drones

Article

A Distributed Collaborative Allocation Method of Reconnaissance and Strike Tasks for Heterogeneous UAVs

Hanqiang Deng, Jian Huang *, Quan Liu, Tuo Zhao, Cong Zhou and Jialong Gao

College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China
* Correspondence: nudtjHuang@hotmail.com

Abstract: Unmanned aerial vehicles (UAVs) are becoming more and more widely used in battlefield reconnaissance and target strikes because of their high cost-effectiveness, but task planning for large-scale UAV swarms is a problem that needs to be solved. To solve the high-risk problem caused by incomplete information for the combat area and the potential coordination between targets when a heterogeneous UAV swarm performs reconnaissance and strike missions, this paper proposes a distributed task-allocation algorithm. The method prioritizes tasks by evaluating the swarm's capability superiority to tasks to reduce the search space, uses the time coordination mechanism and deterrent maneuver strategy to reduce the risk of reconnaissance missions, and uses the distributed negotiation mechanism to allocate reconnaissance tasks and coordinated strike tasks. The simulation results under the distributed framework verify the effectiveness of the distributed negotiation mechanism, and the comparative experiments under different strategies show that the time coordination mechanism and the deterrent maneuver strategy can effectively reduce the mission risk when the target is unknown. The comparison with the centralized global optimization algorithm verifies the efficiency and effectiveness of the proposed method when applied to large-scale UAV swarms. Since the distributed negotiation task-allocation architecture avoids dependence on the highly reliable network and the central node, it can further improve the reliability and scalability of the swarm, and make it applicable to more complex combat environments.

Keywords: heterogeneous UAV swarm; reconnaissance and strike; distributed negotiate; time coordination; deterrent maneuver

check for updates

Citation: Deng, H.; Huang, J.; Liu, Q.; Zhao, T.; Zhou, C.; Gao, J. A Distributed Collaborative Allocation Method of Reconnaissance and Strike Tasks for Heterogeneous UAVs. *Drones* **2023**, *7*, 138. https://doi.org/10.3390/drones7020138

Academic Editor: Oleg Yakimenko

Received: 13 January 2023
Revised: 10 February 2023
Accepted: 14 February 2023
Published: 15 February 2023

1. Introduction

The popularity of UAVs in civil aerial photography, agriculture, surveillance, and mapping [1] has made people see its application prospects in more fields. As a low-cost, low-risk, and cost-effective weapon or carrier, UAVs have frequently appeared on the battlefield. It has become the focus of researchers to endow decentralized, heterogeneous, and low-cost UAV swarms with autonomous coordination capabilities to complete more complex tasks, because this is an important way to improve the flexibility and reliability of small UAV swarms to perform combat tasks [2,3].

Since the battlefield is a highly confrontational environment, distributed collaboration architecture is an important way to achieve large-scale swarm collaboration [4]. The centralized architecture that has been widely researched and applied has the advantage of a simpler algorithm design, but it also has the problem of high requirements on the network and central computing nodes. In contrast, each UAV node in a distributed architecture communicates and cooperates with other UAVs as an independent entity. Since there are no critical nodes in the network, the architecture is highly scalable and reliable.

According to the analysis of relevant researchers, the commonly used methods of distributed task collaborative assignment can be divided into heuristic optimization algorithms [5,6], market-based methods [7–10] and alliance-based methods [2,11] and so on [12].

Heuristic optimization algorithms are widely used because they do not require gradient information and do not rely on problem models with good mathematical properties. For example, in the literature [13], an improved pigeon-inspired optimization algorithm is proposed to solve the optimization problem of cooperative target searches, while it adopts a centralized control architecture. For multi-UAV cooperative execution of reconnaissance missions, ref. [5] proposed an intelligent self-organized algorithm (ISOA) mission-planning method. UAVs exchange status and planning information with each other, and locally optimize route planning using the improved distributed ant colony algorithm to update route planning, and repeat the process until the task is completed. However, the article assumes that all UAVs are homogeneous and that the targets are find-and-destroy elements.Another paper [14] implements a distributed task assignment method for UAV swarm reconnaissance missions based on the wolf pack algorithm, including a cooperative search algorithm based on wolf reconnaissance behavior and a cooperative attack task assignment method after the target is discovered. The algorithm has good scalability, but it does not consider the risk of searching the unknown environment when optimizing the scheme.

The market-based method is one in which the bidders estimate the benefits of completing different tasks, broadcast the bids to each other, and win with the best one, and the bidders re-evaluate after the environment or allocation plan is updated until there is no conflict. Aiming at the task assignment problem of heterogeneous cooperative UAV, a paper [7] proposed a task assignment algorithm based on improved CBGA (improved consensus-based grouping algorithm, derived from CBBA [15]). The algorithm has a simple structure, but less consideration is given to factors such as the cooperative relationship between UAVs. To deal with real-time task allocation in resource-constrained wireless-sensor networks, the authors of [16] proposed a reverse auction-based scheme using an adaptive algorithm for each node (bidder) to locally calculate its best bid response with a non-smooth and concave payoff function.

The formation of the alliance divides the large-scale UAV swarm into several small UAV alliances through strategies such as cooperative games [11]. This architecture first distributes tasks among the alliances, and then redistributes the received tasks within the alliance to effectively reduce the dimension of the problem. Authors [2] use a layered extended contract network protocol to realize the collaborative control of UAV swarms, which has the advantage of solving speed when the swarm scale is large. However, this literature ignores the influence of the division method of UAV subsets on the effect of swarm behavior. For example, two UAVs that should have cooperated are divided into different alliances, resulting in a decrease in the quality of the solution.

Researchers have also tried to combine the advantages of different architectures. When the problem has complex constraints, it is difficult to converge to a good result by directly applying CBBA and other methods, and repeated negotiations will cause high communication costs. Therefore, some researches combine heuristic algorithms with market-based methods. For example, the paper [17] considers task-time constraints and obstacle constraints, uses intelligent optimization algorithms locally to optimize the scheme, and then negotiates with other UAVs. Similarly, ref. [18] regards the minimum distance sum and the minimum maximum completion time as the optimization goals, and first uses the genetic algorithm (GA) to locally optimize, and then uses the CBAA-derived algorithm to reach a consensus among nodes.

In terms of the factors concerned in the research of UAV swarm mission collaboration, the factors considered mainly include UAV maneuvering distance [8,19], area coverage [5,15], route planning [5,20], avoidance of no-fly zones [2], etc., while the threat of cooperation between enemy platforms is rarely considered.

Aiming at the scenario where there may be a potential cooperative relationship between enemy targets, this paper proposes a distributed collaborative optimization method for heterogeneous UAVs based on a negotiation mechanism and GA.

The main contributions of this paper include the following aspects:

- The priority of tasks is evaluated by the swarm's capability superiority over the tasks to reduce the search space. The capability superiority is represented by the spatial density and the capability availability of the tasks, and the attention mechanism is combined to suppress the distant tasks to evaluate the task priority;
- The time coordination mechanism and deterrent maneuver strategy is used to reduce the risk of reconnaissance missions. Due to the incomplete information of the task, multiple UAVs are used to reconnaissance the dense tasks synchronously, and the UAVs with strike capabilities are deployed with deterrent maneuver strategy to reduce the risk of reconnaissance missions;
- A distributed task-assignment negotiation mechanism is designed so that UAVs can run in a completely distributed manner. Compared with the centralized GA, the proposed method can reduce the problem search space, improve the optimization speed and the quality of the solution, and the distributed framework can also improve the scalability and reliability of the swarm.

The remainder of this paper is organized as follows: the problem is defined and described in Section 2. The distributed collaborative allocation method for heterogeneous UAVs is described in Section 3. In Section 4, a distributed simulation environment is built, and the proposed method is verified in this environment. Finally, we conclude the paper in Section 5.

2. Problem Description

Assuming that there are several suspicious areas on the battlefield, a heterogeneous UAV swarm with different reconnaissance and strike capabilities needs to be dispatched to perform the reconnaissance and strike mission, and the UAV nodes communicate with each other through a multi-hop ad hoc network. UAVs autonomously negotiate task-allocation schemes for reconnaissance and strike targets. Since there may be a synergistic relationship between enemy targets, the mission risk and mission completion time should be minimized during mission execution.

The problem can be formalized as the problem of N_U UAVs $U = \{u_i | i = 1, 2, \cdots, N_U\}$ completing N_T tasks $T = \left\{t_j | j = 1, 2, \cdots, N_T\right\}$.

The state of UAV u_i is denoted as $u_i = \left\langle p_i, v_i, \boldsymbol{a}_i, T_i^p, T_i^b, \bar{T}_i^b, U_i \right\rangle$ where p_i is the current position; v_i is the maximum flight speed; $\boldsymbol{a}_i$ is the load capacity matrix of u_i, as shown in Table 1, the capacity between loads can be added but a single load cannot be split; T_i^p is the task set that is perceived but has not decided the assignment; T_i^b and $\bar{T}_i^b$ are the task queue that u_i will participate in and the task set that will not participate; U_i is the UAV swarm status perceived by the u_i, which can be updated through communication with other UAVs.

Table 1. Example of payload capacity of u_i.

Payload Type	Scout Speed	Penetration Ability	Damage Ability	Reusable
Scout payload	50	0	0	Y
Strike payload 1	0	40	60	N
Strike payload 2	0	80	40	N

The state of the task can be expressed as $t_j = \langle p_j, s_j, \boldsymbol{a}_j \rangle$, where p_j is the position of the task, s_j is the area of the suspicious area where the task is located, and $\boldsymbol{a}_j$ is the strike capability required by the task. In the process of reconnaissance of suspicious areas by UAVs with reconnaissance capabilities, existing targets can be found and $\boldsymbol{a}_j$ can be obtained; but when there is no target in the area, this conclusion can only be drawn after the UAV has scouted the entire area, in this case $\boldsymbol{a}_j = \mathbf{0}$.

The connection relationship between UAVs is expressed as an adjacency matrix $\boldsymbol{L} = [l_{im}]_{i,m \in [1, N_U]}$, and there is $l_{im} = 1$ when distance $d_{im} \leq d_\delta$, otherwise $l_{im} = 0$, and d_δ is the maximum distance for single-hop communication. When the reconnaissance node completes the reconnaissance task, it broadcasts the reconnaissance result (that is, whether there is a target in the area and the required strike capability) to the swarm by using the ad hoc network. Each UAV utilizes the perceived task status and the status of other UAVs to optimize the distribution of reconnaissance and strike tasks by negotiating with neighboring UAVs.

3. The Proposed Method

To solve the above problems, this paper proposes a distributed collaborative allocation method of reconnaissance and strike tasks for heterogeneous UAVs, and its framework is shown in Figure 1.

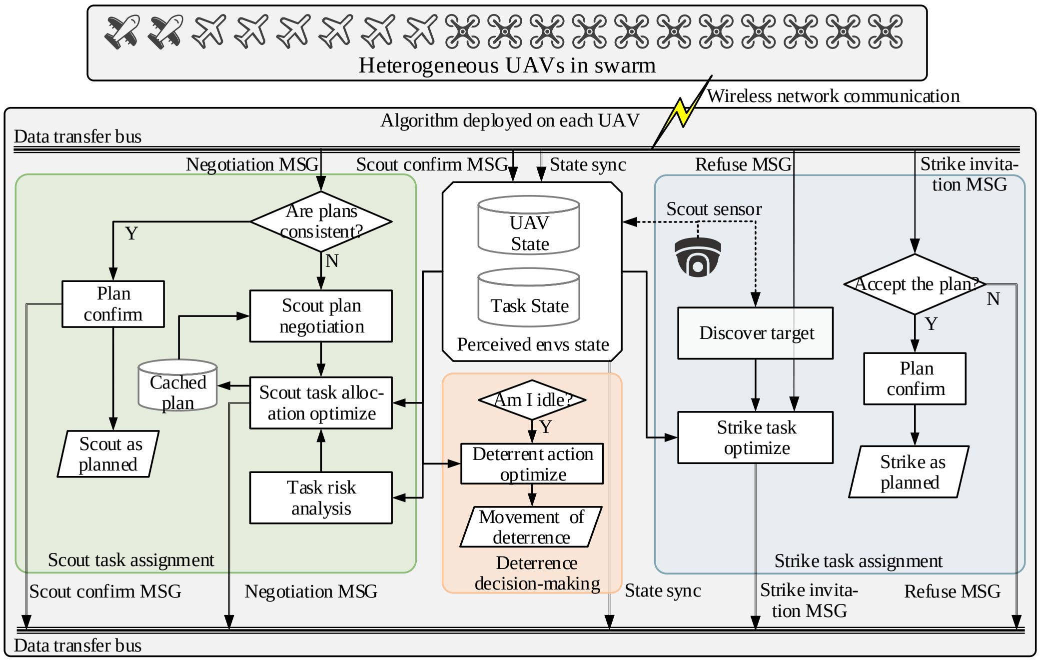

Figure 1. The framework of the collaborative allocation algorithm for reconnaissance and strike tasks.

This method consists of three main modules: negotiate for scout task assignment, strike task assignment, and deterrence decision-making. When negotiating scout tasks, this method first evaluates the tasks risk according to the degree of superiority of the UAVs over the enemy, and assigns tasks with the goal of minimizing the degree of task risk and the task completion time. Based on the perceived environmental information and the historical status information obtained by communicating with neighbor nodes, each UAV uses the GA to generate a local task-allocation and time-coordination plan after analyzing the priority of each task, and negotiates with neighbors to resolve conflicts. After the reconnaissance node discovers the enemy target, it will optimize the strike plan locally and request the relevant nodes to coordinate execution. If the request is rejected, it will re-optimize the strike plan until the strike mission is successfully assigned. When the nodes with strike capability are idle, they will fly to the reconnaissance nodes with weak strike capability to enhance their deterrence against the enemy and shorten the time from target discovery to striking.

3.1. Negotiate for Scout Task Assignment

When a large number of reconnaissance tasks need to be allocated, this paper first selects the reconnaissance tasks that should be completed first based on heuristic rules, and then uses the local optimization and distributed negotiation mechanism to allocate the tasks.

3.1.1. Heuristic Rules

To reduce the risk of UAVs executing reconnaissance missions, task allocation shall be based on the following rules:

- Give priority to the tasks that are isolated and in weak areas of the enemy;
- Give priority to the tasks where our strike capability is dominant;
- Give priority to nearby tasks.

Based on these rules, the priority evaluation method of tasks is defined as follows:

Definition 1. *S-Sig function. To make each UAV pay more attention to the local environment, referring to the sigmoid function, function* $f_{ssig}(x)$ *is defined as:*

$$f_{ssig}(x) = \frac{1}{1 + \exp(4x - 4)} \tag{1}$$

When $0 < x < 0.5$*,* $f_{ssig}(x)$ *decays slowly. The decay speed increases with the increase of* x *and reaches the maximum at* $x = 1$*. When* $x > 1$*, its decay speed decreases and* $\lim_{x \to +\propto} f_{ssig}(x) = 0$*. In task-priority evaluation, this function can be used to smoothly suppress the priority of tasks that are far away, while the priority of nodes that are close to the reference node is almost unaffected by distance.*

Definition 2. *Spatial density of tasks. For the convenience of analysis, the typical influence radius of a single UAV is set to* φ *according to the cruising speed and combat radius of the UAV. Referring to the concept of kernel density estimation (KDE) in literature [21], we make the mutual influence between targets attenuate with the increase of distance, and assume that the probability of mutual cooperation between two targets within radius* φ *is large. Therefore, Equation (1) is used as the kernel function to calculate the task space density, and for any task* $t_j \in T$*, its space density* ρ_j *is defined as:*

$$\rho_j = \sum_{t_n \in T, j \neq k} f_{ssig}(d_{jn}/\varphi) \tag{2}$$

where d_{jn} *is the Euclidean distance between task* t_j *and* t_n*. It can be inferred that* ρ_j *focuses on the radius within* 2φ*, because when* $d_{jn} > 2\varphi$*,* $f_{ssig}(d_{jn}/\varphi) < 0.017$*.*

Definition 3. *Capability availability estimation of UAV. UAV* u_i *estimates the capability availability* $f_{im}^{\mathbb{C}}(\tau)$ *of neighboring UAV* u_m *according to its internal perception state at time* τ*, which can be expressed as:*

$$f_{im}^{\mathbb{C}}(\tau) = \prod_{c \in \mathbb{C}} f_{imc}(\tau)^{1/|\mathbb{C}|} \tag{3}$$

$$f_{imc}(\tau) = \sum_{u_k \in U_i} f_{ssig}(d_{mk}/\varphi)\lambda^{\tau - \tau_{ik}} \varsigma_{kc} \tag{4}$$

where $\mathbb{C}$ *is the set of capability types involved in the problem;* $f_{imc}(\tau)$ *is the availability of capabilities of type c;* U_i *is the collection of UAVs perceived by* u_i*;* d_{mk} *is the distance between* u_m *and* u_k*;* τ_{ik} *is the timestamp when* u_i *receives the status of* u_k*;* λ *is the coefficient that the weight of the information from the neighboring UAV decays with time, that is, the longer the status of a UAV is updated, the lower the weight.* ς_{kc} *is the capability value of type c possessed by* u_k*.*

Definition 4. *Capability availability estimation of task. Similar to Definition 3, UAV* u_i *estimates the capability availability* $f_{ij}^{\mathbb{C}}(\tau)$ *of task* t_j *according to its internal perception state at time* τ*, which can be expressed as:*

$$f_{ij}^{\mathbb{C}}(\tau) = \prod_{c \in \mathbb{C}} f_{ijc}(\tau)^{1/|\mathbb{C}|} \tag{5}$$

$$f_{ijc}(\tau) = \sum_{u_k \in \mathbf{U}_i} f_{ssig}\left(d_{jk}/\varphi\right) \lambda^{\tau - \tau_{ik}} \varsigma_{kc} \tag{6}$$

Definition 5. *Task prioritization assessment. Define $\eta_{imj}(\tau)$ as the priority of task t_j to u_m that is evaluated by u_i, and then $\eta_{imj}(\tau)$ can be expressed by the capability coverage of u_m at t_j, namely:*

$$\eta_{imj}(\tau) = \frac{1}{\rho_j} f_{im}^{\mathbb{C}}(\tau) \cdot f_{ssig}\left(\frac{d_{mj} + \alpha_1 \cdot \max\left(0, d_{ij} - \varphi\right)}{\varphi}\right) \tag{7}$$

the function $\max(\cdot)$ *means to take the maximum value, and α_1 is the weighting coefficient of the extra distance. $d_{mj} + \alpha_1 \cdot \max\left(0, d_{ij} - \varphi\right)$ indicates that the distance between u_i and t_j should also be considered when evaluating the capability coverage of u_m at t_j, and the farther the distance is, the greater the priority of task t_j is suppressed.*

It can be inferred from the definition of Equation (7) that the closer the task t_j is to u_i and u_m, the lower its spatial density, and the more sufficient the UAV capability that can cover it, the higher priority u_i thinks u_m will gives to t_j. This formula can be used for u_i to measure the superiority of our UAVs to different tasks, which is consistent with the heuristic rules.

3.1.2. Collaborative Optimization of Reconnaissance Tasks Assignment

When assigning the given reconnaissance and strike tasks, the algorithm should strive to maximize the proportion of task completion, minimize the total task completion time, and minimize the degree of task risk. Therefore, the objective function of reconnaissance task-allocation is defined as:

$$\max_{\beta^{sc}} J(\beta^{sc}) \tag{8}$$

where

$$J(\beta^{sc}) = \sum_{\left(u_i, t_j, \tau_j^s, g_j\right) \in \beta^{sc}} g_j(\beta^{sc}) \tag{9}$$

$$g_j(\beta^{sc}) = r_j(\beta^{sc}) \cdot e^{-\alpha_r \left(\tau_j^s - \tau\right)} \tag{10}$$

$$r_j(\beta^{sc}) = r_{\max} - \alpha_t \cdot \sum_{(u_m, t_n, \tau_n^s, g_n) \in \beta^{sc}} \left[\!\left[\tau_j^s < \tau_n^s\right]\!\right] \cdot f_{ssig}\left(d_{jn}/\varphi\right) \tag{11}$$

where the quaternion $\left(u_i, t_j, \tau_j^s, g_j\right)$ indicates that u_i will start reconnaissance tasks t_j at time τ_j^s, and the expected benefit is g_j; α_r is the coefficient of time for the discount of rewards; reconnaissance task plan β^{sc} is a collection of task assignment quaternions; τ_j^s and τ_n^s are the start execution times of tasks t_j and t_n, respectively; $r_j(\beta^{sc})$ is the expected reward for β^{sc} to complete t_j, which is composed of the maximum reward $r_{\max}$ and the estimated risk for completing the task, and α_t is the weight of risk; $[\![P]\!] = \begin{cases} 1 & \text{If } P \text{ is true} \\ 0 & \text{Otherwise} \end{cases}$ is an Iverson bracket, and if the start time τ_n^s is later than τ_j^s in the scheme, t_n will pose a threat to t_j.

It can be inferred from Equation (9) that if adjacent tasks can be scouted at the same time, the threat to each other can be reduced and the reward can be increased. However, starting reconnaissance at the same time also means that some tasks need to be deliberately postponed, leading to a decline in overall reward. Therefore, it is necessary to optimize the time synergy of each plan.

(1) Time-collaborative optimization of plan

The time-collaborative optimization of a plan is to optimize the specific start time of each assigned reconnaissance task. For any two assignments $\left(u_i, t_j, \tau_j^s, g_j\right)$ and $(u_m, t_n, \tau_n^s, g_n)$ in plan β^{sc}, if $\tau_j^s < \tau_n^s$, u_i can postpone its task start time to the same as u_m to improve its task reward $r_j(\beta^{sc})$. Express the updated plan as $\beta^{sc\prime}$, then there is $\tau_j^{s\prime} = \tau_n^s$. Let $\Delta\tau_{jn}^s = \tau_n^s - \tau_j^s$, then the gain of reward for time collaboration is:

$$\begin{aligned}
\Delta g_{jn} &= g_j(\beta^{sc\prime}) - g_j(\beta^{sc}) = r_j(\beta^{sc\prime}) \cdot e^{-\alpha_r(\tau_n^s - \tau)} - r_j(\beta^{sc}) \cdot e^{-\alpha_r\left(\tau_j^s - \tau\right)} \\
&= e^{-\alpha_r\left(\tau_j^s - \tau\right)}\left[r_j(\beta^{sc\prime})e^{-\alpha_r \cdot \Delta\tau_{jn}^s} - r_j(\beta^{sc})\right] \\
&= e^{-\alpha_r\left(\tau_j^s - \tau\right)}\left[\left(r_j(\beta^{sc}) + \alpha_t \cdot f_{ssig}(d_{jn}/\varphi)\right)e^{-\alpha_r \cdot \Delta\tau_{jn}^s} - r_j(\beta^{sc})\right] \\
&= e^{-\alpha_r\left(\tau_j^s - \tau\right)}\left[r_j(\beta^{sc})\left(e^{-\alpha_r \cdot \Delta\tau_{jn}^s} - 1\right) + \alpha_t \cdot f_{ssig}(d_{jn}/\varphi)e^{-\alpha_r \cdot \Delta\tau_{jn}^s}\right]
\end{aligned} \tag{12}$$

Since the time collaboration between any two assignments in a plan will depend on the recalculation of Equation (8), and the time collaborative optimization is an underlying algorithm that will be called repeatedly, we define Algorithm 1 based on Equation (12) to quickly optimize the time collaboration of the given plan.

Algorithm 1 Fast time-collaborative optimization

Input: The plan β^{sc} that needs to optimize its time collaboration
Output: The updated plan $\beta^{sc\prime}$.

```
while True do
    Δg_{j*n*} ← max_{(u_i,t_j,τ_j^s,g_j)∈β^sc, (u_m,t_n,τ_n^s,g_n)∈β^sc, τ_j^s<τ_n^s} (Δg_{jn})    ▷ Find the best time collaboration pair using Equation (12)
    if Δg_{j*n*} > ε_g then
        τ_{j*}^s ← τ_{n*}^s    ▷ Time collaboration when the gain of reward meets the threshold
    else
        return the updated β^sc
    end if
end while
```

(2) Optimization of task-assignment plan

The negotiation algorithm for reconnaissance tasks assignment consists of two parts: (i) optimizing the assignment scheme under specified conditions; (ii) negotiating with other nodes for conflict resolution.

The algorithm for optimizing the allocation plan under specified conditions is shown as Algorithm 2. The optional input $\overset{\leftrightarrow}{\beta}^{sc} = \left\{\left(\overset{\leftrightarrow}{u}_i, \overset{\leftrightarrow}{t}_i\right) | i = 1, 2, \cdots, \overset{\leftrightarrow}{N}^{sc}\right\}$ is the specified partial of the task-allocation plan, where $\left(\overset{\leftrightarrow}{u}_i, \overset{\leftrightarrow}{t}_i\right)$ indicates that task $\overset{\leftrightarrow}{t}_i$ is assigned to $\overset{\leftrightarrow}{u}_i$, and $\overset{\leftrightarrow}{N}^{sc}$ is the number of assigned pairs. The optional input $\langle U^{sc}, T^{sc}\rangle$ is the set of scout UAVs U^{sc} and the set of tasks T^{sc} that need to be optimized for allocation.

If $\langle U^{sc}, T^{sc}\rangle$ is not given, the algorithm will automatically select the set of scout UAVs within n_c hops to u_i as U^{sc}, and select tasks according to the priority $\eta_{imj}(\tau)$ of each task t_j. This strategy meets the heuristic rules described in Section 3.1.1, and can reduce the optimization search space while preserving the high-quality solution space.

Algorithm 2 Scout plan optimization within UAV u_i

Input: $\overleftrightarrow{\beta^{sc}}$: Part of the task-assignment scheme that has been specified
$\langle U^{sc}, T^{sc} \rangle$: The set of scout UAVs and tasks that need to be optimized
Output: The optimized plan β^{sc}.
if $\langle U^{sc}, T^{sc} \rangle$ is not given **then** ▷ Automatically select scout UAVs and tasks
  $U^{sc} \leftarrow$ the set of scout UAVs within n_c hops to u_i
  Init T_0 as an empty list
  while $|T_{sc}| < N^{sc\,\max}$ **do** ▷ Iteratively add tasks according to the evaluated priority
    for u_m in U^{sc} **do**
      if the latest task added to T_0 for u_m is duplicated with existing tasks **then**
        $T_0 \leftarrow$ add the next preferred task of u_m according to $\eta_{imj}(\tau)$ to T_0
      end if
    end for
    $T^{sc} \leftarrow \left\{ t_j | t_j \in T_0, (t_j, \cdot) \notin \overleftrightarrow{\beta^{sc}} \right\}$ ▷ Remove duplicate tasks and tasks in $\overleftrightarrow{\beta^{sc}}$
    if no task is added to T_0 **then**
      Break
    end if
  end while
end if

function $f_\beta(\beta^{sc})$
  $\beta^{sc} \leftarrow \beta^{sc} \cup \overleftrightarrow{\beta^{sc}}$ ▷ Merge the specified and generated plans
  Calculate τ_j^s of each assignment $\left(u_i, t_j, \tau_j^s, g_j\right) \in \beta^{sc}$ with the predicted previous task finish time and the travel time between u_i and t_j
  Optimize the time collaborate of β^{sc} with Algorithm 1
  return $J(\beta^{sc})$ ▷ Calculate the fitness of β^{sc} with Equation (9)
end function

$\beta^{sc^*} \leftarrow$ Using GA to optimize the assignment of tasks in T^{sc} to U^{sc} with the goal to maximize $f_\beta(\beta^{sc})$
return the best plan β^{sc^*}

After determining $\langle U^{sc}, T^{sc} \rangle$, the algorithm optimizes the assignment of the tasks based on GA, and its optimization goal is to maximize its fitness function $f_\beta(\beta^{sc})$. In $f_\beta(\beta^{sc})$, it first merges plan β^{sc} with the specified part of plan $\overleftrightarrow{\beta^{sc}}$; Then the start time of the task is estimated according to the predicted finish time of the preceding task of the UAV in each assignment and the travel time from the UAV to the corresponding task. Then, Algorithm 1 is used to quickly optimize the time collaboration between the assignments. Finally, the fitness is calculated according to the objective function defined by Equation (9).

When Algorithm 2 is used for conflict resolution optimization of multiple plans, the non conflict part of the plan can be regarded as $\overleftrightarrow{\beta^{sc}}$ and the conflict part as $\langle U^{sc}, T^{sc} \rangle$, which can reduce the search space of the optimization problem and improve the optimization speed.

(3) Negotiation-based conflict resolution

After each UAV has generated or updated the best plan β^{sc^*} for scout task allocation, it will send the plan to UAVs within n_c hops. Let $B_i^{sc} = \{\beta_m^{sc} | hop(u_i, u_m) \leqslant n_c\}$ be the set of scout task-allocation plans received by u_i from other UAVs, then the scout task negotiation and conflict resolution algorithm can be expressed as Algorithm 3. The received UAV uses Algorithm 3 to resolve the conflict between its own plan and the received plan to update its plan until there is no conflict between the plans of neighboring nodes.

Algorithm 3 Scout plan conflict-resolving within UAV u_i

Input: β_i^{sc}: The latest local scout plan
B_i^{sc}: The set of received scout plans
Output: The updated local plan β_i^{sc}.

$\beta_i^{sc'} \leftarrow \beta_i^{sc}$
$U_i^{sc} \leftarrow \{u_m | (u_m, t_n, \tau_n^s, g_n) \in \beta_i^{sc}\}$
for β_m^{sc} in B_i^{sc} **do**
    $\overset{\leftrightarrow}{\beta}{}_i^{sc} \leftarrow$ consistent assignments between β_m^{sc} and $\beta_i^{sc'}$. ▷ Lock the consistent part
    $\beta_i^{conflict} \leftarrow \left(\beta_i^{sc'} \cup \beta_m^{sc}\right) - \overset{\leftrightarrow}{\beta}{}_i^{sc}$ ▷ The part of conflict assignments
    $U_i^{conflict} \leftarrow \left\{u_m | (u_m, t_n, \tau_n^s, g_n) \in \beta_i^{conflict}\right\}$ ▷ Extract the UAVs in conflict part
    $T_i^{conflict} \leftarrow \left\{t_n | (u_m, t_n, \tau_n^s, g_n) \in \beta_i^{conflict}\right\}$ ▷ Extract the tasks in conflict part
    $\beta_i^{sc^*} \leftarrow$ Re-optimize using Algorithm 2 with input $\left(\overset{\leftrightarrow}{\beta}{}_i^{sc}, \left\langle U_i^{conflict}, T_i^{conflict}\right\rangle\right)$
    $\beta_i^{sc'} \leftarrow \{(u_m, t_n, \tau_n^s, g_n) | (u_m, t_n, \tau_n^s, g_n) \in \beta_i^{sc^*}, u_m \in U_i^{sc}\}$
    ▷ Only the task assignments of the UAVs belonging to U_i^{sc} are retained
end for
if β_i^{sc} inconsistent with $\beta_i^{sc'}$ **then** ▷ If the task assignment changes
    Broadcast $\beta_i^{sc'}$ ▷ Broadcast the updated plan to neighbors within n_c hops
end if
$\beta_i^{sc} \leftarrow \beta_i^{sc'}$ ▷ Replace the local plan with the new plan
$B_i^{sc} \leftarrow \varnothing$ ▷ Clear the sets of received scout plan
return the updated β_i^{sc}

In Algorithm 3, the received plans are conflict resolved with the local plan one by one. For each task β_m^{sc}, the consistent part between it and $\beta_i^{sc'}$ is locked, the UAVs and tasks involved in the inconsistent part are extracted, and Algorithm 2 is used for re-optimization.

The reason for not merging all the plans at the same time is that the more plans received, the lower the probability of obtaining assignments containing consistent parts, which makes each iteration almost equal a full re-assignment, leading to low convergence efficiency.

3.2. Optimization of Strike Task Allocation

Let T_i^{st} represent the set of targets to strike in u_i, and the capability requirements $\boldsymbol{a}_j$ of each discovered target t_j is known; T_i^{sc} is the set of assigned reconnaissance tasks perceived by u_i; U_i^{st} is the idle UAVs with strike capability within n_c hops to u_i.

Since the targets that need to be struck are discovered dynamically, the scheduling of UAVs with strike capability is not only related to the currently discovered tasks, but also related to the tasks that may be discovered in the future. When optimizing strike capability scheduling, it is necessary to take the nearby reconnaissance tasks into consideration, that is, on the basis of ensuring that the capability requirements of discovered targets can be met, the capability of deterrent reconnaissance tasks should be enhanced as much as possible.

Therefore, the optimization objective of strike task allocation is defined as:

$$\min_{\beta^{st}} J\left(\beta^{st}\right) \tag{13}$$

where

$$\begin{aligned} J\left(\beta^{st}\right) = & \sum_{\left(U_j, t_j, \tau_j^s\right) \in \beta^{st}} \left[\left[\!\left[t_j \in T_i^{st} \right]\!\right] \cdot \left(f^{us}\left(t_j, U_j\right) \cdot \Xi + \alpha_c \cdot \Delta\varsigma\left(t_j, U_j\right) + \alpha_\tau \cdot \tau_{\max}\left(t_j, U_j\right) \right) \right] \\ & - \alpha_{th} \cdot \min_{\substack{\left(U_j, t_j, \tau_j^s\right) \in \beta^{st} \\ t_j \in T_i^{sc}}} \left(\sum_{u_m \in U_j} \varsigma_m \right) \Big/ \left(\frac{1}{\left|U_j\right|} \sum_{u_m \in U_j} \frac{d_{mj}}{\nu_m} \right) \end{aligned} \tag{14}$$

$$f^{us}\left(t_j, U_j\right) = \begin{cases} 0, & \boldsymbol{a}_j \leqslant \sum\limits_{u_m \in U_j} \varsigma_m \\ 1, & else \end{cases} \tag{15}$$

$$\Delta\varsigma\left(t_j, U_j\right) = \sum_{c \in \mathbb{C}} \left(-\boldsymbol{a}_{jc} + \sum_{u_m \in U_j} \varsigma_{mc} \right) \tag{16}$$

$$\tau_{\max}\left(t_j, U_j\right) = \max_{u_m \in U_j} \left(d_{mj} / v_m\right) \tag{17}$$

β^{st} is the strike plan that composed of the strike capability assignment triplet $\left(U_j, t_j, \tau_j^{st}\right)$, and the triplet indicates that the set of UAVs U_j need to arrive and strike t_j at time τ_j^{st}. $f^{us}\left(t_j, U_j\right)$ is used to judge whether the capability requirements of task t_j can be met by the strike plan, and if not, a large constant Ξ will be added to the objective function to make the algorithm give priority to met the capability requirements of t_j. $\Delta\varsigma\left(t_j, U_j\right)$ represent the redundancy value of the strike capability assigned to t_j, and $\tau_{\max}\left(t_j, U_j\right)$ is the latest arrival time of the strike capability assigned to t_j, and these two are minimized by the algorithm on the basis of meeting the capability requirements. α_c and α_τ are weight coefficients. The part weighted by α_{th} is expected to maximize the minimum deterrent degree of reconnaissance tasks.

When a UAV discovers the target during reconnaissance, it triggers Algorithm 4 for strike task allocation, which uses GA to minimize the objective function Equation (13). After optimization, the number of strike loads required for each UAV is calculated in detail, and the invitations are send to the UAVs participating in the strike of the target that discovered by u_i in the plan.

Algorithm 4 Strike plan Optimization within UAV u_i

Input: n_{st}: Strike UAV invitation hops
$U^{\neg st}$: The exclude set of strike UAVs
t_{ui}: The target that discovered by u_i
T_i^{st}: The set of strike targets discovered by other nodes and to be assigned
T_i^{sc}: The set of assigned scout tasks perceived by u_i
Output: Strike plan or the result that failed

```
U_i^th ← Idle strike UAVs within n_st to u_i, and not in U^¬st
T_i^union ← T_i^st ∪ T_i^sc ∪ {t_ui}            ▷ Taking strike and scout tasks into consideration
β^st* ← Using GA to optimize the assignment of U_i^th to T_i^union with the goal of Equation (13)
(U_ui, t_ui, τ_ui^st) ← (U_ui, t_ui, τ_ui^st) ∈ β^st*      ▷ Extract the assignment for t_ui in β^st*
if meets the capacity requirements of t_ui then
    U_ui ← Sort U_ui in ascending according to the distances between UAVs and t_ui
    for u_m in U_ui do                ▷ Calculate the loads that each UAV will contribute in detail
        Occupy strike load of u_m one by one until t_ui is satisfied or all loads are occupied
    end for
    Recalculate the strike time τ_ui^st as the latest arrival time of the occupied UAVs
    Send invitation to u_m ∈ U_ui with the occupied loads and the strike time τ_ui^st
    return (U_ui, t_ui, τ_ui^st)
else if n_st < n_st max then          ▷ Expand the request range for strike UAVs until n_st max
    n_st ← n_st + 1
    Recursive optimize strike plan using Algorithm 4
else
    return Failed                     ▷ There is not enough strike UAVs to execute this task
end if
```

However, if any UAV rejects the invitation, it will be excluded and the scheme will be optimized again until the strike task is successfully assigned.

3.3. Deterrence Maneuver Optimization

To enhance the capability deterrence against potential enemy targets in the reconnaissance area and shorten the time from target detection to strike execution, each idle UAV with strike capability tends to accompany other UAVs on reconnaissance tasks to provide potential capability deterrence.

In the deterrence maneuver optimization, each UAV u_i only considers the UAVs within n_c hops and the corresponding reconnaissance tasks of these UAVs. As there is no specific requirement on arrival time and capability for deterrence, each UAV takes maximizing the minimum task capability coverage as the optimization goal, and decides the destination according to the perceived situation without negotiating with other UAVs. Consistent with the part weighted by α_{th} in Equation (14), the objective function of deterrence maneuver optimization is as Equation (18) shows, and it periodically calls Algorithm 5 to update its deterrence maneuvers.

$$\max_{\beta^{th}} J\left(\beta^{th}\right) \tag{18}$$

where

$$J\left(\beta^{th}\right) = \min_{(U_j,t_j)\in\beta^{th}} \left(\sum_{u_m\in U_j} \varsigma_m\right) \Big/ \left(\frac{1}{|U_j|}\sum_{u_m\in U_j}\frac{d_{mj}}{\nu_m}\right) \tag{19}$$

Algorithm 5 Deterrence maneuver optimization within UAV u_i

1: $U_i^{th} \leftarrow$ Idle strike UAVs within n_c hops to u_i
2: $U_i^{sc} \leftarrow$ Scouting UAVs within n_c hop to u_i
3: $T_i^{sc} \leftarrow$ The tasks being scouted by U_i^{sc}
4: $\beta^{th*} \leftarrow$ Using GA to optimize the assignment of U_i^{th} to T_i^{sc} with the goal of Equation (18).
5: Extract the deterrence tasks of u_i from β^{th*} and maneuver to it.

4. Experiment and Result Analysis

4.1. Experiment Settings

In order to verify the distributed collaborative allocation method of reconnaissance and strike tasks for heterogeneous UAVs proposed in this paper, a simulation environment for heterogeneous UAV reconnaissance and strike tasks is built in Python 3.6 in this section, and its framework is shown in Figure 2. The simulation environment control module runs as an independent thread to support graphical user interface (GUI), scene generation, simulation progress control, UAV model scheduling, message exchange between UAVs, and interactive result determination between UAVs and tasks. The GUI is developed based on PyQt5 (5.15.4), and pyqtgraph (0.11.1) is used for real-time graphing of the status of UAVs and tasks. The GA used in Algorithms 2, 4 and 5 and the global optimization in Section 4.5 are from the package sko (0.6.6).

A screenshot of the GUI is shown in Figure 3, and the meanings of different elements are shown in the legend on the right. The display of elements such as text labels and topological connections can be controlled in this interface to better observe the experimental effect.

To verify the effectiveness of the method in the heterogeneous UAV swarm scenario, four kinds of UAVs are set in the simulation scene, including a mini scouter, mini striker, mini scout and strike UAV, and medium scout and strike UAV. The UAVs are different in flight velocity, scout speed, number of strike loads, and the capability of strike load. The UAV types and their parameter settings are shown as Table 2.

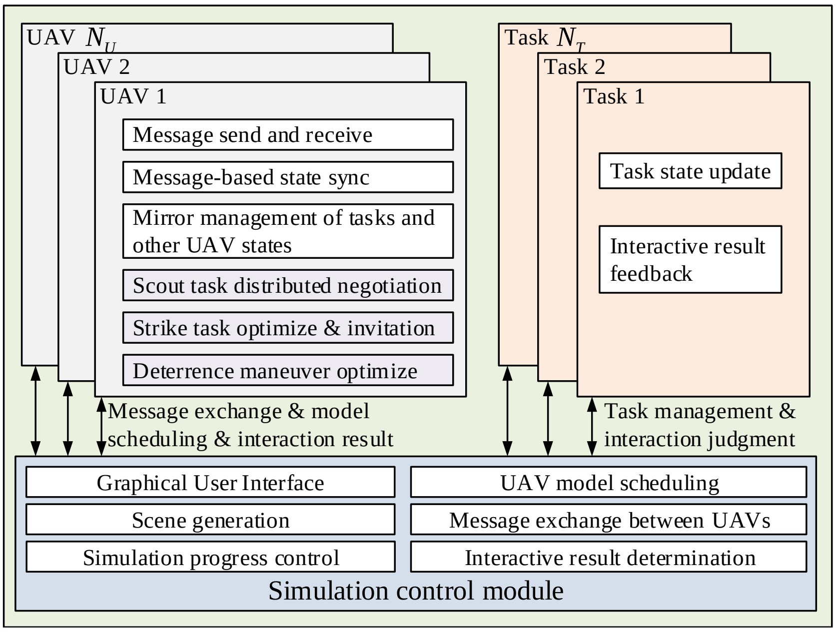

Figure 2. Distributed simulation environment for heterogeneous UAV reconnaissance and strike tasks.

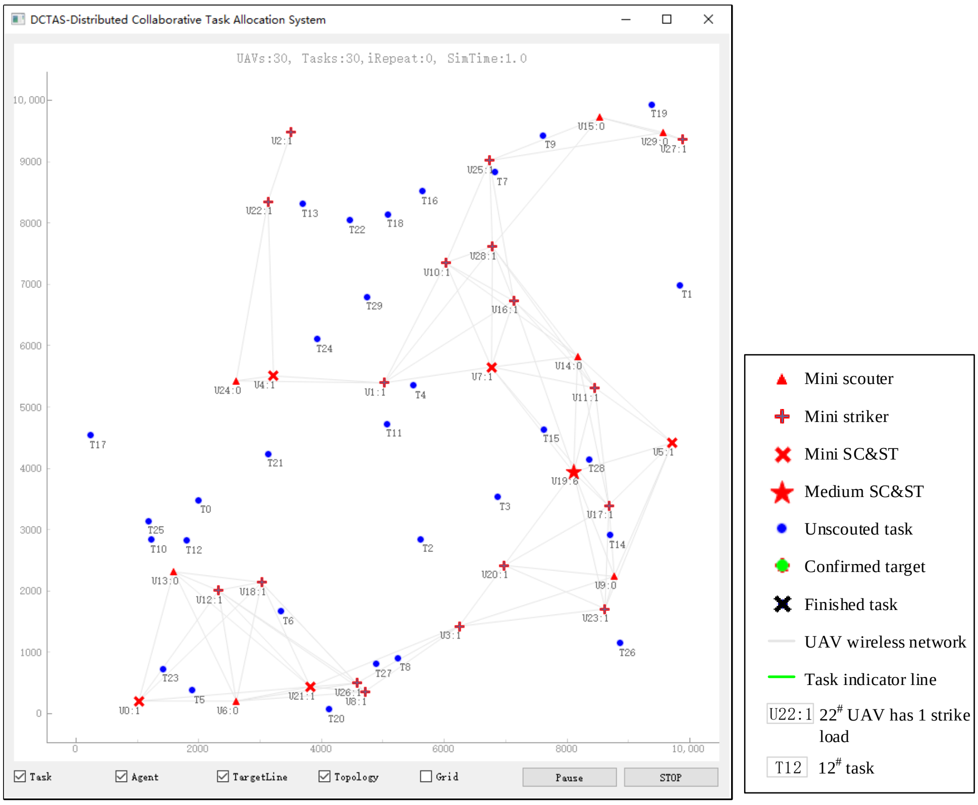

Figure 3. The generated initial scenario contains 30 UAVs and 30 tasks.

Table 2. UAV type and its parameter setting.

UAV Type	Velocity (m/s)	Scout Speed (m^2/s)	Number of Strike Loads	Capability Vector of Strike Loads
Mini scouter	40	10,000	0	—
Mini striker	50	—	1	[40, 40]
Mini SC&ST	50	6000	1	[80, 80]
Medium SC&ST	80	15,000	6	[100, 100]

To simulate the war fog and the dynamic characteristics of the mission, five types of tasks are set up in the experiment, each of which has a differently sized suspicious area and required capability vector, as shown in Table 3. The fake target indicates that there is no actual target in the region, and before the completion of reconnaissance, the specific information of any target is unknown. Therefore, UAV reconnaissance and strike forces need to cooperate more flexibly to reduce mission risk and the time interval from discovery to strike.

Table 3. Task type and its parameter setting.

Task Type	Area Size (m^2)	Required Capability Vector
Fake target	1×10^6	—
Target type1	5×10^5	[25, 30]
Target type2	2×10^6	[100, 80]
Target type3	2×10^6	[40, 150]
Target type4	4×10^6	[200, 200]

The optimization result of GA is greatly affected by the population size π_n, iteration number π_i and mutation probability π_p. The larger the problem space, the larger π_n and π_i should be, so as to carry out a broader search. For the GA used in Algorithms 2 and 4, we let the number of permutations for assigning tasks to UAVs as n_{perm}, and the parameters of the GA are adaptively adjusted according to n_{perm} before each task assignment optimization.

4.2. Scene Generation

The experiments were set up in a rectangular area of 10×10 Km, without considering the height. In order to better present the cooperative effect of the UAVs and observe the operation effect, we set the total number of UAVs and tasks to 30, respectively. The proportions of four types of UAVs are 6:20:6:2, respectively, and the specific number is the total number of UAVs multiplied by the ratio and then rounded down, and the excess number is added to the mini scouter. Therefore, when the total number of UAVs is set to 30, the specific number of each UAV type is: 7, 17, 5, 1. Similarly, the ratio between the five types of tasks is set to 15:7:4:2:1, and when the total number of tasks is 30, the number of corresponding tasks is 16, 7, 4, 2, 1, respectively.

When generating a scene, the tasks are first randomly assigned to the experimental area with a uniform distribution, each task is randomly assigned a task type, and the quantity requirements of each task type is met. Similarly, initial positions and UAV types are randomly assigned to each UAV. Then, the generated UAVs and tasks are registered into the simulation control module of the experimental environment, and the information of all UAVs and tasks is broadcast to each UAV as the initial information for decision-making.

The maximum single-hop communication distance of radio between UAVs is set as $d_\delta = 3$ Km, and multi-hop transmission is supported. The synchronization of state information and the negotiation of task assignments can only take place between two UAVs when there is a communication link. Under the above constraints, the generated initial scenario contains 30 UAVs and 30 tasks, as Figure 3 shows, and the corresponding task types are shown in Table 4.

Table 4. Task IDs and corresponding task types in the scenario.

Task Type	Tasks
Fake target	T1, T5, T6, T7, T8, T12, T13, T18, T19, T20, T21, T22, T23, T24, T27, T28
Target type1	T9, T10, T11, T14, T16, T17, T29
Target type2	T3, T4, T15, T25
Target type3	T0, T26
Target type4	T2

4.3. Reconnaissance Task Priority Assessment Results

Before optimizing the reconnaissance task assignment, each evaluator, i.e., reconnaissance-capable UAV, evaluates the prior order of each task to each UAV using the evaluation method defined in Section 3.1.1. We set the number of negotiation hops to $n_c = 2$, and the UAVs will take other reconnaissance UAVs within 2 hops into consideration. For the scenario in Figure 3, the prior order of tasks to each UAV evaluated by different UAVs are shown in Table 5, and the tasks marked in bold are selected by each evaluator using Algorithm 2 to participate in this round of assignment.

Table 5. The prior order of tasks to each UAV evaluated by different UAVs at time=1.0. The bold numbers are the tasks selected to participate in this round of assignment.

Evaluator	Task Prior Order to Each UAV	Evaluator	Task Prior Order to Each UAV
U0	U0: (**5, 23, 6, 20, 12, 10**, 25, ...) U6: (**5, 23, 6, 20, 12, 10**, 25, ...) U13: (**23, 5, 6, 12, 10**, 25, 20, ...) U21: (**5, 6, 23, 20, 12**, 27, 10, ...)	U14	U14: (**15, 28, 3, 1, 14, 4, 7**, ...) U5: (**28, 15, 14, 3, 1, 4, 11**, ...) U7: (**15, 28, 3, 4, 14, 1**, 11, ...) U9: (**28, 15, 14, 3, 4, 1, 11**, ...) U15: (**1, 7**, 15, 28, 9, 4, 3, ...) U19: (**28, 15, 3, 14, 4, 1, 11**, ...)
U4	U4: (**24, 21, 29, 11**, 4, 0, 13, ...) U7: (**4**, 11, 29, 24, 21, 22, 13, ...) U24: (**24, 21, 29, 11**, 4, 0, 13, ...)	U15	U15: (**9, 19, 7, 16, 1, 18**, 22, ...) U7: (**7, 9, 16, 19, 1**, 18, 15, ...) U14: (**7, 9, 1, 19**, 16, 18, 15, ...) U29: (**19, 9, 7, 1, 16**, 18, 22, ...)
U5	U5: (**28, 14, 15, 3, 1, 26, 4**, ...) U7: (**15, 28, 14, 3, 1, 4**, 26, ...) U9: (**14, 28, 15, 3, 26, 1**, 2, ...) U14: (**28, 15, 14, 3, 1, 26, 4**, ...) U19: (**28, 14, 15, 3, 1, 26, 4**, ...)	U19	U19: (**14, 28, 15, 3, 2**, 26, 4, ...) U5: (**28, 14, 15, 3**, 26, 2, 4, ...) U7: (**15, 3, 28, 14**, 4, 11, 2, ...) U9: (**14, 28, 3, 15, 26**, 2, 4, ...) U14: (**15, 28, 3, 14, 4**, 2, 11, ...)
U6	U6: (**6, 20, 5, 23, 27, 8**, 12, ...) U0: (**5, 23, 6, 20** , 27, 12, 10, ...) U13: (**6, 23, 5, 20**, 12, 27, 10, ...) U21: (**6, 20, 5, 27, 23**, 8, 12, ...)	U21	U21: (**6, 20, 27, 8, 5, 23**, 2, ...) U0: (**5, 23, 6, 20, 27, 8**, 12, ...) U6: (**6, 20, 5, 27, 23, 8, 12**, ...) U9: (**8, 27, 2**, 20, 6, 3, 5, ...) U13: (**6, 5, 23, 20, 27, 8, 12**, ...)
U7	U7: (**15, 28, 3, 4, 11, 29, 24**, ...) U4: (**4, 11, 29, 24**, 3, 15, 28, ...) U5: (**28, 15, 3, 14, 4, 11**, 1, ...) U9: (**28, 3, 15, 14, 4, 11**, 2, ...) U14: (**15, 28, 3, 4, 11, 29, 14**, ...) U15: (**7**, 15, 28, 16, 4, 29, 9, ...) U19: (**15, 28, 3, 4, 14, 11, 2**, ...) U24: (**4, 11, 24**, 29, 3, 15, 28, ...)	U9	U9: (**14, 28, 26, 3, 15**, 2, 8, ...) U5: (**14, 28, 15, 3, 26, 2, 8**, ...) U7: (**28, 15, 3, 14, 26, 2, 4**, ...) U14: (**28, 15, 14, 3, 26, 2, 4**, ...) U19: (**14, 28, 3, 15, 26, 2, 8**, ...) U21: (**3, 14, 26, 2, 28**, 8, 15, ...)
U24	U24: (**21, 24, 0**, 29, 11, 4, 12, ...) U4: (**24, 21, 11**, 29, 0, 4, 13, ...) U7: (**24, 4**, 11, 29, 21, 13, 22, ...)	U29	U29: (**19, 9, 1**, 7, 16, 18, 15, ...) U15: (**19, 9, 7**, 1, 16, 18, 15, ...)
U13	U13: (**6, 23, 5, 12, 0**, 10, 25, ...) U0: (**23, 5, 6, 12**, 10, 25, 0, ...) U6: (**6, 5, 23, 12, 20**, 10, 0, ...) U21: (**6, 5, 23, 20**, 12, 0, 10, ...)	—	—

Taking the evaluation results of U7 as an example, the reconnaissance capable UAVs within 2 hops are U4, U5, U9, U14, U15, U19, U24, and U7 itself. The priority of task T15 ranks first for U7 since T15 is relatively close to U7, and the position of T15 can obtain more sufficient strike capability. Although T4 is the closest to U7, it gets a prior order of 4th for U7 because the unknown potential synergistic relationship between T11 and T4 increases the risk of T4, and the low coverage of UAVs at T4 further reduces its priority.

It can be found that there is a large gap between U4's task prior order as assessed by U7 and U4. T24 ranked after U4 because T24 has a great advantage in distance. However, the attention mechanism of U7 makes it pay more attention to the surrounding tasks, so the priority of T24 to U4 is suppressed in the evaluation of U7. This cognitive difference caused by inconsistent environmental cognition or subjective preferences can be corrected during the negotiation process with the other party.

Because each evaluator only considers several tasks with the highest priority in each round of allocation, the search space for task-allocation optimization can be effectively reduced.

4.4. Reconnaissance Task Assignment

After the reconnaissance UAVs generate a reconnaissance plan and share it with each other, each node uses Algorithm 3 to fuse the received plan with its own plan. The fusion process of U7 is as Figure 4 shows, and at time=1, U7 tends to give priority to nearby tasks due to its own attention mechanism, so its assignments include U4→T11, U14→T4, and U24→T29. However, this attention mechanism can cause inconsistencies among the generated plans when the distance between two UAVs is large. When merging the received plans with its own, U7 will take out the inconsistent part and redistribute it. At this time, its attention mechanism will be disabled, making the integrated plan more consistent with the views of each participant.

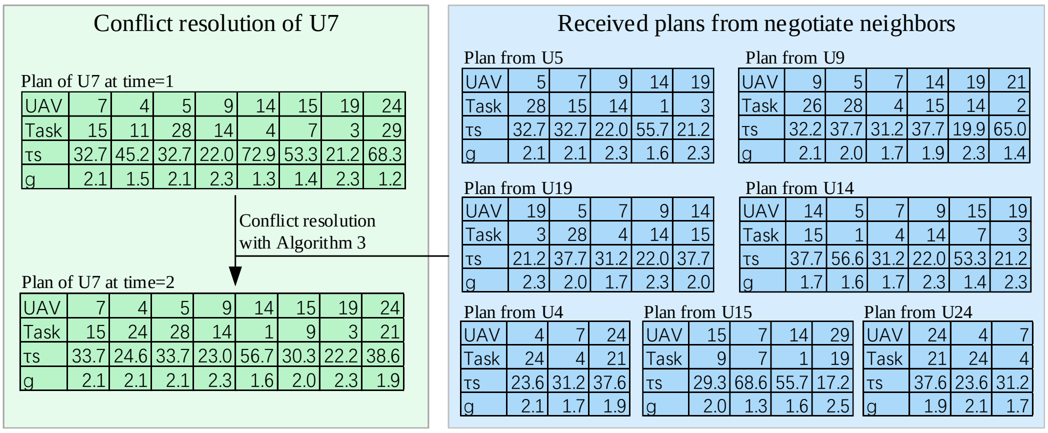

Figure 4. Negotiation process based on conflict resolution.

When the conflict is resolved, the assignment of U7 in Figure 4 is consistent with the updated assignments of other nodes, so this assignment is finally adopted and implemented. The allocation results of the first round of reconnaissance tasks are shown as the green target lines of each reconnaissance node in Figure 5a, and the specific allocation information is shown in Table 6. The topology after a period of execution is shown in Figure 5b.

From the allocation results, it can be found that the distributed allocation algorithm generally follows the principle of minimizing the completion time, but it also reflects the algorithm's expectation of enhancing the superiority over enemy and reducing mission risks. For example, U13 chose T6 instead of T12 as the first mission, because it is easier for the UAV swarms to form a superiority over enemy at T6. In contrast, the location of T12 is too dense and risky, and it should be executed after more UAVs are concentrated. The mission groups (T5, T23) and (T15, T28) are also relatively dense, but since the UAV swarm has a capability advantage here, the strategy of coordinating in time is adopted to reduce the mission risk. In the time-coordinated formation, the UAVs that could have arrived earlier choose to reduce the flight speed so that the formation can reach the targets at the same time, thereby avoiding the coordinated strike of the enemy due to individual exposure.

When each UAV adopts the distributed task assignment algorithm in this paper, the task scheduling Gantt chart is as Figure 6 shows. In it, the blue boxes represent the reconnaissance behavior, and the red boxes represent strike behavior. From the Gantt chart, it can be found that for UAVs with both reconnaissance and strike capabilities, such as U0 and U5, when their own capabilities can meet the target capability requirements, the strike can be carried out immediately. For example, U0's strike on T10 and U5's strike on T11 are instant. For targets with strong defense capabilities, the coordinated strike of multiple UAVs is required. For example, the strikes on T0 and T26 are all completed by the cooperation of four UAVs. Since the nodes with strike capability will maneuver to the reconnaissance nodes for deterrence when they are idle, it allows reconnaissance nodes that do not have strike capability can also strike quickly after discovering the target. For example, U25 can launch a strike on T9 (discovered by U15) within 8 s after confirming the strike mission.

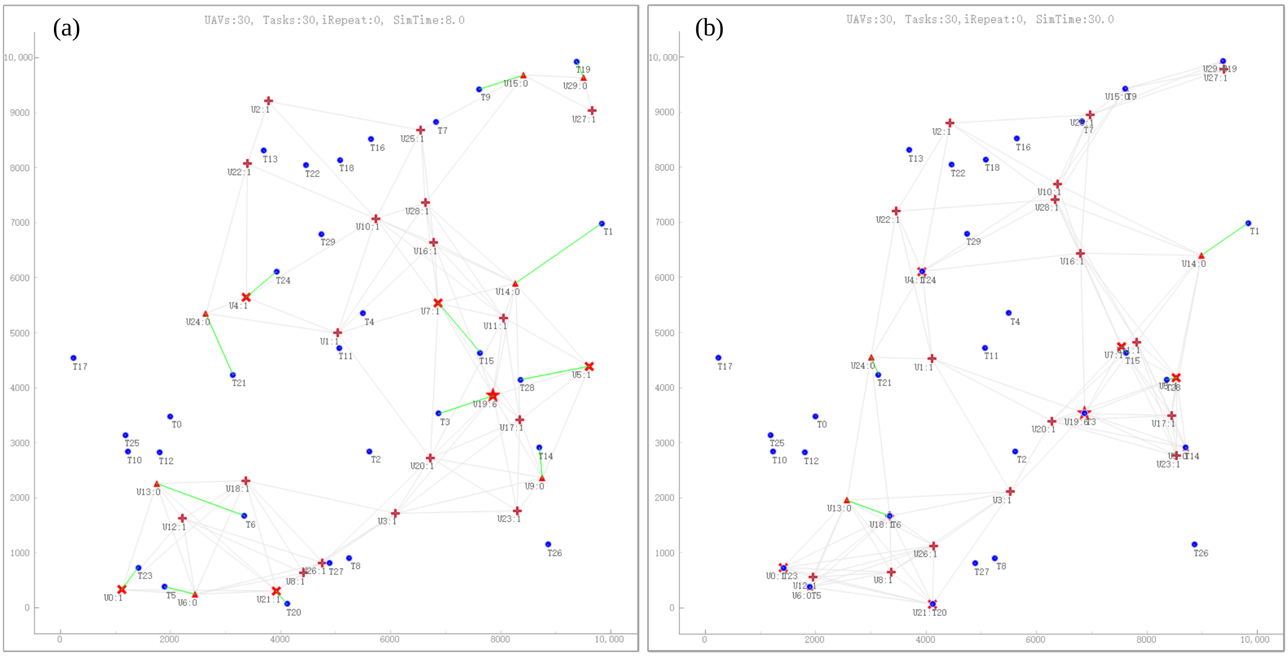

Figure 5. Topology diagram for different simulation times. (**a**) Each reconnaissance UAV has confirmed the reconnaissance task before time = 8. (**b**) Each reconnaissance node executes reconnaissance task, and the strike nodes perform deterrence maneuver at time = 30. The meanings of elements are consistent with those in Figure 3.

Table 6. Allocation information and time coordination relationship of the first round of reconnaissance tasks.

UAV ID	Current Pos	Task ID	Task Pos	τ_j^s(s)	Planned v_i(m/s)	Maximum v_i(m/s)	Collaborate UAVs
U0	(1029.8, 204.3)	T23	(1420.5, 729.8)	23.38	35.63	50	U6
U6	(2618.0, 202.9)	T5	(1904.0, 377.2)	23.38	40.00	40	U0
U5	(9722.0, 4410.0)	T28	(8364.7, 4143.4)	34.66	50.00	50	U7
U7	(6776.9, 5650.7)	T15	(7634.8, 4636.8)	33.66	48.01	50	U5
U4	(3223.9, 5507.1)	T24	(3931.8, 6108.0)	23.57	50.00	50	—
U9	(8777.8, 2235.3)	T14	(8712.1, 2910.3)	22.95	40.00	40	—
U13	(1596.7, 2313.2)	T6	(3348.8, 1669.4)	51.66	40.00	40	—
U14	(8179.1, 5827.7)	T1	(9846.0, 6986.1)	56.75	40.00	40	—
U15	(8543.8, 9725.0)	T9	(7617.9, 9429.2)	30.30	40.00	40	—
U19	(8114.8, 3937.6)	T3	(6881.7, 3528.7)	22.24	80.00	80	—
U21	(3826.9, 436.9)	T20	(4124.1, 67.6)	15.48	50.00	50	—
U24	(2615.7, 5425.9)	T21	(3144.3, 4235.8)	39.56	40.00	40	—
U29	(9578.5, 9477.7)	T19	(9387.9, 9928.5)	17.24	40.00	40	—

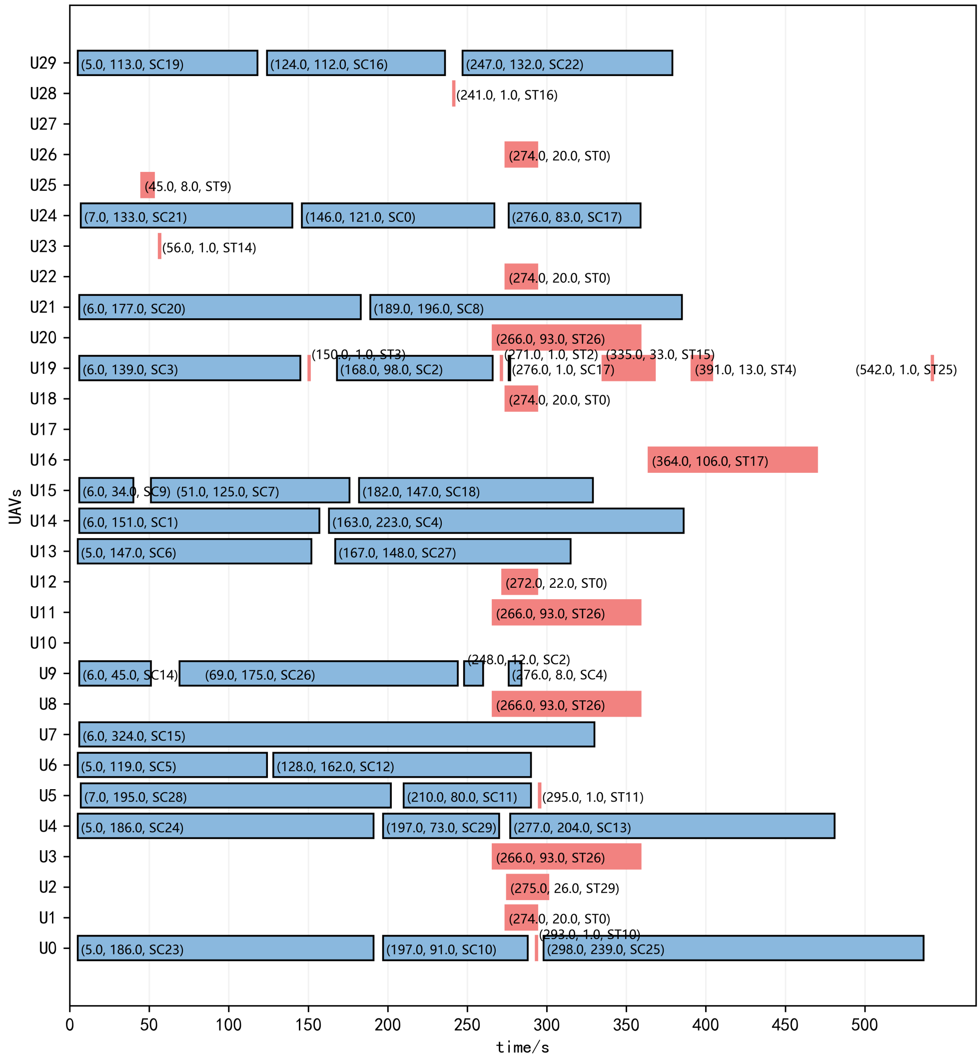

Figure 6. Gantt chart for reconnaissance and strike tasks. Blue boxes represent reconnaissance and red boxes represent strikes, and the triple elements represent mission confirmation time, duration of maneuver and task execution, and the concatenation of task type and task ID respectively.

4.5. Comparison with Centralized Global Optimization Based on GA

In the proposed distributed framework, each UAV only focuses on the nearby UAVs and tasks, and negotiates with the nearby UAVs to resolve the conflicts in the combat plans. This distributed solution not only realizes decoupling between UAVs, but also effectively reduces the search space of the task-allocation problem, and can improve the speed of solving the problem. Based on the scenario shown in Figure 3, we compare the proposed method with the centralized global optimization method based on GA in terms of solving speed and quality. The experimental platform is a MSI GS65 notebook installed with Windows system. Its CPU is i7-8750H, GPU is GTX1070 Max-Q, memory is 32 G, and the SSD is 512 G.

We have counted the decision-making time consumption of each reconnaissance UAV in steps 1-8 during the optimization of reconnaissance task allocation, because all UAVs have confirmed their tasks at the end of the 8th simulation step. The results are shown in Figure 7, and it can be found that the time consumed by each decision of each UAV is less than 0.25 s. Without considering the communication delay, if the maximum decision time of each step is taken as the cycle of this round of iteration, the total time of 8 simulation steps is 1.02 s. From the time-consumption distribution, it can also be found that the first step of decision-making only consumes a little time, while the second step of decision-making takes the most time, and then decreases gradually. This is because before the first decision, there is no communication between nodes, and each node makes decisions independently. The second decision is made after exchanging the results of the first round, and at this time, each node is performing conflict resolution on multiple collected plans, so it takes a lot of time. After the second step, the conflict between plans is gradually resolved, so the decision-making time is also shortened and the final task-allocation plan is formed. The optimization results are shown in Table 6, and from the global perspective, its fitness is 27.48 using Equation (9).

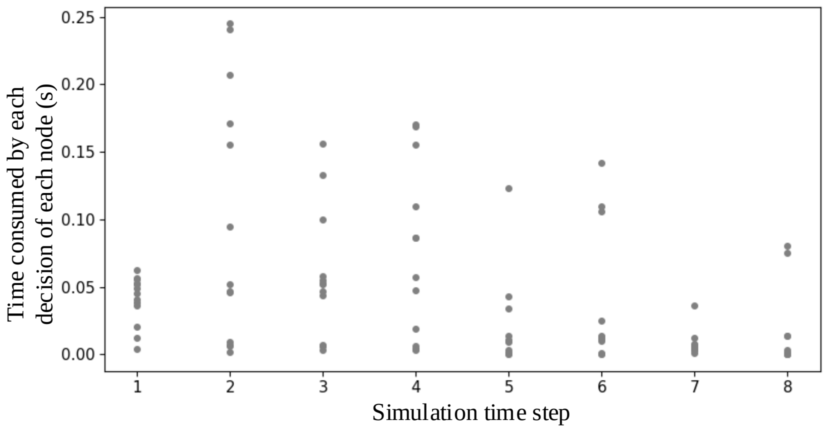

Figure 7. Time consumption of each UAV in the proposed method.

In the same scenario, we further use the centralized GA to optimize the reconnaissance task-allocation problem from a global perspective, and the fitness curve obtained is shown as the two `CGA` curves in Figure 8. In our proposed algorithm, the population number π_n and iteration number π_i of GA are automatically adjusted according to the number of permutations of the assignment problem. When this strategy is applied to the centralized method, the obtained parameters of GA are $\pi_n = 32$, $\pi_i = 30$ and $\pi_p = 0.1$. However, the search space for the global optimization problem of assigning 30 tasks to 13 UAVs is too large, so GA is difficult to converge to a good result under these parameters. In order to further expand the search of the global GA to obtain a better result, we adjust the parameters to $\pi_n = 400$, $\pi_i = 200$, $\pi_p = 0.3$. We can find that after 27 rounds of iteration, it has obtained a result with fitness close to that of the proposed paper, which consumes about 4.9 s.

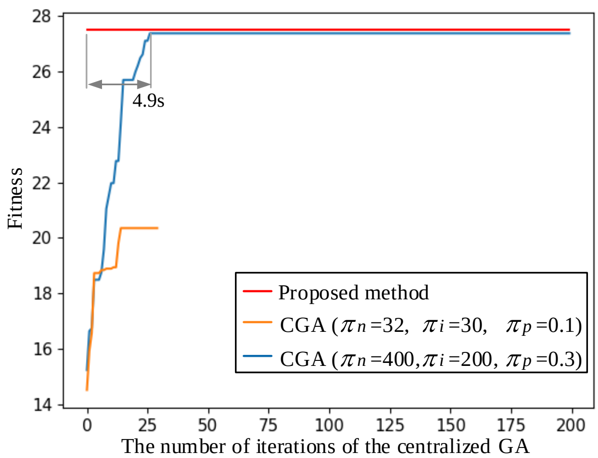

Figure 8. The fitness and time consumption of centralized GA.

Comparing the time consumption of the two methods, it can be found that the proposed method effectively reduces the computing load of each single UAV through the idea of divide and conquer, and can be applied to more types of small UAVs. By observing the change of fitness, we found that it is easy for centralized global optimization to fall into local optimal solution when the problem space is large. If the swarm size is further increased, it will be difficult for the centralized global optimization method to obtain good results, while distributed collaborative optimization has better scalability.

4.6. Comparison with No Time Coordination and Deterrence Maneuver

To verify the effect of the proposed method on reducing the risk of mission execution and improving the deterrence of strike capability, the spatial density of tasks represented by Equation (2) is taken as the scout risk; the strike capability availability of tasks represented by Equation (5) is taken as the capability coverage. In each simulation scenario, the average risk at the beginning of each scout task and the average strike capability coverage for scouting tasks during the whole simulation process are counted. The comparison results with the algorithms without time coordination or deterrence maneuver are shown in Table 7, and each result is the statistics of the mean and standard deviation of 10 simulation scenarios.

Table 7. The scout risk and strike capability coverage compared with no time coordination or deterrence maneuver. Each result is the mean and standard deviation of 10 simulation scenarios.

Deterrence Maneuver Type	**Enable Time Collaboration**		**Disable Time Collaboration**	
	Scout Risk	**Cap. Coverage**	**Scout Risk**	**Cap. Coverage**
Enable deterrence maneuver	0.238 ± 0.065	153.1 ± 61.3	0.311 ± 0.090	166.6 ± 63.9
Disable deterrence maneuver	0.234 ± 0.048	131.5 ± 33.7	0.311 ± 0.091	116.7 ± 31.5

The statistics of the results show that time coordination can reduce the scout risk by about 23%, and deterrence maneuver can improve the strike capability coverage by about 30%, which verifies the effectiveness of time coordination and maneuver deterrence.

4.7. Discussion

4.7.1. Computational Complexity Analysis

In the proposed method, the optimization of scout-task assignment needs to optimize task allocation and time coordination between UAVs, which is the part with high computational complexity of our proposed method. Therefore, analyzing the computational complexity of this part will aid further improvement.

In Algorithm 2, the most time-consuming process is to use line 24 to optimize the plan using GA. The computational complexity of GA can be expressed as $O(\pi_p \times \pi_i)$, where π_p is the population size and π_i is the number of iterations. For each plan generated by the GA, its fitness will be calculated through line 17-21 of Algorithm 2. Among them, line 20 uses Algorithm 1 for time collaborative optimization. Let the number of UAVs in the plan be N. The worst case of the while loop of Algorithm 1 will iterate $N-1$ times, and each loop needs to calculate $N \times (N-1)$ time alignment reward gains according to Equation (12), so the complexity of Algorithm 1 is about $O(N^3)$. Then line 21 of Algorithm 2 uses Equation (9) to calculate the fitness of the plan, in which the start time between any two UAVs needs to be compared to evaluate the task threat, and thus the complexity is about $O(N^2)$.

Therefore, the overall computational complexity of Algorithm 2 is about $O(\pi_p \times \pi_i \times (N^3 + N^2)) \approx O(\pi_p \times \pi_i \times N^3)$. Among them, the settings of π_p and π_i not only affect the calculation cost, but also affect the quality of the optimization results. In this paper, these two parameters are simply linearly mapped from the number of allocation combinations, and before deployment, it is necessary to further study the setting strategy of these parameters to compromise between the calculation cost and the optimization quality.

4.7.2. Method Characteristics under Different Network Connectivity

When the distributed UAV swarm is running, each UAV needs to perform regular state synchronization and event-triggered task assignment negotiation with UAVs within n_c hops, which also means that each UAV needs to process information from other UAVs within n_c hops. When the mission area is relatively scattered and the connectivity of the UAV network is low, the use of a limited range of autonomous collaboration can reduce the consumption of the network and reduce the computational power consumption of each UAV. While improving the reliability of the cluster, the distributed architecture can also enhance the scalability of the cluster.

However, when the UAVs are concentrated, such as when the network is fully connected in extreme cases, adopting a distributed architecture will reduce the operating efficiency of the system. The fully connected network requires each UAV to process the information of the entire battlefield, which indicates that each UAV should be equipped with high-performance computing resources. Moreover, the additional negotiation communication required by the distributed architecture will also increase the consumption of the network.

Therefore, the proposed distributed approach should be combined with centralized control when applied, and dynamically switch between the two according to the network connectivity status.

4.7.3. Influence of Network Instability on the Proposed Method

The mutual communication between UAVs is the basis for task-assignment negotiation and coordination during task execution. Due to the high-speed maneuvering of UAVs in 3D space, the topology of the flying ad hoc network (FANET) [22] changes rapidly, which may cause communication interruption, delay increase, and other problems. These problems may further make the negotiation period of task assignment longer, or even cause conflicts between the assigned plans due to the interruption of communication, and ultimately make the overall task assignment result worse. Therefore, the next step is to evaluate the impact of network instability on the method and study the corresponding strategies to improve its robustness, which can also avoid making this problem a vulnerable point to attack by the enemy.

Since the cooperation of UAVs in the proposed method mainly occurs between UAVs within 2 hops, the route-maintenance strategy of FANET can pay more attention to the optimization of short-distance routes to improve the QoS, which can not only improve the speed of distributed task allocation in this paper, but also help avoid collisions between UAVs and so on.

5. Conclusions

Due to the high risk of UAV clusters in executing reconnaissance and strike tasks under the condition of insufficient enemy information and potential synergy between targets, a distributed task-collaborative allocation method for heterogeneous UAV swarms is proposed. This method establishes a distributed task-allocation framework composed of a reconnaissance task-allocation method based on a negotiation mechanism and a strike task-allocation method based on an invitation mechanism. The reconnaissance task-allocation algorithm evaluates the task priority according to the superiority of the UAVs against the tasks to reduce the complexity of the optimization problem. Reconnaissance UAVs adopt a time-coordination strategy for reconnaissance, and UAVs with strike capabilities perform deterrent maneuvers when they are idle to reduce mission risks during mission execution. This method enables the UAV swarm to negotiate the allocation of tasks in a distributed framework, and at the same time, the evaluation of the capability advantage over the enemy, time coordination, and deterrence maneuver mechanism effectively reduce the risk of unknown targets to UAVs. The distributed framework not only improves the scalability of the swarm, but also enhances its reliability in the battlefield with a more complex electromagnetic environment.

Further research should include a more efficient algorithm that takes the negotiation mechanism and the network state of the swarm as prior information to replace the GA for the local optimization of UAVs, and should aim to obtain the best operating efficiency under different network connectivity. The proposed method should be further combined with a centralized or hierarchical task-allocation framework.

Author Contributions: Conceptualization, H.D. and J.H.; methodology and writing—original draft, H.D.; writing—review and editing, Q.L., C.Z., T.Z. and J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Shakhatreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634.
2. Qin, B.; Zhang, D.; Tang, S.; Wang, M. Distributed Grouping Cooperative Dynamic Task Assignment Method of UAV Swarm. *Appl. Sci.* **2022**, *12*, 2865. [CrossRef]
3. Zhang, J.; Xing, J. Cooperative task assignment of multi-UAV system. *Chin. J. Aeronaut.* **2020**, *33*, 2825–2827. [CrossRef]
4. Jiang, X.; Zeng, X.; Sun, J.; Chen, J. Research status and prospect of distributed optimization for multiple aircraft. *Acta Astronaut.* **2021**, *42*, 524551. (In Chinese) [CrossRef]
5. Zhen, Z.; Xing, D.; Gao, C. Cooperative search-attack mission planning for multi-UAV based on intelligent self-organized algorithm. *Aerosp. Sci. Technol.* **2018**, *76*, 402–411. [CrossRef]
6. 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]
7. Ma, Y.; Zhao, Y.; Bai, S.; Yang, J.; Zhang, Y. Collaborative task allocation of heterogeneous multi-UAV based on improved CBGA algorithm. In Proceedings of the 16th International Conference on Control, Automation, Robotics and Vision, Shenzhen, China, 13–15 December 2020; pp. 795–800.
8. Dai, W.; Lu, H.; Xiao, J.; Zeng, Z.; Zheng, Z. Multi-Robot Dynamic Task Allocation for Exploration and Destruction. *J. Intell. Robot. Syst.* **2019**, *98*, 455–479. [CrossRef]
9. Sheng, W.; Yang, Q.; Tan, J.; Xi, N. Distributed multi-robot coordination in area exploration. *Robot. Auton. Syst.* **2006**, *54*, 945–955. [CrossRef]
10. Ye, F.; Chen, J.; Sun, Q.; Tian, Y.; Jiang, T. Decentralized task allocation for heterogeneous multi-UAV system with task coupling constraints. *J. Supercomput.* **2020**, *77*, 111–132. [CrossRef]
11. Chen, J.; Wu, Q.; Xu, Y.; Qi, N.; Guan, X.; Zhang, Y.; Xue, Z. Joint Task Assignment and Spectrum Allocation in Heterogeneous UAV Communication Networks: A Coalition Formation Game-Theoretic Approach. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 440–452. [CrossRef]
12. Jiang, Y. A Survey of Task Allocation and Load Balancing in Distributed Systems. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 585–599. [CrossRef]
13. Li, L.; Xu, S.; Nie, H.; Mao, Y.; Yu, S. Collaborative Target Search Algorithm for UAV Based on Chaotic Disturbance Pigeon-Inspired Optimization. *Appl. Sci.* **2021**, *11*, 7358. [CrossRef]
14. Hu, J.; Wu, H.; Zhan, R.; Menassel, R.; Zhou, X. Self-organized search-attack mission planning for UAV swarm based on wolf pack hunting behavior. *J. Syst. Eng. Electron.* **2021**, *32*, 1463–1476.
15. Choi, H.; Brunet, L.; How, J.P. Consensus-Based Decentralized Auctions for Robust Task Allocation. *IEEE Trans. Robot.* **2009**, *25*, 912–926. [CrossRef]
16. Edalat, N.; Tham, C.; Xiao, W. An auction-based strategy for distributed task allocation in wireless sensor networks. *Comput. Commun.* **2012**, *35*, 916–928. [CrossRef]
17. Choi, H.; Kim, Y.; Kim, H.J. Genetic algorithm based decentralized task assignment for multiple unmanned aerial vehicles in dynamic environments. *Int. J. Aeronaut. Space Sci.* **2011**, 163–174. [CrossRef]
18. Patel, R.; Rudnick-Cohen, E.; Azarm, S.; Otte, M.; Xu, H.; Herrmann, J.W. Decentralized Task Allocation in Multi-Agent Systems Using a Decentralized Genetic Algorithm. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 3770–3776.

19. Wu, H.; Li, H.; Xiao, R.; Liu, J. Modeling and simulation of dynamic ant colony's labor division for task allocation of UAV swarm. *Physica A* **2018**, *491*, 127–141. [CrossRef]
20. Cao, Y.; Wei, W.; Bai, Y.; Qiao, H. Multi-base multi-UAV cooperative reconnaissance path planning with genetic algorithm. *Clust. Comput.* **2019**, *22*, 5175–5184. [CrossRef]
21. Yu, W.; Ai, T.; Shao, S. The analysis and delimitation of Central Business District using network kernel density estimation. *J. Transp. Geogr.* **2015**, *45*, 32–47. [CrossRef]
22. Khan, M.A.; Safi, A.; Qureshi, I.M.; Khan, I.U. Flying ad-hoc networks (FANETs): A review of communication architectures, and routing protocols. In Proceedings of the 2017 First International Conference on Latest trends in Electrical Engineering and Computing Technologies, Karachi, Pakistan, 15–16 November 2017; pp. 1–9.