (a) A 3D view with formation controller. (b) Distance between UAVs with formation controllers.

### 4.2 Multi-UAV Cooperative Mission Simulation

In order to verify the effectiveness of CISCS, we call one, three, and five airplanes to perform the same mission and carry out simulation experiments according to the flow of takeoff, formation, formation flight, mission execution, and away from the mission area, and obtain the multi-aircraft mission sequences by CISCS, as shown in Figures 15–18.

Task point coordinates are shown in Figure 15 and Table 4.
Table 4. Task point information.

<table>
<thead>
<tr>
<th>Number</th>
<th>Location</th>
<th>Number</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(100, 50, −1.2)</td>
<td>6</td>
<td>(175, 0, −0.9)</td>
</tr>
<tr>
<td>2</td>
<td>(200, 50, −0.8)</td>
<td>7</td>
<td>(125, 0, −1.5)</td>
</tr>
<tr>
<td>3</td>
<td>(150, 25, −1.2)</td>
<td>8</td>
<td>(100, −50, −0.7)</td>
</tr>
<tr>
<td>4</td>
<td>(150, 0, −0.7)</td>
<td>9</td>
<td>(150, −25, −1.0)</td>
</tr>
<tr>
<td>5</td>
<td>(200, 0, −0.9)</td>
<td>10</td>
<td>(200, −50, −1.1)</td>
</tr>
</tbody>
</table>

Figure 15. Map of take-off points and mission points.

Figure 16. Three-dimensional trajectory of a stand-alone mission flight.
Figure 17. Three-dimensional trajectory of a three-engine mission flight.

Figure 18. Three-dimensional trajectory of a five-engine mission flight.

The mission execution sequence and mission completion time obtained from CISCs are shown in Table 5. From this, it can be seen that there is a reduction of 52.55% and 58.27% for three and five airplanes, respectively, as compared to a single airplane.

Table 5. Sequence of tasks and completion dates.

<table>
<thead>
<tr>
<th>Number of UAV</th>
<th>Mission Sequence</th>
<th>Time of Mission</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T1 → T2 → T5 → T10 → T8 → T7 → T3 → T6 → T9 → T4</td>
<td>223.584 s</td>
</tr>
<tr>
<td>3</td>
<td>T7 → T4 → T6 → T5; T8 → T9 → T10; T1 → T3</td>
<td>106.096 s</td>
</tr>
<tr>
<td>5</td>
<td>T3 → T4; T2 → T5; T7 → T9; T1 → T6; T8 → T10</td>
<td>93.312 s</td>
</tr>
</tbody>
</table>
5. Conclusions

We propose a multi-UAV-distributed self-organized collaborative intelligent reconnaissance combat strategy. In this strategy, a multi-UAV formation controller with time constraints is designed for fast formation, and EAPF is utilized to prevent collisions within the multi-UAV formation, while a distributed improved ant colony algorithm based on Q-Learning is utilized to generate a sequence of tasks executed by multi-UAVs. Simulation experiments show that the multi-UAV finite-time formation controller can help multi-UAVs form stable formations faster and significantly improve the efficiency of task execution by assigning multiple task points to multiple UAVs through the task assignment algorithm, and prevent internal collisions within the UAV formation to guarantee the air safety of UAVs through the Extended Artificial Potential Field (EAPF) method.

CISCs assumes that UAVs are equipped with enough attack munitions and reconnaissance equipment to reconnaissance and attack arbitrary targets, and in the next research, the actual reconnaissance and combat capabilities of UAVs should be considered, increased with dynamic constraints, and multi-dimensional multi-platform collaborative combat issues, such as multi-kinds of UAVs, and ground equipment should be considered.

Author Contributions: Conceptualization, Z.L. and Q.L.; methodology, Z.L. and Q.L.; formal analysis, Z.L. and Q.L.; investigation, Z.L. and G.F.; writing—original draft preparation, Z.L. and Q.L.; writing—review and editing, Z.L., Q.L. and G.F.; supervision, Q.L. and G.F.; project administration, Q.L. and G.F. All authors have read and agreed to the published version of the manuscript.

Funding: Supported by The National Natural Science Foundation of China (Grant No. 61971048), the National Key Research and Development Program of China (No. 2020YFC1511702, 2022YFF0607400), and the Beijing Scholars Program.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References


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.