Scientific Research Publishing

Journal of Computer and Communications, 2026, 14(11), 164-183
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Post-Disaster Heterogeneous Multi-UAV Mission Planning Based on Bi-Level Multi-Objective Optimization

Ziye Shi[1], Wenxing Wu[2], Liqin Tian[1*]

[1]Graduate School, University of Emergency Management, Yanjiao, China
[2]School of Computer Science, Qinghai Normal University, Xining, China
Email: *tianliqin@ncist.edu.cn

How to cite this paper: Shi, Z.Y., Wu, W.X. and Tian, L.Q. (2026) Post-Disaster Heterogeneous Multi-UAV Mission Planning Based on Bi-Level Multi-Objective Optimization. *Journal of Computer and Communications*, **14**, 164-183.
https://doi.org/10.4236/jcc.2026.1411008

Received: October 14, 2026
Accepted: November 17, 2026
Published: November 20, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/

Open Access

Abstract

Aiming at the cooperative task planning problem of heterogeneous multi-UAVs for post-disaster emergency rescue, this paper proposes a hierarchical task allocation and path optimization method based on bi-level programming. The upper layer employs an improved Multi-Objective Particle Swarm Optimization (MOPSO) for task allocation, optimizing total flight distance and maximum completion time simultaneously. The lower layer applies standard Particle Swarm Optimization (PSO) to independently plan the access sequence for each UAV and generate a complete loop departing from the depot, visiting task points and returning to the depot. To address search difficulties of multi-objective optimization in high dimensional discrete spaces, this paper proposes a warm start strategy based on greedy proximity assignment, which injects initial solutions into the population to elevate search starting points. Meanwhile, a distance aware load balancing post processing mechanism is designed. It re-balances task allocation within homogeneous UAV groups to minimize additional flight range and effectively prevents sharp distance growth induced by post processing. Experiments are conducted on a post-disaster scenario containing 14 heterogeneous UAVs and 50 disaster affected points. Four methods including K means + PSO, single layer MOPSO, CP SAT + PSO and greedy assignment + PSO are used for comparison. Experimental results demonstrate that the proposed method performs excellently in terms of maximum completion time and load balance. It reduces the maximum completion time by 71% relative to CP-SAT + PSO and achieves better load balance than all comparative methods. Meanwhile, its total flight distance is 7.6% lower than K-means + PSO, albeit marginally higher than CP-SAT + PSO and Greedy + PSO. With favorable scalability and generality, the proposed method can flexibly incorporate complex constraints such as

flight range constraints and task time windows.

Keywords

Multi-UAV Task Planning, Bi-Level Programming, Multi-Objective Particle Swarm Optimization, Heterogeneous UAVs, Post-Disaster Rescue, Load Balancing

1. Introduction

With the frequent occurrence of natural disasters and the rapid development of the low altitude economy, unmanned aerial vehicles (UAVs), boasting advantages such as high maneuverability, fast response speed, and the capability to carry multiple payloads, are playing an increasingly important role in emergency rescue scenarios including post-disaster reconnaissance, material delivery and life searching missions [1]. After disasters such as major earthquakes, floods or landslides strike, disaster stricken areas are usually characterized by complex terrain and wide coverage. Restricted by payload capacity, flight endurance and task functions, a single UAV can hardly accomplish the entire rescue mission independently. Accordingly, adopting a cooperative formation composed of multiple heterogeneous UAVs [2] [3] to achieve fast and full coverage operations over multiple disaster sites through reasonable task allocation and path planning has become a research hotspot in the field of emergency rescue.

Nevertheless, cooperative task planning for heterogeneous multi-UAV systems in post-disaster environments represents an extremely challenging combinatorial optimization problem. This problem requires not only determining which subtasks each UAV should undertake, but also planning the optimal visiting sequence of task points for every UAV. The two components are tightly coupled and mutually interactive. The output of task allocation directly determines the feasibility and cost of path planning; in turn, accurate path distances affect the global optimality of task allocation [4]. Meanwhile, type matching constraints, payload carrying constraints and endurance constraints of heterogeneous UAVs, as well as conflicts among multiple objectives such as total flight distance and maximum completion time, further increase the difficulty of problem solving. How to rapidly obtain high quality feasible solutions within this high dimensional, multi-constrained and multi-objective complex solution space constitutes a key scientific issue to be urgently addressed.

Scholars at home and abroad have carried out extensive research on UAV task planning problems. Early studies mainly focused on single UAV path planning, and methods such as the A* algorithm and Rapidly exploring Random Tree (RRT) were frequently adopted for geometric path searching. With the growing demand for multi-UAV cooperation, researchers have turned their attention to the joint optimization of task allocation and path planning. For multi-UAV task allocation,

mainstream approaches include market auction based distributed allocation algorithms such as CBBA [5], evolutionary and swarm intelligence based heuristic optimization algorithms including Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) [6]-[8], as well as exact solution methods based on mixed integer programming [9]. Among them, GA and PSO have been widely applied in handling complex constraints owing to their favorable global search capability. In terms of multi-UAV path planning, the Traveling Salesman Problem (TSP) and its variants serve as the most commonly used modeling frameworks. PSO, ACO and deep learning based methods represent the mainstream technical routes for solving such problems [10] [11].

In recent years, the bi-level programming framework has been introduced into the field of multi-UAV collaborative optimization, offering new insights for addressing the coupled problem of task allocation and path planning. Bi-level programming decomposes a complex problem into upper level and lower level decisions. Outputs from upper level decisions act as inputs for lower level decisions, while feedback from lower level decisions affects the optimality of upper level decisions. Global optimization is achieved through iterative coordination. Yu Jing *et al.* [12] constructed a mutually coupled bi-level UAV planning framework. The upper layer adopted the SAS algorithm for task allocation optimization, and the lower layer employed the ant colony algorithm to obtain task sequences and flight paths. Bi-level iterative optimization can effectively avoid local optima. Wen *et al.* [13] proposed a bi-level collaborative planning framework for multi depot scenarios. An improved NSGA II algorithm was used for upper level task allocation, and an improved PSO was applied at the lower layer for path generation. It outperformed single level solution approaches under complex constraints. Han *et al.* [14] built a bi-level optimization framework for space air ground integrated networks. The outer layer utilized an evolutionary strategy to optimize computation offloading allocation, and the inner layer adopted a branch and price algorithm to solve UAV path problems with time windows, which greatly improved scheduling efficiency for emergency scenarios.

For multi-objective optimization, particle swarm optimization has been widely used in UAV task planning thanks to its fast convergence, few parameters and ease of implementation. In 2004, Coello Coello *et al.* [15] proposed the classic Multi-objective Particle Swarm Optimization (MOPSO). By introducing an external archive mechanism and a crowding distance based leader selection strategy, PSO can efficiently search for the Pareto optimal front, and MOPSO has become one of the benchmark algorithms in multi-objective optimization. Proposed by Deb *et al.* [16], NSGA II realizes a favorable balance between convergence and solution diversity via fast non dominated sorting and crowding distance calculation, and is frequently adopted as a comparison baseline. In recent years, researchers have put forward targeted improvements to MOPSO according to the characteristics of UAV task planning. Wang Yu *et al.* [17] presented a hybrid strategy based multi-objective particle swarm optimization algorithm for heterogeneous

UAV collaborative multi task allocation. A constraint driven dynamic particle preference initialization strategy and a dominance based superior individual selection strategy were adopted to enhance convergence accuracy and solution set diversity. Xiao Peng *et al.* [18] applied chaotic adaptive multi-objective particle swarm optimization to multi-UAV cooperative search missions. Chaotic mapping and dynamic inertia weight were introduced to strengthen the algorithm's adaptability in dynamic environments.

Significant progress has also been made in the research on cooperative task planning for heterogeneous multi-UAVs within post-disaster rescue application scenarios. Yu Qingqing *et al.* [19] systematically sorted out various domestic application scenarios of UAVs for post-disaster rescue, analyzed existing deficiencies in industrial regulations, air space management and equipment technologies, and pointed out that UAV cluster cooperation serves as a critical development direction for improving disaster rescue efficiency. Wang Kan *et al.* [20] reviewed diverse applications of UAV aided mobile edge computing, including post-disaster emergency communication scenarios. They indicated that the joint optimization of UAV trajectory, task offloading and resource allocation constitutes the core foundation for enhancing communication and search and rescue efficiency in disaster stricken areas. Gan Liangqi *et al.* [21] established a UAV cluster networking scheme for post-disaster emergency communication and compared the scenario adaptability of different cluster deployment and network resource scheduling strategies.

Although existing studies have conducted substantial work on the cooperative scheduling of heterogeneous UAVs for post-disaster missions, most solutions solve task allocation and path planning separately and ignore their strong coupling relationship. When directly applied to multiple types of discrete rescue tasks, multi-objective particle swarm optimization tends to generate a large number of infeasible solutions. Moreover, these approaches lack load balancing design, and single layer optimization cannot balance flight range efficiency and rescue timeliness. To address the above mentioned issues, this paper constructs a bi-level multi-objective optimization framework. The improved MOPSO at the upper layer realizes constrained task assignment, and PSO is adopted at the lower layer to solve single UAV paths. Meanwhile, a greedy warm start initialization strategy and a distance aware load balancing post processing strategy are introduced to take global flight range, rescue timeliness and cluster load balancing into consideration. The effectiveness of the proposed method is verified through multiple comparative simulations and ablation experiments.

2. System Model

2.1. Problem Description

This paper investigates the cooperative task planning problem for heterogeneous multi-UAVs in post-disaster emergency rescue scenarios. It is assumed that after a natural disaster occurs, three types of subtasks need to be performed at several

disaster affected points within the target area. Let the set of disaster affected points be $L=\{1,2,\cdots,L\}$, where L represents the total number of disaster affected points. Each disaster affected point $l \in L$ needs to complete three types of subtasks in sequence: reconnaissance, material delivery, and secondary confirmation. To clearly describe the relationships among subtasks, disaster affected points and task types, mapping functions are introduced. For any subtask $t \in T$, its index of affiliated disaster affected point is $l(t)=\lceil t/3 \rceil$, and its task type index is $\tau(t) \in \{1,2,3\}$. Here, $\tau(t)=1$ represents reconnaissance, $\tau(t)=2$ represents material delivery, and $\tau(t)=3$ represents secondary confirmation.

The UAV formation consists of heterogeneous UAVs. Let $K=\{1,2,\cdots,K\}$ denote the set of UAVs, which includes the reconnaissance type UAV subset K_S and the transport type UAV subset K_M. They satisfy $K_S \cup K_M = K$, as well as $K_S \cup K_M = \varnothing$ $|K_S|+|K_M|=K$. Task execution is subject to strict type matching constraints: reconnaissance ($\tau=1$) and confirmation ($\tau=3$) tasks can only be performed by reconnaissance type UAVs ($k \in K_S$), and material delivery ($\tau=2$) tasks can only be carried out by transport type UAVs ($k \in K_M$). Each UAV departs from its respective designated depot, flies at a distinct constant speed, and returns to its own depot upon completing all assigned tasks to form a complete closed loop from the depot via task points back to the depot. Furthermore, transport type UAVs have limited capacity for carrying rescue supplies, so the total amount of delivered materials must not exceed their maximum payload.

Considering the complexity of the problem, a hierarchical planning strategy is adopted to decompose the original problem into two coupled sub problems. The task allocation layer determines which UAV executes each subtask; the path planning layer determines the optimal visiting sequence of assigned disaster affected points for each UAV. These two layers are mutually coupled. The ultimate objective is to simultaneously minimize the total flight distance of the system and the maximum task completion time of the entire formation.

2.2. Mathematical Model

Under the constraint of the capability task relationship matrix, a joint optimization model for task allocation and path planning is constructed. Table 1 lists all system parameters adopted in the model together with their physical meanings.

Table 1. Definition of system parameters.

Symbol	Definition
(x_k^0, y_k^0)	Initial depot coordinates of UAV k
(x_l, y_l)	Geographic coordinates of disaster-affected point l
v_k	Constant flight speed of UAV k
Q_k	Maximum material payload of transport UAV k
q_t	Resource consumption of subtask t
t_s	Basic execution time for reconnaissance and confirmation tasks

Continued

r_m	Unloading time rate per unit material for material-delivery tasks
D_k^{max}	Maximum cruising flight distance constraint of single UAV k
$d_{k,l}^0$	Euclidean distance from the depot of UAV k to disaster-affected point l
$d_{l,l'}$	Euclidean distance between disaster-affected point l and disaster-affected point l'

To fully characterize the bi-level programming problem, two sets of decision variables are defined:

1) Task allocation variable:

$$x_{kt} = \begin{cases} 1, & \text{if subtask } t \text{ is assigned to UAV } k \\ 0, & \text{otherwise.} \end{cases} \quad \forall k \in K, \forall t \in T. \tag{1}$$

2) Path planning variables:

Node v_0^k denotes the base of UAV k, the UAV must take off from the base and eventually return to it. Let $v_k = \left\{v_0^k\right\} \bigcup \left\{l \in L \mid \exists t : l(t) = 1, x_{k,t} = 1\right\}$ be the set of nodes that UAV k needs to visit.

$$y_{k,i,j} = \begin{cases} 1, & \text{UAV } k \text{ travels from node } i \text{ to node } j \\ 0, & \text{otherwise.} \end{cases} \quad \forall k \in K, \forall i, j \in V_k, i \neq j. \tag{2}$$

Based on the above variables, the optimization objectives and constraints of this model are constructed as follows:

Objective function:

$$\begin{aligned} &(\text{F1}) \quad f_1 = \min \sum\nolimits_{k \in K} D_k \\ &(\text{F2}) \quad f_2 = \min \max_{k \in K} C_k \end{aligned} \tag{3}$$

where F1 minimizes the total flight distance to reduce the total flight energy and resource consumption of the UAV cluster and enhance system operational efficiency, while F2 minimizes the maximum task completion time to shorten the latest finish time of all tasks, optimizing the overall response speed and ensuring full coverage of rescue operations in the shortest possible time.

Constraints:

$$\text{C1}: x_{k,t} \leq \lambda_{k,\tau(t)}, \forall k \in K, \forall t \in T. \tag{4}$$

$$\text{C2}: \sum\nolimits_{k \in K} x_{k,t} = 1, \forall t \in T. \tag{5}$$

$$\text{C3}: \sum\nolimits_{t \in T} q_t x_{k,t} \leq Q_k, \forall k \in K. \tag{6}$$

$$\text{C4}: \sum\nolimits_{\substack{j \in V_k \\ j \neq i}} y_{k,i,j} = 1, \forall k \in K, \forall i \in V_k. \tag{7}$$

$$\text{C5}: \sum\nolimits_{\substack{i \in V_k \\ i \neq j}} y_{k,i,j} = 1, \forall k \in K, \forall j \in V_k. \tag{8}$$

$$\text{C6}: \sum\nolimits_{t:l(t)=1} x_{k,t} \leq \sum\nolimits_{\substack{j \in V_k \\ j \neq 1}} y_{k,i,j}, \forall k \in K, \forall l \in L. \tag{9}$$

$$\begin{aligned} \text{C7}: &T_{k_1,l} + t_s \leq T_{k_2,l}, T_{k_2,l} + r_m q_{t_2} \leq T_{k_3,l}, \forall l \in L, \\ &\forall k_1, k_2 : x_{k_1,t_1} = 1, x_{k_2,t_2} = 1, \tau(t_1) = 1, \tau(t_2) = 2 \end{aligned}. \tag{10}$$

$$\text{C8}: D_k \le D_k^{max} \quad \forall k \in K . \tag{11}$$

where C1 (task-capability matching constraint): ensures that subtask t can only be executed by UAVs equipped with corresponding functions. If UAV k cannot support task type $\tau(t)$ (*i.e.*, $\lambda_{k,\tau(t)} = 0$), then $x_{k,t}$ equals 0 to forbid invalid assignments. C2 (task uniqueness constraint): each subtask must be assigned to exactly one UAV to avoid missing or repeated execution. C3 (load capacity constraint): the total weight of supplies assigned to transport UAV k (*i.e.*, the sum of q_t for all tasks with $\tau = 2$) shall not exceed its maximum load capacity Q_k. For reconnaissance UAVs, this constraint is automatically satisfied since $Q_k = 0$ and such UAVs cannot perform tasks of $\tau = 2$. C4 and C5 (path flow conservation constraints): jointly ensure that the flight path of each UAV forms a single closed loop starting from the base, visiting all assigned nodes and finally returning to the base, without isolated nodes or sub-tours. C6 (task-path consistency constraint): if at least one subtask of disaster-affected node l is assigned to UAV k, node l must appear in the flight path of UAV k, preventing the case where tasks are allocated but the node is not visited. C7 (subtask sequence dependency constraint): it enforces that the three types of subtasks at the same disaster-affected node are completed in the order of "reconnaissance-supply delivery-secondary confirmation". Even if these subtasks are performed by different UAVs, their arrival times must satisfy the precedence logic. C8 (endurance distance constraint): the total flight distance of each UAV is restricted not to exceed its maximum range D_k^{max}, so as to guarantee flight safety and task feasibility. In the implementation of the solution in this paper, the payload constraint (C3) and range constraint (C8) are incorporated into the constraint violation value in the form of penalty functions to guide the search direction and filter feasible solutions. For the sequence dependency of reconnaissance-material delivery-secondary confirmation characterized by C7, the following treatment is adopted in the solution process. The three types of subtasks are undertaken by reconnaissance UAVs and transport UAVs separately, and each UAV flies independently according to its own visiting sequence. The makespan is defined as the maximum completion time across all UAVs, where the completion time of each UAV is the sum of its flight time and the service time of its assigned subtasks, and no inter-UAV waiting term is included. For reconnaissance and secondary confirmation subtasks at the same disaster site, duplicate visits to this site are eliminated in path planning, while the service time is accumulated separately for each subtask. If these two subtasks are executed by different UAVs, they are carried out in parallel without additional waiting time superimposed on the completion time.

3. Algorithm Design

For the post-disaster heterogeneous multi-UAV task planning model, this paper proposes a hierarchical collaborative optimization framework based on bi-level programming. The framework decomposes the complex task planning problem into an upper layer and a lower layer. The upper layer is the task allocation layer,

which adopts the improved multi-objective particle swarm optimization (MOPSO). Taking the UAV assignment of all subtasks as decision variables and the minimization of total flight distance and maximum completion time as two objectives, it solves the optimal task-UAV mapping under the constraints of task-type matching, load capacity and flight range. The lower layer is the path planning layer. For the set of target points assigned to each UAV, the standard particle swarm optimization (PSO) is used to independently solve the traveling salesman problem and optimize the visiting sequence of disaster-affected nodes to minimize the individual flight distance of each UAV (Figure 1). The two layers are connected sequentially: the allocation results from the upper layer serve as the input for the lower-layer path optimization. During the search process, the upper layer adopts a nearest-neighbor heuristic to estimate the individual path distance of each UAV, and the lower layer subsequently conducts precise PSO optimization on the visiting sequence of each UAV. There is no iterative feedback between the two layers. Furthermore, a distance-aware load-balancing post-processing mechanism is embedded between the upper and lower layers. Within each homogeneous group of UAVs, tasks are balanced according to the number of disaster sites. The ideal target number per UAV in this group is first calculated together with an upper bound on the node count, after which overloaded and underloaded UAVs are identified. The overloaded UAV selects targets to be transferred in descending order of distance from itself, and these targets are reassigned to the nearest underloaded UAV, until the target count of overloaded UAVs drops below the upper bound or all underloaded UAVs become saturated. This transfer targets task-count balancing while tending to minimize the extra flight distance. It should be noted that payload and range constraints are not rechecked after transfer. This mechanism further improves the balance and robustness of solutions. The core advantage of this hierarchical architecture is that it reduces the original high-dimensional multi-objective combinatorial optimization problem into multiple low-dimensional subproblems, greatly decreasing the complexity of the search space while balancing global optimality and local accuracy.

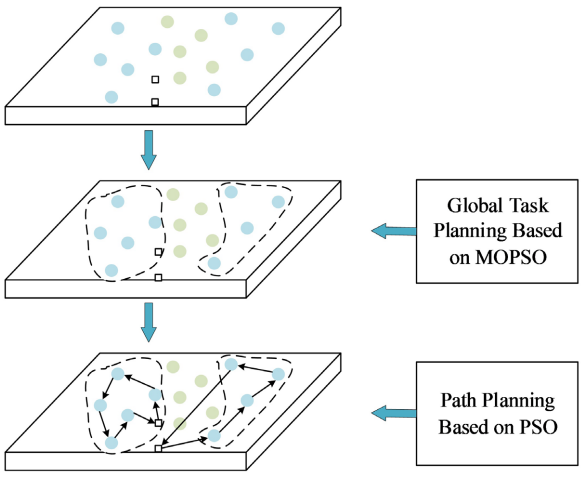


Figure 1. Bi-level programming framework.

3.1. Upper-Level MOPSO Model

The post-disaster heterogeneous multi-UAV task allocation problem has the following characteristics: the decision variables correspond to discrete task-UAV assignment relationships; strict task-type matching and load capacity constraints are involved; and two conflicting objectives, namely total flight distance and make span, are considered. Direct application of standard MOPSO will suffer from a large number of infeasible solutions, low search efficiency and slow convergence. To tackle these issues, targeted improvements are proposed from five aspects: encoding strategy, initialization method, update mechanism, multi-objective guidance strategy and fitness evaluation, enabling the algorithm to efficiently solve this combinatorial multi-objective optimization problem.

3.1.1. Encoding and Constraint Embedding Mechanism

This paper adopts a hard-constraint embedding strategy to directly repair infeasible solutions at the encoding layer. Let the index range of reconnaissance UAVs be $[0, N_S - 1]$, and the index range of transport UAVs be $[N_S, N_U - 1]$. For subtask j, its type $k(j)$ determines the feasible UAV set $U_{legal}(j)$. The particle position x_j is a real number, which is mapped to a valid UAV index via the following transformation:

$$u_j = \begin{cases} \lfloor |x_j| \rfloor \bmod N_S, & k(j) \in \{0, 2\} \\ N_S + \left(\lfloor |x_j| \rfloor \bmod N_V \right), & k(j) = 1 \end{cases}. \tag{12}$$

This mapping is enforced during the decoding phase to ensure that the assignment scheme corresponding to each particle always satisfies the capability-matching constraint. The search space of the population is compressed within the feasible region, thereby eliminating invalid searches completely.

3.1.2. Greedy Warm-Start Strategy

The initial population of standard MOPSO is usually generated in a fully random manner. For the task allocation problem, the dimension of decision variables $N_M = 3N_T$ can reach hundreds. Random particles are uniformly distributed in such a high-dimensional space and stay far away from the optimal solution, requiring numerous iterations to enter promising regions. If heuristic domain knowledge of the problem can be exploited to generate high-quality initial solutions, the convergence speed can be significantly accelerated. The steps are described as follows:

1) Traverse all subtasks $j \in M$, and determine the candidate UAV set according to its type $k(j)$.

2) For supply delivery tasks ($k = 1$): among transport UAVs, select the UAV closest to disaster-affected node $t(j)$ with sufficient remaining load capacity. If no UAV satisfies the payload constraint, the transport UAV closest to the disaster site is selected.

3) For reconnaissance and confirmation tasks: select the reconnaissance UAV

closest to the disaster-affected node. Specifically, for confirmation tasks, priority is given to assigning them to the same UAV that performs the reconnaissance task at the same disaster-affected node, so as to reduce the back-and-forth flight of UAVs among different disaster-affected nodes.

4) Generate a complete assignment scheme x_{heu} according to the above rules.

5) Treat this scheme as one particle in the initial population. Meanwhile, perform random mutation on it by randomly changing the assignments of a small number of subtasks to obtain another particle for enhancing initial diversity.

3.1.3. Probabilistic Learning Mechanism

The concept of continuous velocity is discarded, and a probabilistic learning mechanism is adopted to update the value of each dimension of particles. For the j-th dimension of particle i (corresponding to subtask j), its value $x_j^{(t+1)}$ in the next generation is determined by the following rules:

$$x_j^{(t+1)} = \begin{cases} pbest_{i,j}, & \text{if } r_1 < P_{learn} \\ leader_j, & \text{if } P_{learn} \le r_1 < P_{learn} + l_{learn} \\ \text{Random mutation}\left(\text{uniformly sample from feasible UAV set}\right), & \text{if } r_2 < P_{mutate} \\ x_j^{(t)}, & \text{otherwise} \end{cases} \tag{13}$$

where $r_1, r_2 \sim U(0,1)$, P_{learn} denotes the probability of learning from personal historical optimum, l_{learn} denotes the probability of learning from the global leader, and P_{mutate} denotes the mutation probability. Typical parameter settings are $P_{learn} = 0.4$, $l_{learn} = 0.4$, $P_{mutate} = 0.1$.

This update scheme directly operates on discrete UAV indices. Each dimension independently selects values from the personal best, global leader or random mutation, maintaining the propagation of high-quality information fragments.

3.1.4. Leader Selection and External Archive Management

Multi-objective optimization maintains a set of non-dominated solutions (Pareto front) rather than a single optimal solution. Standard MOPSO employs an external archive to store non-dominated solutions, from which leaders are selected. Improper archive management may result in uneven distribution of the front or loss of extreme solutions. A simplistic leader selection strategy causes the population to converge to a local region of the Pareto front and lose diversity. The algorithm maintains an external archive A with capacity N_{arc}. After each iteration, all non-dominated solutions in the current population are inserted into A, and dominated solutions are eliminated. When the size of A exceeds N_{arc}, the archive is pruned based on crowding distance: calculate the crowding distance of each solution in the objective space and preferentially remove solutions with the smallest crowding distance to preserve solutions in sparsely distributed regions of the Pareto front.

Leader selection: A crowding-distance-based roulette-wheel mechanism is adopted. The probability that solution $a \in A$ is selected as the global leader is proportional to its crowding distance $d_{crowd}(a)$:

$$P(a)=\frac{d_{crowd}(a)}{\sum_{b\in A} d_{crowd}(b)}. \tag{14}$$

Solutions with larger crowding distances (*i.e.*, those located in sparse regions of the Pareto front) have higher chances to be selected as leaders, thereby guiding the population to explore these insufficiently searched regions.

3.1.5. Final Solution Selection Strategy

The multi-objective optimization algorithm outputs a Pareto solution set. In practical applications, however, decision-makers need to select one final implementation scheme from it. The selection principle should first satisfy all hard constraints, and then choose among feasible solutions according to practical preferences.

This paper proposes a two-stage selection strategy with constraint priority: For each solution in archive A, calculate its total constraint violation ϕ:

$$\phi=\sum_{u\in U_V}\max\left(0,\sum_j x_{j,u}r_j-c_u\right)+\sum_{u\in U}\max\left(0,L_u-D_{\max}\right) \tag{15}$$

where L_u is estimated via the nearest-neighbor method. Given a threshold $\varepsilon>0$, filter out solutions satisfying $\phi\le\varepsilon$ from A. If this set is non-empty, select the solution with the minimum total flight distance f_1 as the final output. If no solution meets $\phi\le\varepsilon$, pick the solution with the smallest ϕ (*i.e.*, the one with the lightest constraint violation), and prompt the decision-maker to adjust the parameters.

3.1.6. Overall Algorithm Flow

The flow of the improved upper-level MOPSO algorithm is presented as follows:

1) Parameter initialization: set population size N_p, external archive capacity N_{arc}, learning probabilities p_{learn}, l_{learn}, and mutation probability p_{mutate}.

2) Population initialization: generate one high-quality solution and its mutated copy using the greedy warm-start strategy, while the remaining particles are randomly generated.

3) Fitness evaluation: for each particle, estimate the path distance of each UAV and calculate objectives f_1 and f_2.

4) Personal best update: update the personal historical optimum pbest of each particle according to the Pareto dominance relationship.

5) External archive update: insert non-dominated solutions of the current population into the archive and prune the archive to its maximum capacity.

6) Leader selection: select a global leader for each particle from the archive via the crowding-distance-based roulette-wheel mechanism.

7) Particle position update: update each dimension of every particle following the probabilistic learning mechanism.

8) If the termination criterion (maximum iteration number or convergence criterion) is not satisfied, go back to Step 3; otherwise, proceed to final solution selection.

9) Apply the constraint-priority two-stage selection strategy to the final archive and output the recommended task allocation scheme.

3.2. Lower-Level PSO Model

Given the task allocation results obtained from the upper layer, *i.e.*, each UAV is assigned a set of disaster-affected nodes to visit, the access sequence of disaster-affected nodes is optimized independently for each UAV to minimize the flight distance of a single UAV and form a closed loop of start node-task nodes-start node. This problem is essentially the classic Traveling Salesman Problem (TSP). The standard particle swarm optimization algorithm is adopted to solve this problem.

For the u-th UAV, let the set of disaster-affected nodes to be visited be $\tau_u = \{t_1, t_2, \cdots, t_n\}$, where $n = |T_u|$. Construct an $(n+1)\times(n+1)$ distance matrix including the starting point:

$$D_{0,i}^{(u)} = D_{i,0}^{(u)} = d_{u,t_i}, i = 1,2,\cdots,n. \tag{16}$$

$$D_{i,j}^{(u)} = d_{t_i,t_j}, i,j = 1,2,\cdots,n \tag{17}$$

where $d_{u,t}$ denotes the Euclidean distance from UAV u to disaster-affected node t, and d_{t_i,t_j} represents the Euclidean distance between disaster-affected nodes t_i and t_j. Index 0 refers to the starting point of the UAV.

The objective for each UAV is to minimize the total length of the closed loop containing the starting point:

$$L_u = D_{0,\pi_1}^{(u)} + \sum_{k=1}^{n-1} D_{\pi_k,\pi_{k+1}}^{(u)} + D_{\pi_n,0}^{(u)}. \tag{18}$$

Perform the above PSO path planning sequentially for all N_U UAVs, and the specific procedure is as follows:

1) Task aggregation: traverse the assignment results of all subtasks and collect the set T_u of disaster-affected nodes to be visited for each UAV u;

2) UAV-wise planning: for each UAV $u \in U$;

3) If $T_u = \varnothing$, mark it as no task and set its path distance to 0. Otherwise, construct the distance matrix and initialize the PSO population;

4) Carry out PSO iterative optimization to obtain the optimal path π^* and the shortest distance L_u^*;

5) Summary output: accumulate the flight distances of all UAVs to obtain the total system flight distance. After path optimization, each UAV executes its assigned subtasks sequentially along its access sequence. The completion time is calculated as the sum of flight time and the service time of all assigned subtasks. When a single UAV undertakes multiple subtasks at the same disaster site, this site is recorded only once in the path to avoid repeated visits, while the service times of individual subtasks are accumulated sequentially. Subtasks at the same disaster site performed by different UAVs are executed in parallel, with no inter-UAV waiting delay introduced.

4. Simulation Experiments and Analysis

4.1. Model Simulation Results

The experiments are conducted in a two-dimensional plane of 1000 × 1000. A

total of 50 disaster-affected nodes are randomly generated, whose coordinates are uniformly distributed within the range [100, 1000]. Each disaster-affected node contains three types of subtasks: reconnaissance, material delivery and secondary confirmation, resulting in 150 subtasks in total. The resource consumption of the three types of subtasks is randomly generated from integer intervals [50, 200], [2, 12] and [10, 50], respectively. The basic execution time of reconnaissance and confirmation tasks is 5 minutes. The execution time of material delivery tasks is proportional to the material quantity, with an unloading rate of 0.5 minutes per unit.

The UAV fleet consists of 14 heterogeneous UAVs, including 8 reconnaissance UAVs (IDs 0 - 7) and 6 transport UAVs (IDs 8 - 13). The starting positions of all UAVs are randomly distributed within [0, 200], and their flight speeds are randomly generated in the range of 0.08 to 0.16 m/s. Each transport UAV has an individual load capacity limit: 5, 10, 15, 8, 12 and 6 units, respectively. Reconnaissance UAVs have zero load capacity. Restricted by battery endurance, the maximum flight range of each UAV is set to 3500 m. The initial spatial distribution of UAVs and disaster-affected nodes is shown in Figure 2.

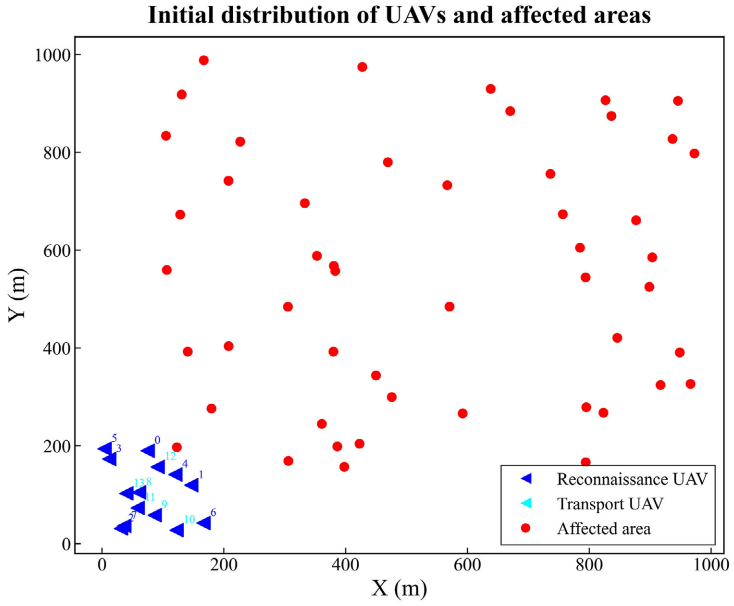


Figure 2. Initial distribution of UAVs and disaster-affected nodes.

The upper-level MOPSO algorithm adopts a population size of 80, a maximum iteration number of 120, and an external archive capacity of 40. The particle learning probability is 0.15, and the mutation probability is 0.10. The algorithm performs multi-objective sorting based on the Pareto dominance relationship, and selects leaders via the crowding-distance-based roulette-wheel mechanism. The population size and iteration number of the lower-level PSO for path planning are adaptively adjusted according to the target number n of each UAV: the population size is min (50, 10 n), and the maximum iteration number is max (50, 10 n). The

inertia weight $w = 0.8$, and the learning factors $c_1 = c_2 = 1.5$. In the load-balancing post-processing stage, each reconnaissance UAV and each transport UAV are assigned at most 12 disaster-affected nodes.

Figure 3 presents the visualization results of multi-UAV cooperative task assignment, where different colors denote the flight paths of UAVs. The visualization demonstrates that the UAVs can achieve favorable task assignment and path planning under the given constraints.

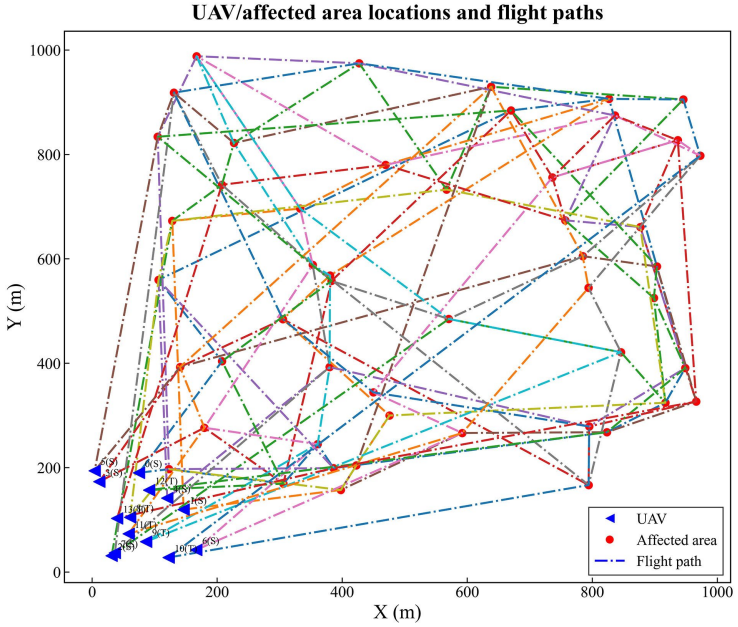


Figure 3. UAV task assignment and flight path planning.

4.2. Performance Metrics and Comparative Analysis

To comprehensively evaluate the algorithm performance, the following metrics are defined: Total flight distance, the sum of flight path lengths of all UAVs. Each UAV departs from the starting point, sequentially visits all assigned target nodes and finally returns to the start, and the total length of this closed route is calculated. Makespan, the maximum time consumed by any UAV to finish all assigned tasks and return to its starting point. Load balance degree, the standard deviation of the task quantity for each UAV, which reflects the workload balance among UAVs. UAV utilization rate, the percentage of UAVs assigned at least one subtask among the total fleet.

In the tests, the proposed bi-level programming model is compared with four baseline algorithms:

K-means + PSO: First, K-means clustering divides the 50 disaster-affected nodes into 14 clusters (consistent with the number of UAVs). Each cluster is mapped to reconnaissance or transport UAVs according to material demand within the cluster, and PSO is then adopted for path planning. This method verifies the combined performance of clustering-based assignment and precise path

planning.

Single-layer MOPSO: Without hierarchical strategy, MOPSO directly handles the 150-dimensional task assignment problem. The path distance is estimated by the nearest-neighbor heuristic, and no accurate second-layer path optimization is performed. This case demonstrates the superiority of the hierarchical strategy over the single-layer solution.

CP-SAT + PSO: The OR-Tools CP-SAT solver is used to formulate and solve the integer programming model for task assignment, with constraints including task-type matching and load capacity limits. PSO is subsequently applied for path planning. This comparison evaluates the difference between exact solvers and heuristic algorithms.

Greedy + PSO: A greedy nearest assignment strategy is adopted. Traverse all subtasks and sequentially assign each subtask to the nearest UAV that satisfies the type-matching constraint, followed by PSO path planning. This test reveals the performance gap between local greedy strategies and global optimization strategies.

Table 2. Performance comparison of different algorithms under the same scenario.

Algorithm	Total flight distance	Makespan	Load balance degree(σ)	UAV utilization rate
Proposed algorithm	29236.2	29735.7	3.7	100%
K-means + PSO	31623.2	37528.8	4.2	100%
Single-layer MOPSO	44983.5	44006.5	4.6	100%
CP-SAT + PSO	27091.6	102660.9	18.5	64%
Greedy + PSO	27702.5	111684.2	18.5	64%

Table 2 verifies the comparison results of performance metrics for the five algorithms in the given scenario. The data indicate that in terms of total flight distance, the proposed algorithm yields a value of 29236.2, ranking third among all algorithms. This value is higher than those of CP-SAT + PSO (27091.6) and Greedy + PSO (27702.5). Nevertheless, both CP-SAT + PSO and Greedy + PSO achieve a UAV utilization rate of merely 64%, with five idle UAVs for each method, which implies poor engineering feasibility. Compared with K-means + PSO with the same UAV utilization rate, the proposed algorithm reduces the flight distance by 7.6%.

In addition, the makespan of the proposed algorithm is 29735.7, which is significantly better than CP-SAT + PSO (102660.9, a 71% improvement) and Greedy + PSO (111684.2, a 73% improvement), and 20.8% lower than K-means + PSO. This demonstrates that the proposed algorithm has obvious advantages in balancing the task load of each UAV. The load balance degree of the proposed algorithm is 3.7, the lowest among all algorithms, indicating the most uniform task allocation across UAVs. By contrast, CP-SAT + PSO and Greedy + PSO reach up to

18.5, suffering from severe task concentration.

For UAV utilization rate: the proposed algorithm, K-means + PSO and single-layer MOPSO all achieve 100% UAV utilization rate, while CP-SAT + PSO and Greedy + PSO only use 64% of the UAVs, leaving 5 UAVs idle. It should be noted that the objective definitions of the comparative methods are not entirely consistent. CP-SAT + PSO takes the linear assignment cost as its optimization objective instead of total flight distance or makespan, and its favorable distance performance is achieved at the expense of task balance and UAV utilization rate. By contrast, the proposed algorithm is formulated as a bi-objective optimization problem minimizing total flight distance and makespan, thus possessing obvious advantages in makespan and load balance.

To clarify the differences among the comparative methods in optimization objectives, constraint handling and post-processing, Table 3 summarizes the solution protocols and computational configurations of the five approaches.

Table 3. Comparison of solution protocols and computational configurations for compar ative methods.

Method	Optimization objective in allocation stage	Path optimization	Payload/range constraint handling	Load balancing post-processing	Computational budget
Proposed (Bi-layer MOPSO + PSO)	Bi-objective: total flight distance and makespan (the upper layer adopts nearest-neighbor estimation, and the standard deviation of task count multiplied by 800 is incorporated into the distance term)	Precise single-UAV TSP optimization via PSO in the lower layer	Soft constraints: penalty × 10 for payload overload, penalty × 100 for material shortage, penalty × 50 for range violation	Yes (task-count balancing within homogeneous UAV group with distance-aware transfer)	MOPSO: population size 80, iterations 120, archive size 40; PSO: population size min(50,10n)
K-means + PSO	Minimize the sum of squared intra-cluster distances via K-means clustering	Single-UAV TSP via PSO in the lower layer	Payload and range are not checked	No	Iterations for K-means convergence; PSO settings are the same as above
Single-layer MOPSO	Same bi-objective as the upper layer of the proposed method (nearest-neighbor estimation, no precise path calculation)	None (no layering, no path planning)	Payload and range are not checked	No	Population size 80, iterations 120, archive size 40
CP-SAT + PSO	Linear assignment cost: $\min \sum x \cdot \text{round}\left(2 \cdot d_{u,t}\right)$, with a weight coefficient of 1.5 for material tasks	Single-UAV TSP via PSO in the lower layer	Hard constraint: $\sum(\text{Demand} \cdot x) \leq$ payload upper bound; no range constraint	No	CP-SAT time limit: 60 s, 4 threads
Greedy + PSO	Greedily assign each task to the nearest UAV with compatible type	Single-UAV TSP via PSO in the lower layer	Soft constraint: UAV with sufficient payload is prioritized; the nearest UAV is selected if unavailable; no range constraint	No	Deterministic assignment (no iteration); PSO settings are the same as above

4.3. Parameter Sensitivity Analysis

To verify the effectiveness of each improvement to the MOPSO algorithm, ablation experiments are designed by removing each improved module one by one. With other conditions unchanged, the variation of algorithm performance is observed. All ablation variants adopt the same initialization strategy to fairly evaluate the contribution of each improvement under identical search starting points. Six comparative variants are set in the experiments, and each variant corresponds to the removal of one improvement module: Full MOPSO: equipped with all five improvements, including greedy warm start, embedded encoding constraints, probability learning, crowding-distance archive management, and final selection strategy. No warm start: the greedy warm start is removed, and the initial population is fully randomly generated while other improvements are retained. No encoding embedding: the forced mapping of encoding-layer constraints is eliminated, and penalty functions are adopted to handle type-matching constraints instead. No probability learning: the probability learning mechanism is removed, and the standard continuous velocity-position update formula of PSO is applied. No archive management: crowding-distance-guided leader selection is discarded, and leaders are randomly selected from the archive. No final selection strategy: the constraint-first two-stage final selection strategy is removed, and solutions are randomly picked from the archive.

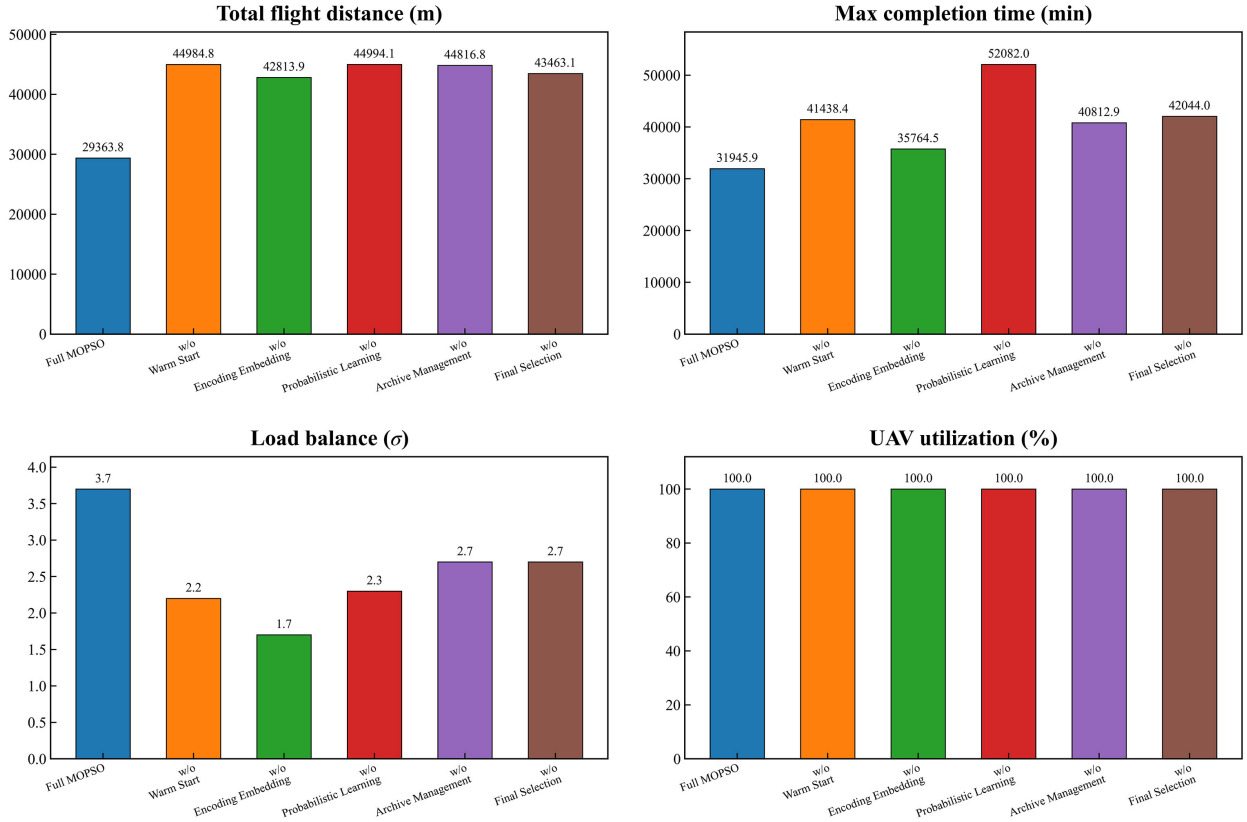


Figure 4. Ablation experiments.

Figure 4 systematically evaluates the effect of each mechanism on the optimization performance of multi-UAV task assignment by disabling key modules item by item. The results show that removing any module leads to a notable increase in total flight distance, among which the greedy warm start, probability learning mechanism and crowding-distance archive management exert the most significant impacts. The analysis reveals that for high-dimensional combinatorial optimization problems, high-quality initial solutions are more critical than fine-tuning the search algorithm. Embedding constraints directly at the encoding layer can effectively avoid interference from infeasible solutions and improve computational efficiency. The probability learning mechanism designed for discrete spaces overcomes information distortion during the continuous-to-discrete mapping of standard PSO and greatly improves load balancing. Leader selection based on crowding distance is essential for maintaining population diversity and preventing premature convergence. Furthermore, the constraint-first final selection strategy guarantees that the obtained solutions satisfy practical engineering constraints and prevents the selection of infeasible candidates from the Pareto front. In summary, these mechanisms work synergistically to support the efficient optimization capability of the algorithm in high-dimensional, multi-constrained and multi-objective scenarios.

5. Conclusions

Aiming at the complex combinatorial optimization problem of heterogeneous multi-UAV cooperative task planning for post-disaster emergency rescue scenarios, this paper constructs a hierarchical task assignment and path collaborative optimization model based on bi-level programming. The original problem is decomposed into two interrelated subproblems: upper-level task assignment and lower-level path planning, which balances global optimization and local accuracy. To solve this model, an improved multi-objective particle swarm optimization (MOPSO) algorithm is proposed. With five improvements including greedy warm start, embedded encoding constraints, probability learning, crowding-distance archive management and constraint-first final selection strategy, the search efficiency and solution quality of the algorithm in the high-dimensional discrete solution space are effectively enhanced.

Experimental results demonstrate that, in multi-task post-disaster scenarios, compared with four baseline methods including K-means + PSO, single-layer MOPSO, CP-SAT + PSO and Greedy + PSO, the proposed method achieves outstanding performance in terms of makespan and load balance. Its total flight distance is significantly reduced relative to K-means + PSO, and ablation experiments further verify the effectiveness of each improvement. At present, the model focuses on task planning under static scenarios, assuming that all task information and environmental parameters are known and unchanged during the planning phase, which brings certain limitations. Future work will mainly introduce dynamic task planning mechanisms to respond to unexpected events such as newly

added tasks, UAV failures and environmental changes during flight, so as to realize real-time replanning.

Author Contributions

Conceptualization, Ziye Shi and Wenxing Wu; methodology, Ziye Shi; software, Ziye Shi; validation, Ziye Shi, Wenxing Wu, and Liqin Tian; formal analysis, Ziye Shi; investigation, Ziye Shi; resources, Liqin Tian; data curation, Ziye Shi; writing—original draft preparation, Ziye Shi; writing—review and editing, Wenxing Wu; visualization, Ziye Shi; supervision, Liqin Tian; project administration, Liqin Tian; funding acquisition, Liqin Tian. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported by the Fundamental Research Funds for the Central Universities (Grant No. ZY20250106); Hebei Major Science and Technology Support Program (262Y5403D); Hebei Provincial Innovation Capacity Enhancement Program (25350802D); Langfang Municipal Science and Technology Support Program (2025011099).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Lu, Y.C., Zhou, K.Y. and Zhang, J.J. (2023) Development Status of UAV and Its Application in Aviation Emergency Rescue. *Chinese Medical Equipment Journal*, **44**, 108-113. (In Chinese)

[2] Ouyang, Q., Wu, Z., Cong, Y. and Wang, Z. (2023) Formation Control of Unmanned Aerial Vehicle Swarms: A Comprehensive Review. *Asian Journal of Control*, **25**, 570-593. https://doi.org/10.1002/asjc.2806

[3] Guo, C.Y., Li, Y.F., Mo, Z.L., *et al.* (2026) Review on Cooperative Control Technology of Low Altitude UAV Formation. *Journal of Chongqing University of Posts and Telecommunications* (*Natural Science Edition*), **38**, 630-648. (In Chinese)

[4] Ning, C., Fan, J. and Sun, S.K. (2025) Review of Multi-UAV Collaborative Planning Research. *Computer Engineering and Applications*, **61**, 42-58. (In Chinese)

[5] Zhang, Y., Xie, S., Zhang, L., Zhang, J., *et al.* (2017) Optimal Task Decision-Making for Heterogeneous Multi-UAV Cooperation Reconnaissance. *Journal of Northwestern Polytechnical University*, **35**, 385-392. (In Chinese)

[6] Duong, M.P. and Phuc, Q.H. (2021) Safety-Enhanced UAV Path Planning with Spherical Vector-Based Particle Swarm Optimization. *Applied Soft Computing*, **107**, Article 107376. https://doi.org/10.1016/j.asoc.2021.107376

[7] Zhang, X.Y., Xia, S. and Zhang, T. (2023) Adaptive Genetic Learning Particle Swarm Optimization Based Cooperative Task Allocation for Multi-UAVs. *Control and Decision*, **38**, 3103-3111. (In Chinese)

[8] Wang, J., Jia, G., Lin, J. and Hou, Z. (2020) Cooperative Task Allocation for Heterogeneous Multi-UAV Using Multi-Objective Optimization Algorithm. *Journal of Cen-*

tral South University, **27**, 432-448. https://doi.org/10.1007/s11771-020-4307-0

[9] Wang, J.F., Jia, G.W., Xin, H.B., *et al.* (2024) Joint Optimization for Heterogeneous Multi-UAV Configuration and Mission Planning within SEAD Scenarios. *Journal of National University of Defense Technology*, **46**, 32-41. (In Chinese)

[10] Mavrovouniotis, M., Li, C. and Yang, S. (2017) A Survey of Swarm Intelligence for Dynamic Optimization: Algorithms and Applications. *Swarm and Evolutionary Computation*, **33**, 1-17. https://doi.org/10.1016/j.swevo.2016.12.005

[11] Fu, Y., Ding, M. and Zhou, C. (2012) Phase Angle-Encoded and Quantum-Behaved Particle Swarm Optimization Applied to Three-Dimensional Route Planning for UAV. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, **42**, 511-526. https://doi.org/10.1109/tsmca.2011.2159586

[12] Yu, J., Yong, E.M., Chen, H.Y., *et al.* (2022) Bi-level Mission Planning Method for Multi Cooperative UAV Air to Ground Attack. *Systems Engineering and Electronics*, **44**, 2849-2857. (In Chinese)

[13] Wen, J., Wang, F. and Su, Y. (2025) A Bi-Layer Collaborative Planning Framework for Multi-UAV Delivery Tasks in Multi-Depot Urban Logistics. *Drones*, **9**, Article 512. https://doi.org/10.3390/drones9070512

[14] Han, Z., Zheng, G., Li, C. and Ji, B. (2026) Time-Window-Constrained Computation Offloading via Cost-Efficient Multi-UAV Scheduling in Space-Air-Ground Integrated Networks. *Physical Communication*, **75**, 103042. https://doi.org/10.1016/j.phycom.2026.103042

[15] Coello, C.A.C., Pulido, T.G., Lechuga, S.M. (2004) Handling Multiple Objectives with Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, **8**, 256-279.

[16] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002) A Fast and Elitist Mult Objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**, 182-197. https://doi.org/10.1109/4235.996017

[17] Wang, Y., Ma, C.R. and Zhao, M. (2025) Collaborative Multi-Task Assignment of Heterogeneous UAVs Based on Hybrid Strategies Based Multi-Objective Particle Swarm. *Journal of Zhejiang University: Engineering Science*, **59**, 821-831. (In Chinese)

[18] Xiao, P., Yu, H.X., Huang, L., *et al.* (2025) 3D Path Planning for Unmanned Aerial Vehicle Based on MDEPSO Algorithm. *Acta Armamentarii*, **46**, Article 240710.

[19] Yu, Q. and Xing, G. (2021) Application and Development of UAV in Emergency Rescue in Our Country. *Occupational Health and Emergency Rescue*, **39**, 350-355. (In Chinese)

[20] Wang, K., Cao, T.L., Li, X.J., *et al.* (2025) A Survey on Trajectory Planning and Resource Allocation in Unmanned Aerial Vehicle Assisted Edge Computing Networks. *Journal of Electronics & Information Technology*, **47**, 1266-1281. (In Chinese)

[21] Gan, L.Q., Dong, C., Wang, W., *et al.* (2026) Intent Driven UAV Swarm Network for Emergency Disaster Relief. *Acta Aeronautica et Astronautica Sinica*, **47**, Article 533471.