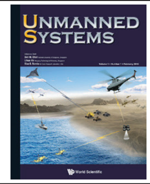
Unmanned Systems, Vol. 14, No. 3 (2026) 589–604

DOI: 10.1142/S2301385026410037

Unmanned Systems
https://www.worldscientific.com/worldscinet/us



Post-Disaster Multi-UAV Task Planning Based on Graph Neural Network Decoupling

Rui Hong [*,†], Jia Zhang [*,†,§], Minggang Gan [*,†], Qing Wang [*,†], Bin Xin [*,†], Jie Chen [†,‡]

[*] *School of Automation, Beijing Institute of Technology, Beijing 100081, P. R. China*

[†] *State Key Laboratory of Autonomous Intelligent Unmanned Systems, Beijing Institute of Technology, Beijing 100081, P. R. China*

[‡] *Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, P. R. China*

This paper addresses multiple unmanned aerial vehicles (UAVs) task planning for post-disaster supply delivery, a problem that comprises two coupled multi-objective optimization problems (MOPs): task allocation and path planning. These sub-problems are highly coupled, which therefore requires a layered decoupling strategy. To this end, a multi-UAV task planning method that combines graph neural network (GNN) with an improved multi-objective particle swarm optimization (MOPSO) algorithm is proposed in this paper. A GNN is employed to model the complex relationships between UAVs and tasks, extracting node and graph embeddings through message-passing mechanisms to provide shared, high-quality global prior information. A multi-objective optimization model is constructed for task allocation and operation planning. Given its high-dimensional, nonlinear and strongly coupled characteristics, an MOPSO algorithm with adaptive dynamic weight is designed to enhance convergence efficiency. Simulations show the proposed method outperforms benchmark methods in task completion rate (TCR), task completion time (TCT), energy consumption and MOP performance metrics across varying scales.

Keywords: Multi-UAV task planning; multi-objective optimization; graph neural network; dynamic inertia weight; multi-objective particle swarm optimization.

1. Introduction

Natural disasters, characterized by their sudden onset, wide impact and destructive power, pose a serious threat to public safety and infrastructure. Post-disaster rescue efforts often face challenges like paralyzed transportation and disrupted communications, which limit the efficiency of traditional rescue methods [1]. Post-disaster emergency supply delivery tasks are typically characterized by tight time constraints, large-scale material demand and numerous dispersed rescue points. Multiple unmanned aerial vehicles (UAVs), leveraging their rapid response and high maneuverability, offer a key solution to overcoming terrain restrictions and achieving precise supply delivery [2–6].

Multi-UAV cooperative task planning is a problem for optimizing task execution efficiency and overall system performance. It achieves this by coordinating a fleet of UAVs to allocate tasks effectively, plan flight trajectories and avoid conflicts and obstacles. This process comprises two core, and interdependent, components: task allocation and operation planning [7], both of which fall into the category of Nondeterministic Polynomial (NP) problems.

At present, research on multi-UAV cooperative task planning algorithms is primarily directed along two lines: centralized approaches and hierarchical decoupling

Received 30 October 2025; Accepted 2 December 2025; Published 8 January 2026. This paper was recommended for publication in its revised form by Special Issue Editors, Jie Chen, Ben M Chen, and Zongli Lin.
Email Address: [§] *zhangjia@bit.edu.cn*
[§] Corresponding author.

approaches [8]. For centralized approaches, Gharrad *et al.* [9] addressed task allocation as a sub-problem of path planning to achieve joint planning for multiple UAVs. Liu *et al.* [10] employed a multi-agent reinforcement learning (RL) method to solve task allocation and path planning simultaneously. Wang *et al.* [11] proposed a unified framework that integrates task pre-planning and replanning. Centralized approaches are suitable for scenarios requiring high coordination, as they can effectively avoid task conflicts and improve resource utilization. However, they suffer from high computational complexity, leading to challenges such as low computational performance and response delays in large-scale problems.

For hierarchical decoupling approaches, Yao *et al.* [12] developed a two-layer planning model for UAV delivery sequencing and cooperative trajectory planning. Zhao *et al.* [13] proposed a RL framework comprising a centralized task allocation module and a multi-agent path planning module. Li *et al.* [14] introduced a hierarchical task planning framework with task-priority awareness. Hierarchical decoupling approaches simplify problem complexity and are well-suited for scenarios with high real-time requirements or limited computational resources. However, these methods often weaken or ignore the coupling between the sub-problems of task allocation and operation planning [15]. Solving these sub-problems separately may lead to sub-optimal solutions or task execution conflicts. Therefore, designing effective hierarchical decoupling strategies remains a pressing challenge.

Based on the cooperative control structure of multiple UAVs, the methods for solving task allocation problems can be divided into centralized and distributed approaches [16]. Centralized task allocation is managed by a central control system that makes decisions by collecting all information for global optimization. Ghassemi and Chowdhury [17] and Lindsay *et al.* [18] computed globally optimal solutions for multi-agent task allocation problems using integer programming methods. Gao *et al.* [19] and Han *et al.* [20] utilized heuristic optimization algorithms to accelerate the convergence speed for solving multi-UAV task allocation problems. However, when the task scale is large, the computational complexity and load of centralized methods increase significantly, potentially leading to decision delays.

In distributed task allocation, UAVs possess autonomous decision-making capabilities and collaborate with other UAVs to complete tasks. Xiong *et al.* [21] and Dong *et al.* [22] implemented task allocation based on market mechanisms, enhancing solution stability and robustness. Wang *et al.* [23] proposed a distributed ant colony algorithm, enabling UAVs to cooperatively select tasks to ensure solution optimality. Distributed approaches improve system reliability, robustness and scalability. However, they are limited by local-information-based decision-making, making it difficult to achieve global optimization.

With the development of RL, multi-UAV task allocation based on RL has become a research hotspot in recent years. Zhu and Fang [24] proposed a semi-random Q-learning algorithm that improves solution efficiency and quality by adjusting action selection probabilities. Peng *et al.* [25] introduced an evolutionary deep Q-learning algorithm to address the uncertainty in the optimal solution space for task allocation and to enhance convergence rate. RL continuously optimizes decision-making strategies through interaction with the environment to cope with complex scenarios. However, RL-based methods face challenges such as high training cost, large computational complexity and limited model scalability.

Operation planning involves formulating executable action sequences from an initial state to a target state, including path planning and execution action planning. Algorithmic research for UAV operation planning is mainly divided into exact algorithms and approximate algorithms [26]. Exact algorithms are typically used for path planning in deterministic environments. Du [27] and Yoo and Moon [28] proposed an improved A^* algorithm to generate the shortest collision-free paths for UAV swarms. Jabbar *et al.* [29] introduced a weight-adjustable Dijkstra algorithm to find the shortest path. These methods can obtain optimal solutions, but their computational cost grows exponentially with the problem size, making them suitable only for small-scale scenarios.

Approximate algorithms mainly use heuristic methods to find near-optimal solutions. Chen *et al.* [30] applied an adversarial-learning-based artificial bee colony algorithm to improve path planning efficiency. Ali *et al.* [31] and Niu *et al.* [32] employed improved ant colony optimization algorithms to address collision avoidance and coordination among multi-UAVs, while enhancing robustness and global convergence speed. Bouaziz *et al.* [33], Rosas-Carrillo *et al.* [34] and Liu *et al.* [35] leveraged the strong global search capability of improved particle swarm optimization to increase the coverage of globally optimal paths. Li *et al.* [36] proposed an algorithm framework based on smoothed particle hydrodynamics to address complex obstacle environments. The advantage of approximate algorithms lies in finding near-optimal solutions with lower computational cost. However, relying solely on path planning or motion planning is often insufficient to address the demands of large-scale and complex task coordination.

Focusing on post-disaster emergency supply delivery, this study explores multi-UAV cooperative task planning in scenarios with multiple objectives and constraints. To tackle the complexity, we build a hierarchical framework and propose an decoupling algorithm that leverages deep learning for feature extraction and optimization techniques for precision search, thereby ensuring the solution efficiency. The contributions of this paper are as follows:

- A hierarchical task planning strategy based on graph neural network (GNN) decoupling is proposed. This strategy divides the task planning problem into three stages of weakly coupled problems during the solution procedure, reducing the difficulty of solving the problem.
- Global scenario features such as spatial adjacency of task points, task priority hierarchies and cooperative relationships among UAVs are extracted in real time using GNN. This not only improves the mastery of global information in the solving process, but also provides high-quality initial solutions for task allocation, thereby accelerating the overall optimization process.
- The improved MOPSO algorithm introduces dynamic inertia weights and adapts them to the task allocation and operation planning problems in this study. The improvement effectively balances the global exploration and local development capabilities of the algorithm, and improves the multi-objective optimization performance of task planning.

The remainder of this paper is organized as follows: Section 2 conducts problem formulation; Section 3 introduces a task planning method based on GNN decoupling; Section 4 presents the comparative results; Section 5 concludes the paper.

2. Multi-UAV Cooperative Task Planning Problem Formulation

2.1. *Problem description*

Emergency supply delivery in post-disaster urban environments is highly time-critical, requiring a rapid multi-UAV response. This necessitates a coordinated approach that accounts for individual UAV capabilities, task time windows, inter-UAV collision avoidance and urban operational constraints, to effectively assign tasks and plan trajectories for collaborative mission execution.

In post-disaster scenarios, different task points exhibit varying requirements for both the types and quantities of resources, as well as differing levels of task urgency. Consequently, in light of the characteristics of the post-disaster environment, the state of UAV i is defined as $U_i = \langle P_i^{\text{uav}}, D_i, E_i, W_i, V_i \rangle$, where P_i^{uav} denotes the current position of UAV i, $P_i^{\text{uav}} = (x_i, y_i, z_i)$, D_i is its flight distance, E_i is its total energy, W_i is its maximum payload and V_i is its maximum speed. The state of task point j is defined as $T_j = \langle P_j^{\text{task}}, Q_j, R_j \rangle$, where P_j^{task} represents the position of task point j, Q_j denotes the task demand, including three categories: medical supplies, food and clothing and R_j indicates the task priority, divided into three levels: high, medium and low. For clarity, the variables used in the modeling process are listed in Table 1.

2.2. *Task allocation model*

In emergency missions, timeliness is paramount, while the energy consumption of UAVs directly limits their operational range and endurance. Consequently, an equitable task allocation is critical to prevent overloading some UAVs while others remain idle, thus enhancing overall efficiency and mitigating the risk of single-point failures. The objective function encapsulating these goals is formulated in (1)–(3).

(i) Total Task Completion Time (TCT)
The maximum time for a single UAV to complete all its assigned tasks. The completion time for UAV i to finish all its tasks is the sum of the flight time to each

Table 1. Variable descriptions.

Variable symbols	Variable meaning	Variable description
N	Number of UAV	UAV indices: $1, 2, \ldots, N$
M	Number of task	Task indices: $1, 2, \ldots, M$
$\mathcal{I}_N$	Index set	$\mathcal{I}_N = \{1, 2, \ldots, N\}$
$\mathcal{I}_M$	Index set	$\mathcal{I}_M = \{1, 2, \ldots, M\}$
x_{ij}	Task allocation decision variables	If task point j is allocated to drone i, $x_{ij} = 1$; otherwise, $x_{ij} = 0$
d_{ij}	Expected flight path length	Expected flight path length of UAV i from its current position to task point j
y_{ijk}	Operation planning decision variables	If drone i chooses the path connecting task points j and k, $y_{ijk} = 1$; Otherwise, $y_{ijk} = 0$
d_{ijk}	Flight path length	Flight path length of UAV i from task point j to task point k
r_{eff}	Task execution efficiency	The amount of tasks executed per unit time
k_f	Distance energy consumption	The energy consumption coefficient per unit distance
k_w	Task energy consumption	The energy consumption coefficient per unit task
$G = (V, E)$	Undirected graph representation of the problem	V represents the set of nodes E represents the set of edges
$h_i^{(k)}$	Node feature	The feature of node v_i at the k layer
$N(i)$	Neighbor node	The set of neighboring nodes of node v_i

target task point and the time required to execute the tasks.

$$f_1 = \max_i \sum_{j=1}^{M} \left(\frac{d_{ij}}{V_i} + \frac{Q_j}{r_{\text{eff}}} \right) \cdot x_{ij}. \tag{1}$$

(ii) UAV Energy Consumption (UEC)
The total energy consumed by all UAVs to complete their assigned tasks, which is the sum of the energy consumed for flying to the target task points and the energy required to execute the tasks.

$$f_2 = \sum_{i=1}^{N} \sum_{j=1}^{M} (k_f \cdot d_{ij} + k_w \cdot Q_j) \cdot x_{ij} \tag{2}$$

(iii) Task Load Balancing
This objective function aims to minimize the variance of TCT to ensure that the task loads among all UAVs are as balanced as possible.

$$f_3 = \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{j=1}^{M} \left(\frac{d_{ij}}{V_i} + \frac{Q_j}{r_{\text{eff}}} \right) \cdot x_{ij} - \bar{t} \right)^2, \tag{3}$$

where $\bar{t}$ denotes the average TCT of all UAVs.

The objective function of the task allocation problem is defined in (4), which includes minimizing TCT, minimizing energy consumption and balancing task allocation.

$$F = \min[f_1, f_2, f_3]. \tag{4}$$

The constraint conditions are defined in (5)–(8).

(i) UAV Payload Constraint
The total payload of the UAV cannot exceed its maximum carrying capacity.

$$\sum_{j=1}^{M} Q_j \cdot x_{ij} \leq W_i, \quad \forall\, i \in \mathcal{I}_N. \tag{5}$$

(ii) UAV Endurance Time Constraint
The total energy consumption of the UAV during mission execution cannot exceed its total power consumption E_i.

$$k_f \cdot \sum_{j=1}^{M} d_{ij} \leq E_i. \tag{6}$$

(iii) Exclusive Task Assignment Constraint
Each task can only be executed by one UAV at most.

$$\sum_{i=1}^{N} x_{ij} \leq 1, \quad \forall\, j \in \mathcal{I}_M. \tag{7}$$

(iv) Task Priority Constraint
UAVs are assigned to high-priority tasks first.

$$\sum_{i=1}^{N} x_{ij} = 1, \quad \text{if } R_j = \text{high}, \tag{8}$$

where R_j = high represents that task point j has a higher priority compared to other unallocated task points.

2.3. *Operation planning model*

The decision variable of the operation planning model is the path selection variable y_{ijk}, which indicates whether UAV i moves along edge k when traveling from its current position to task point j. This variable is a discrete binary variable. Given the time-critical nature of post-disaster supply delivery and the need to minimize operational cost, the operation planning problem is formulated with the objective functions defined in (9)–(10).

(i) Total Flight Path Length
The sum of the flight path lengths of all UAVs to complete all assigned tasks.

$$g_1 = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k \in E} d_{ijk} \cdot y_{ijk}. \tag{9}$$

(ii) UAV Operation Time
The total operation time for a UAV to complete all its tasks, which is the sum of its flight time and task execution time.

$$g_2 = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k \in E} \left(\frac{d_{ijk}}{V_i(t)} + \frac{Q_j}{r_{\text{eff}}} \right) \cdot y_{ijk}. \tag{10}$$

The objective function of the operation planning problem is defined in (11), which includes minimizing the total flight path length and minimizing the UAV operation time.

$$G = \min[g_1, g_2]. \tag{11}$$

The constraint conditions are defined in (12)–(15).

(i) Path Feasibility Constraint
This constraint ensures that the number of times the drone enters the task point k is equal to the number of times it leaves the task point k.

$$\sum_{j=1}^{M} y_{ijk} = \sum_{m=1}^{M} y_{ikm}, \quad \forall\, i, k \in \mathcal{I}_M. \tag{12}$$

(ii) UAV Energy Constraint
The energy consumption of the drone during mission execution does not exceed the initial power consumption.

$$\sum_{j=1}^{M} \sum_{k=1}^{M} k \cdot d_{ijk} \cdot y_{ijk} \leq E_i, \quad \forall\, i \in \mathcal{I}_N. \tag{13}$$

(iii) Track Safety Constraint
UAVs must ensure a minimum safe distance during flight.

$$d_{ij} \geq d_{\min}, \quad \forall\, i, j \in \mathcal{I}_N, \tag{14}$$

where d_{ij} is the distance between UAV i and UAV j, and $d_{\min}$ is the minimum safe distance.

(iv) Flight Speed Constraint
The flight speed of the UAV shall not exceed its maximum speed.

$$v_i(t) \geq V_i, \quad \forall\, i \in \mathcal{I}_N, \tag{15}$$

where $v_i(t)$ is the speed of UAV i at time t.

2.4. *Model analysis*

The presented model defines a spatiotemporally coupled multi-UAV cooperative planning problem, which is a Mixed-Integer Programming (MIP) formulation. Furthermore, both its constituent problems — task allocation and path planning are NP-hard. At the decision variable level, the task allocation decision variables x_{ij} determine which operation planning decision variables p_{ijk} need to be optimized, while the path operation cost in p_{ijk} influence the evaluation of the quality of x_{ij}.

At the objective level, a tight coupling exists between the two models. The objective of the task allocation model (1) relies on UAV path lengths from the operation plan, while the execution time in the operation model (9) is contingent on the allocation scheme. This bidirectional dependency, combined with shared energy constraints (3) and (10), creates a coupled, closeloop system. Furthermore, both models share the dual objectives of minimizing time and energy. Crucially, task allocation pursues global minima for total mission time and energy, whereas operation planning seeks local minima for the path length (directly impacting energy) and operation time of individual UAVs. Since the aggregation of local optima does not guarantee a global optimum, coordinated optimization using global information is essential.

3. Multi-UAV Task Planning Algorithm Based on GNN Decoupling

3.1. *GNN decoupling algorithm framework*

For the above multi-UAV cooperative task planning problem, traditional methods often employ a decoupled optimization of task allocation and operation planning, leading to significant loss of global information. However, a centralized approach to the task planning problem, which accounts for the strong coupling between the task allocation and operation planning sub-problems, notably increases the computational complexity. To overcome this challenge, we propose a hierarchical decoupling optimization algorithm. This method leverages a weak-coupling structure to decompose the problem, enhancing computational efficiency while guaranteeing the optimality of the overall mission plan. The algorithm that combines GNNs with a dynamic inertia weight-based multi-objective particle swarm optimization (GNN-ADWMOPSO). Its core innovation lies in first using GNN to construct a unified model of the complex topology involving UAVs, task points, and the environment. The GNNs message-passing mechanism explicitly captures inter-UAV cooperation and competition, producing node and graph embeddings that encode the global system state. These rich contextual embeddings then serve as high-quality feature representations for informed decision-making in the subsequent optimization stage.

The multi-UAV task planning problem is mapped to a weighted undirected graph model, as illustrated in Fig. 1. UAV nodes and task nodes serve as the graph's vertices, with node features encompassing UAV states and task point states. Edge weights are calculated based on UAV — task execution cost and inter-UAV cooperation relationships.

To enable rapid decision-making, our framework first employs the A* algorithm for efficient path prediction on an abstract environmental map. This is followed by an Adaptive Dynamic Weight Multi-Objective Particle Swarm Optimization (ADWMOPSO) algorithm for integrated decision optimization. During the task allocation phase, the GNN facilitates global cooperative decision-making via iterative message passing. UAV and task nodes exchange critical information — such as resource requirements, path conflict risks and load status — through edges, which is then aggregated across multiple neural layers to produce a probabilistic task allocation distribution. In the operation planning phase, the GNN leverages spatiotemporal features to identify potential conflicts based on the paths generated by ADWMOPSO. It subsequently adjusts UAV flight sequences or generates local detours, thereby enabling efficient supply delivery to all known disaster sites while dynamically mitigating operational risks.

The algorithm framework is shown in Fig. 2. The feature extraction phase employs graph construction and

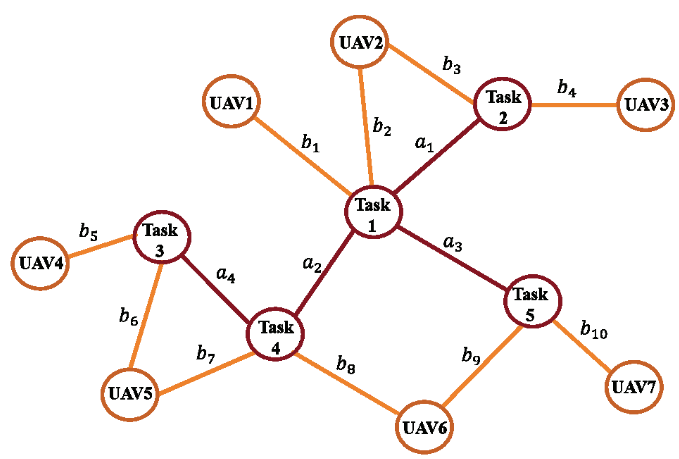


Fig. 1. Weighted undirected graph model.

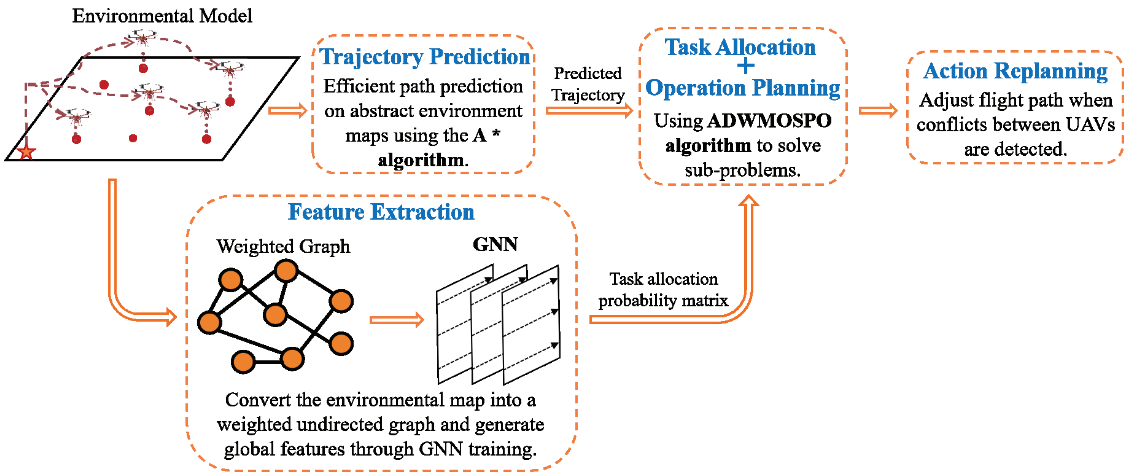


Fig. 2. Hierarchical decoupling algorithm framework.

GNN training to learn inter-task relationships and generate embedded representations, thereby reducing the computational complexity of multi-objective optimization and enhancing global information interaction. Subsequently, the task planning phase leverages these embeddings and predicted trajectories to achieve closed-loop optimization of allocation and operation planning, improving efficiency through feature-driven guidance. This strategy effectively balances computational efficiency with global performance, significantly boosting the adaptability and robustness of multi-UAV cooperation in complex environments. The proposed algorithm is accordingly structured into four integrated components: GNN-based global feature extraction, A*-based trajectory prediction, ADWMOPSO-based task allocation and operation planning and operation replanning. The design and implementation of each component are detailed in the following sections.

3.2. *GNN-based feature extraction*

GNN-based feature extraction provides high-quality input for subsequent multi drone task planning. It generates graph embeddings that encode the global context, alongside node and edge embeddings that capture individual states and semantic relationships. Furthermore, the similarity and distance between these embedding vectors enable the derivation of a UAV-task compatibility matrix. The task allocation problem is modeled as an undirected weighted graph $G = (V, E)$, where V denotes the set of nodes representing UAVs and task points, E represents the connections between nodes and the edge weights ω_{ij} indicate either the Euclidean distance or the task execution cost between nodes v_i and v_j, capturing the relationships between UAVs and task points.

To capture the complex interdependencies arising from the task point distribution and required UAV collaboration, we utilize a GNN for feature extraction. As shown in Fig. 3,

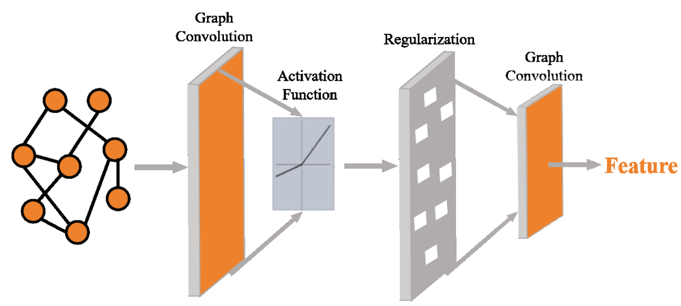


Fig. 3. GNN feature extraction.

GNN applies a multi-layer message passing mechanism to update and enhance node features.

In each graph convolution layer, the GNN updates the features of a target node by aggregating information from its neighboring nodes. The feature update formula for node v_i at the $k+1$ layer is given in (16).

$$h_i^{(k+1)} = \sigma(W^{(k)} \cdot AGG(\{h_i^{(k)}, h_j^{(k)} \mid j \in N(i)\})), \tag{16}$$

where $W^{(k)}$ is the learnable weight matrix at the kth layer, σ is a nonlinear activation function and $AGG(\cdot)$ is the aggregation function used to combine the feature information. The aggregation function is defined in (17).

$$AGG(\{h_i^{(k)}, h_j^{(k)} \mid j \in N(i)\}) = \sum_{j \in N(i)} \omega_{ij} \cdot h_i^{(k)}, \tag{17}$$

where ω_{ij} is the weight of edge e_{ij}, which measures the influence strength of the neighboring node j on the target node i.

Through iterative updates defined by (17), each node's feature representation progressively fuses global contextual information from the entire graph while preserving its local structural characteristics. The resulting node embeddings $h_i^{(k)}$, produced after multiple GNN layers, thus encapsulate the spatial configuration of tasks and UAVs, their collaborative potential and associated cost. These embeddings subsequently serve as efficient and information-rich inputs for downstream task allocation and operation planning.

3.3. *Trajectory prediction based on the A* algorithm*

For efficient route planning, the A* algorithm computes the shortest paths from UAV start points to task locations while adhering to flight constraints. Its core principle leverages a heuristic search to guide the exploration within a discretized 2D grid representation of the disaster area. The algorithm's total cost function is defined as

$$F(n) = G(n) + H(n), \tag{18}$$

where $G(n)$ is the actual cost and $H(n)$ is the heuristic cost, which is the Euclidean distance from the current node to the goal node, calculated as

$$H(n) = \sqrt{(x_n - x_{\text{goal}})^2 + (y_n - y_{\text{goal}})^2}. \tag{19}$$

3.4. *Task allocation and operation planning based on ADWMOPSO*

To effectively address the integrated task allocation and operation planning problem for multiple UAVs, this paper proposes an enhanced multi-objective particle swarm optimization algorithm named ADWMOPSO. The algorithm incorporates a dynamic inertia weight mechanism that adjusts this parameter across different search phases, thereby adaptively balancing global exploration and local exploitation. This strategic adaptation helps mitigate premature convergence and enables the generation of high-quality, diverse solutions for both task assignment and operation sequencing. The approach is particularly suited for navigating the high-dimensional, nonconvex Pareto solution space inherent to this challenging optimization problem.

3.4.1. *Population initialization*

The decision variables of the problem are represented as particles. For the task allocation problem, the particle p is denoted as $[x_{11}, \dots, x_{1M}, \dots, x_{NM}]$, where x_{ij} indicates whether task j is assigned to UAV i. For the operation planning problem, the particle p is denoted as $[p_{11k}, \dots, p_{1Mk}, \dots, p_{NMk}]$, where p_{ijk} indicates whether UAV i moves along edge k when traveling from its current position to task point j.

The node feature matrix output by the GNN is aligned through a weight matrix $W \in R^{D' \times D'}$ and divided according to node type into a UAV node feature matrix $H_U \in R^{N \times D'}$ and a task node feature matrix $H_T \in R^{M \times D'}$. The UAV–task matching score matrix $S \in R^{N \times M}$ is then computed as

$$S_{ij} = \text{ReLU}(H_{U_i} \cdot W \cdot H_{T_j}). \tag{20}$$

The matching score matrix is passed through a ReLU activation function, and its rows are normalized to produce a task allocation probability matrix. This matrix is then used to generate an initial task allocation scheme. Concurrently, an archive is initialized to store Pareto-optimal solutions throughout the optimization process.

3.4.2. *Fitness evaluation*

Based on the established multi-UAV cooperative task planning model, the optimization objectives are defined as follows: the task allocation phase aims to minimize the total mission completion time, reduce total UEC and ensure a balanced workload distribution; the operation planning phase seeks to minimize the cumulative trajectory length and the total UAV operation time. The fitness of the initial particle swarm is evaluated according to these objectives.

3.4.3. *Selecting guided particles*

For each particle p, the raw fitness value f is calculated and then adjusted using a penalty function. The fitness of solutions that violate constraints is dynamically modified. The penalty function is defined as

$$f_{pen} = f + \lambda \cdot \text{pen}, \tag{21}$$

where f_{pen} is the fitness value including the penalty term, λ is the penalty coefficient and pen represents the total constraint violation, indicating the degree to which constraints are violated.

3.4.4. *Particle update*

A dynamic inertia weight is incorporated into the velocity update to adaptively balance global exploration and local exploitation throughout the optimization process. This mechanism assigns a higher weight during the initial phase to promote extensive global search, and systematically reduces it in later phases to fine-tune solutions and guide the swarm toward convergence on the Pareto-optimal front. The velocity update is formulated as follows:

$$\begin{aligned} v_p^{(t+1)} &= \omega^{(t)} \cdot v_p^{(t)} + c_1 \cdot r_1 \cdot (p_{\text{best}} - x_p^{(t)}) \\ &\quad + c_2 \cdot r_2 \cdot (g_{\text{best}} - x_p^{(t)}), \end{aligned} \tag{22}$$

where $\omega^{(t)}$ is the dynamic inertia weight at the tth iteration, which decreases linearly during the iteration process. c_1 and

c_2 are the cognitive and social learning factors, respectively, r_1 and r_2 are random numbers.

Particle positions are updated, and their fitness values are recalculated. Individual best solutions p_{best} and the global best solution g_{best} are then updated accordingly.

3.4.5. *Particle archiving*

Particles satisfying the Pareto-optimal conditions are added to the archive. An overcrowding-based selection mechanism is applied to remove excess solutions, maintaining a fixed archive size.

3.4.6. *Termination condition*

The algorithm terminates when the maximum number of iterations or a convergence criterion is reached, and the final Pareto-optimal solution set is output.

The algorithm flowchart is shown in Fig. 4.

3.5. ***Operation replanning***

During the task execution phase, when flight path conflicts occur between UAVs, operation replanning is required to ensure flight safety. The GNN predicts potential UAV path conflicts based on node and edge embeddings through a message-passing mechanism. After L layers of message passing, the final features of UAV nodes i and j are $h_i^{(L)}$ and $h_j^{(L)}$, respectively. These node features are linearly combined through a fully connected layer and then passed through an activation function to generate a conflict probability matrix $C \in R^{N \times N}$:

$$C_{ij} = \sigma(W_c \cdot [h_i^{(L)} || h_j^{(L)}] + b_c), \tag{23}$$

where W_c is the weight matrix of the fully connected layer, $||$ represents vector concatenation operation, b_c is the bias term and σ is the sigmoid function. When $C_{ij} > \tau$ (where τ is a given threshold), the UAV pair (i, j) is considered to have a conflict that requires resolution.

For a conflicting UAV pair (i, j), based on the task execution priorities S_i and S_j, if $S_i > S_i$, UAV i is given priority to proceed and UAV j must adjust its path. Using the path information of the conflicting UAV pair, the intersection point is identified as the conflict point P_{conflict}. Around P_{conflict}, a circular region with a radius equal to the safety distance threshold is defined. Existing paths within this region are removed and safe waypoint set P_{safe} is generated by uniformly sampling in polar coordinates within the region. For each candidate waypoint P_k in the safe waypoint set, the detour distance added to the original path is calculated. The UAV j selects the waypoint P_{best} that results in the shortest detour and incorporates it into the original path, achieving operation replanning.

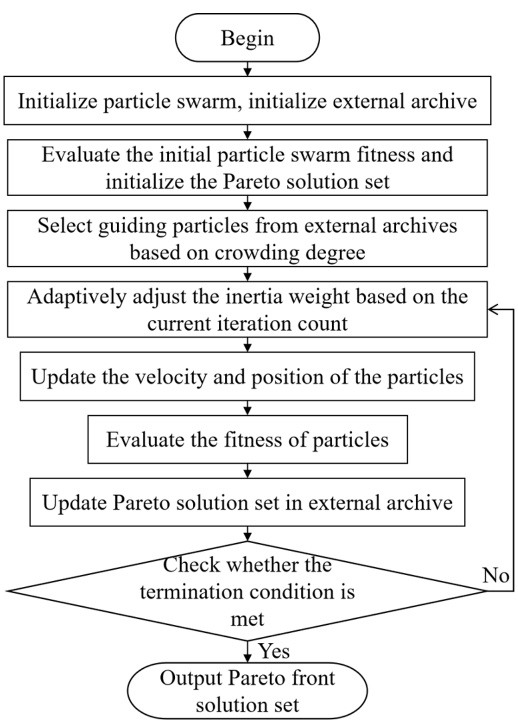


Fig. 4. Flowchart of the ADWMOPSO algorithm.

4. Simulation Experiments

4.1. ***GNN training***

For small-scale problem instances, optimal allocation solutions derived from a solver serve as the training data for the GNN. To ensure comprehensive coverage, the training dataset incorporates diverse representative operational conditions, encompassing variations in task point distribution, priority levels and resource demands. This approach effectively simulates a wide range of real-world complexities, thereby robustly generalizing the model's performance across different environments. Specific scenario configurations are detailed in Table 2.

Table 2. Typical training scenario design.

Position distribution	Priority	Resource quantity
Single-center cluster cluster	Balanced number of different priorities	Resource quantity greater than demand
Multi-center cluster	Same priority	Resource quantity balanced with demand
Uniformly dispersed	—	Resource quantity less than demand

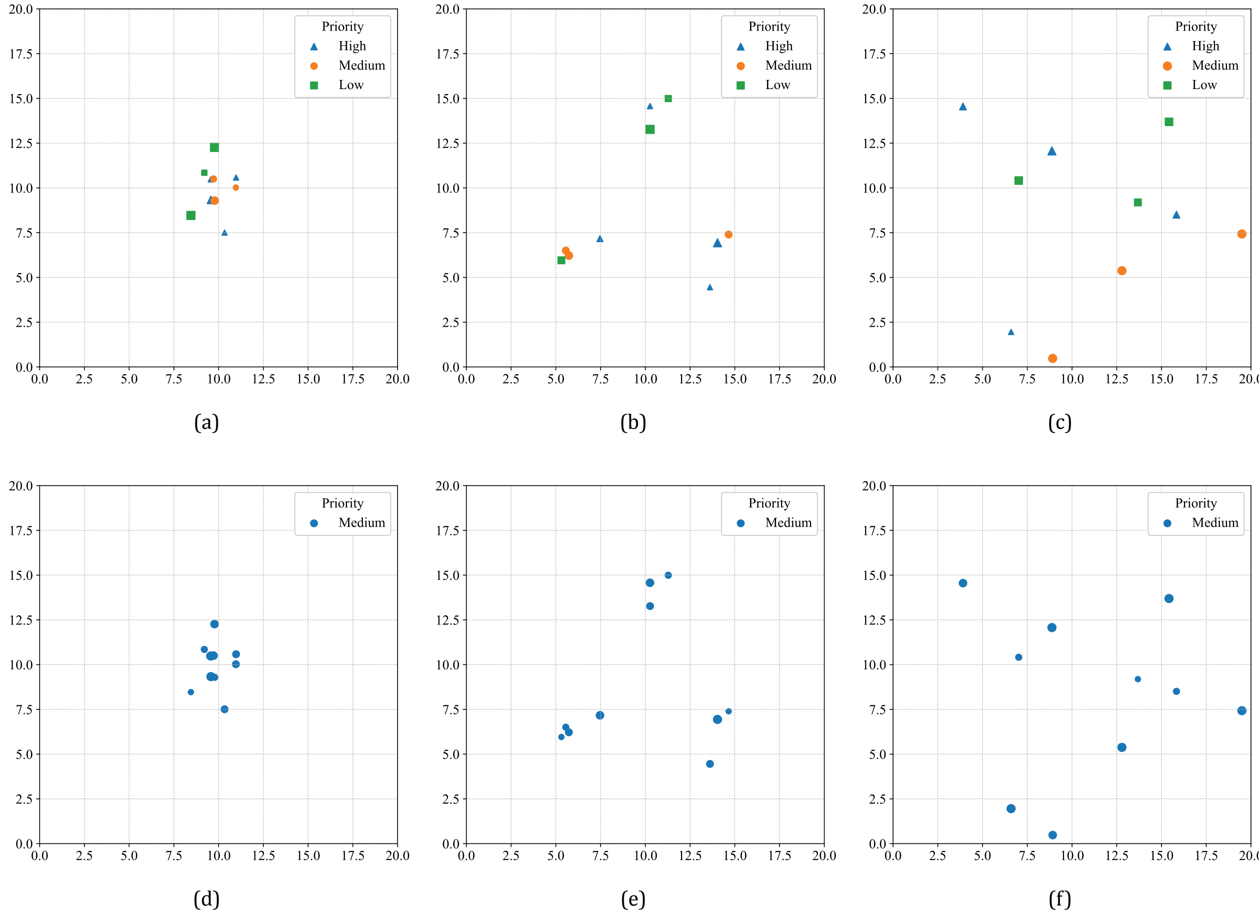


Fig. 5. Typical training scenario diagrams. (a) Single-center different priorities, (b) multi-center different priorities, (c) uniform — different priorities, (d) single-center same priority, (e) multi-center same priority and (f) uniform — same priority.

An example of the typical training scenario is shown in Fig. 5. In the scenario graph, the size of each task point is proportional to its task demand. Since the example graph only illustrates the task demand, the case where the resource quantity exceeds the demand is used as an example.

To validate the GNN model, 100 data samples were generated per scenario, yielding a total dataset of 1800 samples. This dataset was partitioned into a 70% training set, 15% validation set and 15% test set. All typical scenarios are represented in each subset with a consistent proportion to prevent distributional bias. The hyperparameters employed for training are detailed in Table 3.

Table 3. Training hyperparameter.

Hyperparameter	Value
Batch size	16
Learning rate	0.001
Optimizer	Adam
Loss function	CrossEntropyLoss
Number of epochs	100

The convergence of the GNN training is visualized by the loss curve in Fig. 6. During training, model states were checkpointed at every epoch, with the final model being selected for evaluation based on its superior validation performance, thereby effectively circumventing overfitting.

4.2. *Comparative experiments*

4.2.1. *Experimental setup*

To evaluate the effectiveness and superiority of the proposed GNN-ADWMOPSO method under varying scenarios, we designed three task scenarios of different scales, with specific parameters listed in Table 4. To facilitate experimentation, the task scenario is an abstraction of a real environment that preserves the defining features of actual post-disaster missions, thereby ensuring the relevance and applicability of our findings to real-world situations. All scenarios are set in a 20 km $\times$ 20 km rectangular disaster area. Each task point is characterized by a priority level P (High, Medium, or Low) and a demand value D (in kg), where the

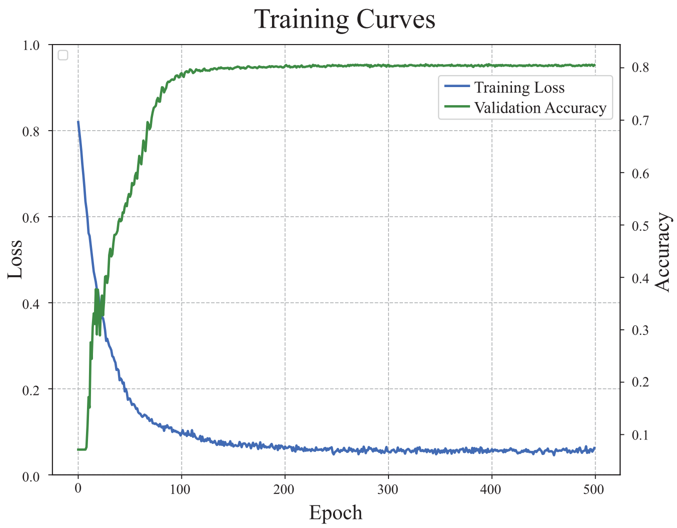


Fig. 6. Training Curves.

Table 4. Experimental scenario settings.

Scenario	Number of UAVs	Number of task points
Scenario 1	3	10
Scenario 2	5	20
Scenario 3	10	50

demand type corresponds to the supply category (medical supplies, food, or clothing). Medical supplies are consistently assigned the highest priority. All UAVs are initialized at coordinates $(0, 0)$, with a maximum flight speed of 30 m/s, a maximum payload capacity of 20 kg, and an endurance of 2 h.

4.2.2. *Experimental results*

Upon completion of training, the GNN transforms its output node feature matrix into a task allocation probability matrix, from which an initial assignment scheme is derived. Building on this initial solution, the ADWMOPSO algorithm is applied to obtain the final task planning result, illustrated in Fig. 7.

Table 5. Task planning data results.

Scenario	UAV flight path length (km)	TCT (min)
Scenario 1	127.12	29.32
Scenario 2	200.94	36.71
Scenario 3	424.77	47.13

The arrows in the figure represent the UAVs' task sequences. Corresponding flight path lengths and mission completion time are quantitatively summarized in Table 5.

4.2.3. *Ablation study*

To systematically evaluate the contribution of each core component in the proposed GNN-ADWMOPSO framework, we conduct an ablation study by comparing several algorithm variants under controlled conditions. This allows for a precise assessment of the individual contributions of the GNN-based feature extraction module and the adaptive strategy in ADWMOPSO. The following variants are designed:

(i) ADWMOPSO Algorithm
This variant removes the GNN feature extraction module from the full GNN-ADWMOPSO algorithm, retaining only the ADWMOPSO optimizer with dynamic adaptive factors and constraint handling. The initial population is generated using conventional random initialization. This setup is used to evaluate the necessity and effectiveness of the GNN training and feature representation.

(ii) GNN-MOPSO Algorithm
In this variant, the dynamic inertia weight mechanism in ADWMOPSO is replaced with the standard MOPSO, while retaining the GNN feature extraction module. It serves to validate the contribution of the adaptive inertia weight strategy in ADWMOPSO.

Each algorithm variant, including the full model, is executed over 25 independent runs across three scenario scales. Performance is measured using three key metrics: shortest

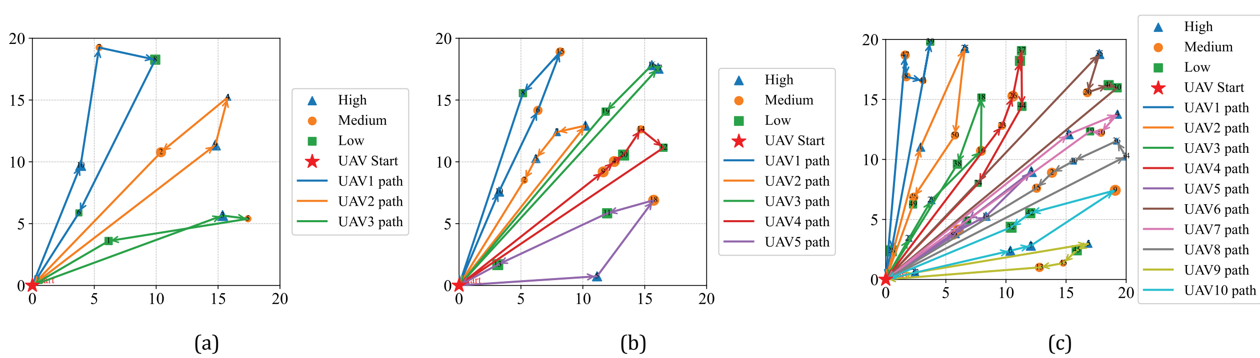


Fig. 7. Task planning result diagram. (a) Scenario 1, (b) Scenario 2 and (c) Scenario 3.

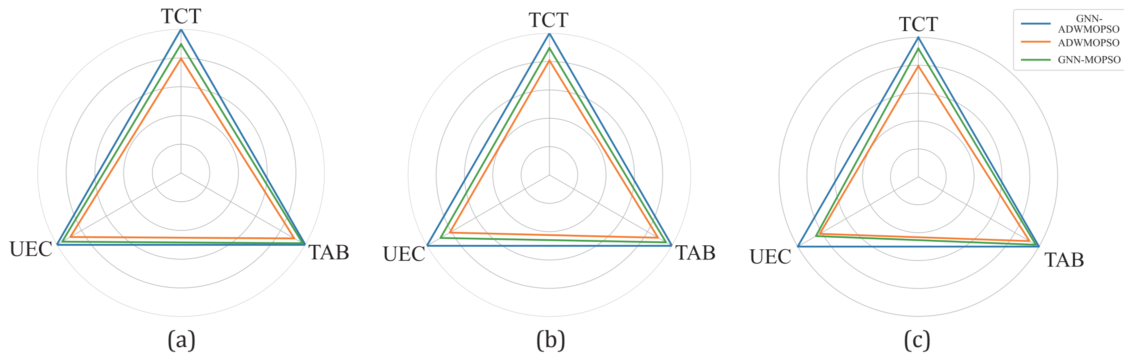


Fig. 8. Radar chart comparison of the ablation experiments. (a) Scenario 1, (b) Scenario 2 and (c) Scenario 3.

TCT, lowest UEC and highest task allocation balance (TAB). A radar chart summarizing the comparative results is provided in Fig. 8.

Across all three experimental scenarios, GNN-ADWMOPSO consistently achieves superior performance compared to the ADWMOPSO variant in mission completion time, energy consumption and TAB. This result confirms that the GNN feature extraction module significantly enhances initial solution quality and effectively guides the search process. Furthermore, GNN-ADWMOPSO also outperforms the GNN-MOPSO variant, validating that the dynamic inertia weight strategy successfully balances global exploration and local exploitation throughout the optimization process.

4.2.4. *Comparative algorithms*

To further compare algorithm performance, two benchmark algorithms were selected in this study, including:

(i) Ant Colony Optimization with Evolution Strategy (ES-ACO) [37].
This algorithm centrally solves the task planning problem, incorporating a roulette wheel selection principle and elite strategy into the ACO framework. The algorithm parameters are listed in Table 6.

(ii) Multi-layer Planner [38]
This algorithm divides the task planning problem into three hierarchical layers, including a task allocation layer based on a genetic algorithm, a conflict resolution layer based on an improved Conflict-Based Search (CBS) variant, and a path planning layer using a spatio-temporal hybrid-state A* algorithm. The algorithm parameter settings are listed in Table 7.

Table 6. ES-ACO parameter.

Parameter	Symbol	Value
Number of ants	m	40
Pheromone importance factor	α	1
Heuristic importance factor	β	7
Pheromone evaporation factor	ρ	0.3
Pheromone reinforcement factor	Q	1

Table 7. Multi-layer planner parameter.

Parameter	Value
Population size	40
Crossover probability	0.8
Mutation probability	0.1
Conflict detection radius	10 m

For the comparison algorithms, the maximum number of iterations is set to 100, and 25 independent repeated experiments are conducted. The final results are reported as the statistical average.

4.2.5. *Results analysis*

Based on the multi-UAV task planning model, the effectiveness of actual task plans can be evaluated using indicators such as maximum task completion rate (TCR), minimum TCT, minimum UEC and highest task load balance (TLB). The algorithm performance can be assessed in terms of convergence and diversity. Convergence measures how close the solution set is to the true Pareto front, while diversity evaluates the uniformity and spread of solutions. The Spacing (SP) and Hypervolume (HV) metrics are calculated for the solution set. SP measures the uniformity between adjacent solutions within the set; a smaller SP indicates a more uniform distribution. HV assesses solution diversity and convergence; a larger HV indicates higher solution quality. The average and standard deviation of results over 25 runs for each algorithm under different scenarios are shown in Table 8.

Table 9 presents the results of the Wilcoxon rank-sum test to assess statistical significance at the 5% level. Based on the Wilcoxon rank sum test results, the GNN-ADWMOPSO

Table 8. Comparison of algorithm performance under different scenarios.

Scenario	Algorithm	TCR	TCT	UEC	TLB	SP	HV
Scenario 1	GNN-ADWMOPSO	$\mathbf{99.9 \pm 0.1}$	$\mathbf{29.3 \pm 2.2}$	$\mathbf{15.2 \pm 1.8}$	$\mathbf{0.968 \pm 0.11}$	$\mathbf{0.10 \pm 0.02}$	$\mathbf{0.96 \pm 0.13}$
Scenario 1	ES-ACO	98.8 ± 1.2	35.6 ± 4.8	17.3 ± 2.4	0.912 ± 0.22	0.17 ± 0.03	0.88 ± 0.17
Scenario 1	Multi-layer planner	97.2 ± 2.2	38.3 ± 5.5	19.6 ± 3.1	0.862 ± 0.23	0.15 ± 0.03	0.82 ± 0.24
Scenario 2	GNN-ADWMOPSO	$\mathbf{99.2 \pm 0.8}$	$\mathbf{36.7 \pm 4.5}$	$\mathbf{59.8 \pm 6.9}$	$\mathbf{0.938 \pm 0.22}$	$\mathbf{0.15 \pm 0.03}$	$\mathbf{0.91 \pm 0.12}$
Scenario 2	ES-ACO	97.1 ± 1.8	43.2 ± 6.0	68.9 ± 8.8	0.878 ± 0.27	0.23 ± 0.04	0.82 ± 0.16
Scenario 2	Multi-layer planner	94.8 ± 4.6	49.3 ± 9.6	78.4 ± 11.7	0.821 ± 0.34	0.21 ± 0.04	0.75 ± 0.18
Scenario 3	GNN-ADWMOPSO	$\mathbf{98.7 \pm 1.2}$	$\mathbf{47.1 \pm 4.1}$	$\mathbf{83.5 \pm 9.2}$	$\mathbf{0.912 \pm 0.16}$	$\mathbf{0.18 \pm 0.04}$	$\mathbf{0.86 \pm 0.13}$
Scenario 3	ES-ACO	95.6 ± 3.5	54.4 ± 4.7	96.8 ± 10.4	0.843 ± 0.24	0.28 ± 0.06	0.74 ± 0.14
Scenario 3	Multi-layer planner	92.3 ± 4.1	59.3 ± 6.2	107.2 ± 12.8	0.782 ± 0.27	0.23 ± 0.07	0.68 ± 0.15

Table 9. Wilcoxon rank-sum test results for GNN-ADWMOPSO against other methods (*p*-value and significance).

Scenario	Algorithm	TCR	TCT	UEC	Task load balance (TLB)	SP	HV
Scenario 1	ES-ACO	0.0000/Yes	0.0000/Yes	0.0004/Yes	0.1541/No	0.0000/Yes	0.0071/Yes
Scenario 1	Multi-layer planner	0.0000/Yes	0.0000/Yes	0.0000/Yes	0.0300/Yes	0.0000/Yes	0.2098/No
Scenario 2	ES-ACO	0.0067/Yes	0.0104/Yes	0.0001/Yes	0.5184/No	0.0000/Yes	0.0042/Yes
Scenario 2	Multi-layer planner	0.0030/Yes	0.0000/Yes	0.0000/Yes	0.3181/No	0.0000/Yes	0.2098/No
Scenario 3	ES-ACO	0.0000/Yes	0.0000/Yes	0.0000/Yes	0.7690/No	0.0001/Yes	0.0001/Yes
Scenario 3	Multi-layer planner	0.0000/Yes	0.0000/Yes	0.0000/Yes	0.2076/No	0.0117/Yes	0.0000/Yes

algorithm exhibits significant advantages in most performance metrics. In terms of TCR, completion time, drone energy consumption and SP indicators, this algorithm significantly outperforms ES-ACO and Multi layer Planner in all scenarios. Especially in the complex environment of Scenario 3, all comparisons showed statistical significance. In terms of task load balancing, the differences between GNN-ADWMOPSO and the comparative algorithms are mostly not significant. Overall, GNN-ADWMOPSO has significant advantages in core performance indicators and is particularly suitable for application scenarios that require high execution efficiency, energy consumption, and success rate. As the complexity of the problem increases, the relative advantage of this algorithm becomes more prominent.

The comparison results of the solution performance of different algorithms are presented using a radar chart, as shown in Fig. 9. The proposed GNN-ADWMOPSO demonstrates superior performance across all evaluation metrics — including TCR, TCT, energy consumption, load balance, spacing and hypervolume — compared to baseline methods. This performance advantage stems from the synergistic integration of GNN-based feature extraction and ADWMOPSO optimization. The GNN component employs message passing to capture global structural patterns, effectively reparameterizing the originally high-dimensional, strongly-coupled search space. Trained on representative data, it provides a learned prior that approximates the Pareto front geometry, enabling ADWMOPSO to initialize solutions nearer the true Pareto front. Furthermore, ADWMOPSO's dynamic inertia weight mechanism adaptively modulates search dynamics: increasing the weight enhances global exploration to extend the Pareto front boundaries, while decreasing it intensifies local exploitation around high-fitness regions. When coupled with crowding-distance-based selection, this approach promotes more uniform solution distribution, broader Pareto coverage and accelerated convergence.

A comparative analysis of the algorithm running time across different scenarios is presented in Table 10, where the bold values represent the shortest algorithm running time under the corresponding scenario. In Scenario 1, characterized by its relatively small scale, the centralized ES-ACO algorithm achieves the shortest runtime. The proposed GNN-ADWMOPSO, employing a hierarchical decoupling approach, demonstrates a modest increase in computational time under these simple conditions. However,

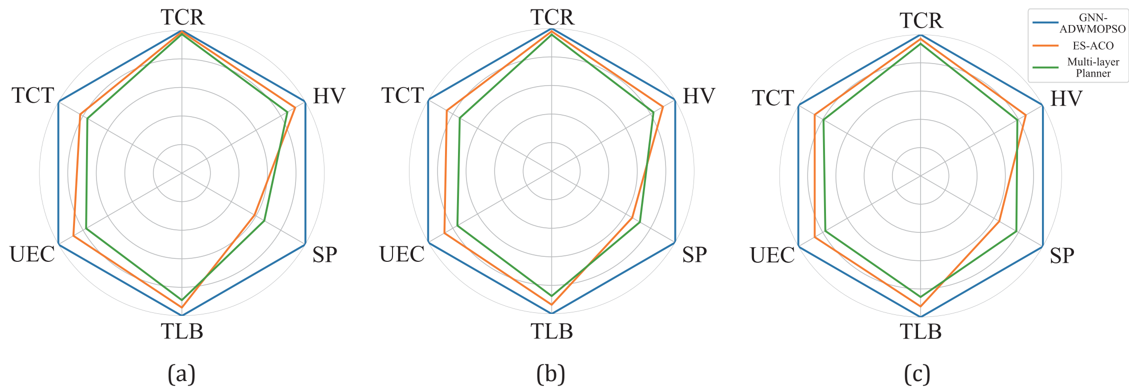


Fig. 9. Radar chart of solution performance for different algorithms. (a) Scenario 1, (b) Scenario 2 and (c) Scenario 3.

Table 10. Comparison of algorithm running time under different scenarios.

Algorithm	Algorithm running time (s)		
	Scenario 1	Scenario 2	Scenario 3
GNN-ADWMOPSO	2.35 ± 0.21	$\mathbf{6.32 \pm 0.64}$	$\mathbf{22.7 \pm 2.5}$
ES-ACO	$\mathbf{2.17 \pm 0.18}$	8.56 ± 0.74	36.3 ± 3.1
Multi-layer planner	3.84 ± 0.29	8.73 ± 1.02	48.4 ± 3.9

as problem complexity escalates in Scenarios 2 and 3, the computational burden of the centralized method grows substantially. In contrast, GNN-ADWMOPSO maintains computational efficiency through its decomposition of the original problem into smaller sub-problems, which significantly reduces search dimensions. Furthermore, the pre-trained GNN model enhances initial solution feasibility, minimizing the computational overhead associated with constraint handling and solution repair. Complementing this, the dynamic inertia weight mechanism effectively curtails search efforts, thereby accelerating convergence.

From a theoretical standpoint, GNN-ADWMOPSO enhances optimization efficiency through two synergistic mechanisms. First, the GNNs message-passing framework performs nonlinear dimensionality reduction and reparameterization of the originally high-dimensional, coupled solution space. This process encodes complex spatiotemporal relationships between UAVs and tasks into compact, dense embeddings, effectively learning a prior representation of the Pareto front geometry. Consequently, the optimization process initiates from regions proximate to high-quality solutions, substantially reducing problem complexity. Second, the dynamic inertia weight in ADWMOPSO enables adaptive search behavior: elevated weights during initial phases promote comprehensive exploration of the solution space guided by GNN embeddings, while reduced weights in later stages facilitate intensive exploitation of promising regions. This strategy, combined with crowding distance-based archive maintenance, ensures both diversity and convergence in the resulting nondominated solution set.

The GNN-ADWMOPSO framework significantly advances solving efficiency for multi-UAV task planning problems. It provides decision-makers with diverse, high-quality alternatives, enhancing both flexibility and scientific rigor in mission planning. This approach offers particularly valuable support for time-sensitive applications such as post-disaster emergency supply delivery, where both solution quality and computational efficiency are paramount.

5. Conclusion

In this study, a hierarchical solution method combining GNN and an improved MOPSO is proposed for the multi-UAV task planning problem. The proposed method leverages GNN to capture global structural relationships among UAVs and tasks, providing valuable guidance for the optimization process. Furthermore, an adaptive dynamic inertia weight mechanism is integrated into the MOPSO framework. This mechanism enables a dynamic balance between global exploration and local exploitation during the search process, effectively mitigating premature convergence. Experimental evaluations demonstrate that our approach consistently outperforms benchmark algorithms across multiple performance metrics, including TCR, mission duration, energy consumption, workload balance and multi-objective optimization criteria. This methodology not only ensures computational efficiency and solution stability but also delivers significantly improved solution quality. Future research will investigate real-time adaptation mechanisms for highly dynamic environments and advanced multi-agent coordination strategies to further enhance the framework's potential for practical deployment. Furthermore, future research will incorporate considerations of damaged infrastructure, communication disruptions and strict time windows in real-world post-disaster scenarios. This is crucial for enhancing the practical applicability of the research and for ensuring that the proposed algorithms and models maintain their efficiency and effectiveness in complex, real-world environments.

Acknowledgments

This paper is for the special issue of NYU Professor Zhong-Ping Jiang's 60th Birthday and is funded by the National Natural Science Foundation of China (62088101).

ORCID

Rui Hong
https://orcid.org/0009-0000-0573-9662
Jia Zhang
https://orcid.org/0000-0002-7792-2308
Minggang Gan
https://orcid.org/0000-0002-2163-2475
Qing Wang
https://orcid.org/0000-0002-4021-3372
Bin Xin
https://orcid.org/0000-0001-9989-0418
Jie Chen
https://orcid.org/0000-0003-2449-9793

References

[1] M. Reina Ortiz, N. K. Le, V. Sharma, I. Hoare, E. Quizhpe, E. Teran, E. Naik, H. M. Salihu and R. Izurieta, Post-earthquake Zika virus surge: Disaster and public health threat amid climatic conduciveness, *Sci. Rep.* **7**(1) (2017) 15408.

[2] A. Khan, S. Gupta and S. K. Gupta, Emerging UAV technology for disaster detection, mitigation, response, and preparedness, *J. Field Robot.* **39**(6) (2022) 905–955.

[3] J. Liao, K. Mao, B. Xin, J. He, Y. Zhang, J. Cheng, D. Luo and S. Zhou, Extended observer-based consensus tracking control for MASs with external disturbance and dynamic event-triggered strategy, *Sci. China Technol. Sci.* **68**(3) (2025) 1320403.

[4] N. Fallahiarezoodar and Z. H. Zhu, Review of autonomous space robotic manipulators for on-orbit servicing and active debris removal, *Space Sci Technol.* **5** (2025) 0291.

[5] S. A. Ghauri, M. Sarfraz, R. A. Qamar, M. F. Sohali and S. A. Khan, A review of multi-UAV task allocation algorithms for a search and rescue scenario, *J. Sens. Actuator Netw.* **13**(5) (2024) 47.

[6] P. Zhu, S. Jiang, J. Zhang, Z. Xu, Z. Sun and Q. Shao, Multi-target firefighting task planning strategy for multiple UAVs under dynamic forest fire environment, *Fire* **8**(2) (2025) 61.

[7] Y. Wu, S. Wu and X. Hu, Multi-constrained cooperative path planning of multiple drones for persistent surveillance in urban environments, *Complex Intell. Syst.* **7**(3) (2021) 1633–1647.

[8] J. Song, K. Zhao and Y. Liu, Survey on mission planning of multiple unmanned aerial vehicles, *Aerospace* **10**(3) (2023) 208–227.

[9] H. Gharrad, N. Jabeur, A. U. H. Yasar, S. Galland and M. Mbarki, A five-step drone collaborative planning approach for the management of distributed spatial events and vehicle notification using multi-agent systems and firefly algorithms, *Comput. Netw.* **198** (2021) 108282.

[10] Z. Liu, C. Qiu and Z. Zhang, Sequence-to-sequence multi-agent reinforcement learning for multi-UAV task planning in 3D dynamic environment, *Appl. Sci.* **12**(23) (2022) 12181.

[11] X. Wang, X. Gao, L. Wang, X. Su, J. Jin, X. Liu and Z. Deng, Resilient multi-objective mission planning for UAV formation: A unified framework integrating task pre-and re-assignment, *Def. Technol.* **45** (2025) 203–226.

[12] W. Yao, N. Qi, N. Wan and Y. Liu, An iterative strategy for task assignment and path planning of distributed multiple unmanned aerial vehicles, *Aerosp. Sci. Technol.* **86** (2019) 455–464.

[13] G. Zhao, Y. Wang, T. Mu, Z. Meng and Z. Wang, Reinforcement learning assisted multi-UAV task allocation and path planning for IIoT, *IEEE Internet Things J.* **11**(16) (2024) 26766–26777.

[14] Y. Li, W. Chen, B. Fu, Z. Wu and L. Hao, A hierarchical decoupling task planning method for multi-UAV collaborative multi-region coverage with task priority awareness, *Drones* **9**(8) (2025) 575.

[15] J. Hu, Z. Jia, Y. Sun and Q. Liu, Survey of analysis and solutions for multi-UAV cooperative mission planning problem under multi-constraint conditions, *Comput. Sci.* **50**(7) (2023) 176–193.

[16] Z. Hu, J. Liang, L. Chen and B. Wu, A hierarchical architecture for formation control of multi-UAV, *Procedia Eng.* **29** (2012) 3846–3851.

[17] P. Ghassemi and S. Chowdhury, Multi-robot task allocation in disaster response: Addressing dynamic tasks with deadlines and robots with range and payload constraints, *Robot. Auton. Syst.* **147** (2022) 103905.

[18] N. Lindsay, R. K. Buehling and L. Sun, A sequential task addition distributed assignment algorithm for multi-robot systems, *J. Intell. Robot. Syst.* **102**(2) (2021) 51.

[19] S. Gao, J. Wu and J. Ai, Multi-UAV reconnaissance task allocation for heterogeneous targets using grouping ant colony optimization algorithm, *Soft Comput.* **25**(10) (2021) 7155–7167.

[20] D. Han, H. Jiang, L. Wang, X. Zhu, Y. Chen and Q. Yu, Collaborative task allocation and optimization solution for unmanned aerial vehicles in search and rescue, *Drones* **8**(4) (2024) 138.

[21] J. Xiong, J. Li, J. Li, S. Kang, C. Liu and C. Yang, Probability-tuned market-based allocations for UAV swarms under unreliable observations, *IEEE Trans. Cybern.* **53**(11) (2023) 6803–6814.

[22] N. Dong, S. Liu and X. Mai, Communication-efficient heterogeneous multi-UAV task allocation based on clustering, *Comput. Commun.* **229** (2024) 107986.

[23] Q. Wang, L. Yang and J. Li, Research on distributed multi-robot motion planning based on ant colony algorithm fusion dynamic window approach, *J. Chongqing Univ. Posts Telecommun. (Nat. Sci. Ed.)* **36**(01) (2024) 20–28.

[24] P. Zhu and X. Fang, Multi-UAV cooperative task assignment based on half random Q-learning, *Symmetry* **13**(12) (2021) 2417.

[25] P. Peng, X. Gong and Y. Zheng, Multi-UAVs task allocation method based on MPSO-SA-DQN, *Meas. Control* **58**(8) (2025) 963–978.

[26] H. Zhang, B. Xin, L. Dou, J. Chen and K. Hirota, A review of cooperative path planning of an unmanned aerial vehicle group, *Front. Inf. Technol. Electron. Eng.* **21**(12) (2020) 1671–1694.

[27] Y. Du, Multi-UAV search and rescue with enhanced A* algorithm path planning in 3D environment, *Int. J. Aerosp. Eng.* **2023**(1) (2023) 8614117.

[28] Y. D. Yoo and J. H. Moon, Study on A-star algorithm-based 3D path optimization method considering density of obstacles, *Aerospace* **12**(2) (2025) 85.

[29] L. S. Jabbar, E. I. Abass and S. D. Hasan, A modification of shortest path algorithm according to adjustable weights based on Dijkstra algorithm, *Eng. Technol. J.* **41**(2) (2023) 1–16.

[30] H. Chen, Y. Liang and X. Meng, A UAV path planning method for building surface information acquisition utilizing opposition based learning artificial bee colony algorithm, *Remote Sens.* **15**(17) (2023) 4312.

[31] Z. A. Ali, Z. Han and Z. Di, Path planning of multiple UAVs using MMACO and DE algorithm in dynamic environment, *Meas. Control* **56**(3–4) (2023) 459–469.

[32] B. Niu, Y. Wang, J. Liu and G. X. Yue, Path planning for unmanned aerial vehicles in complex environment based on an improved continuous ant colony optimization, *Comput. Electr. Eng.* **123** (2025) 110034.

[33] N. I. Bouaziz, O. Mechali, K. L. Besseghieur and N. Achour, Trajectory planning for autonomous formation of wheeled mobile robots via modified artificial potential field and improved PSO algorithm, *Unmanned Syst.* **12**(06) (2024) 1085–1104.

[34] A. S. Rosas-Carrillo, A. Solis-Santome, C. Silva-Sanchez and O. Camacho-Nieto, UAV path planning using an adaptive strategy for the particle swarm optimization algorithm, *Drones* **9**(3) (2025) 170.

[35] S. Liu, W. Zhou, M. Qin and X. Peng, Tent–PSO-based unmanned aerial vehicle path planning for cooperative relay networks in dynamic user environments, *Sensors* **25**(7) (2025) 2005.

[36] R. Li, B. Xin, S. Zhang, M. Lyu and J. Cui, Swarm navigation based on smoothed particle hydrodynamics in complex obstacle environments, *IEEE Robot. Autom. Lett.* **10**(8) (2025) 7851–7858.

[37] X. Dong, C. Shi, W. Wen and J. Zhou, Multi-mission oriented joint optimization of task assignment and flight path planning for heterogeneous UAV cluster, *Remote Sens.* **15**(22) (2023) 5315.

[38] C. Fang, J. Mao, D. Li, N. Wang and N. Wang, A coordinated scheduling approach for task assignment and multi-agent path planning, *J. King Saud Univ.-Comput. Inf. Sci.* **36**(1) (2024) 101930.

Rui Hong received her B.E. degree from Beijing Institute of Technology, Beijing, China. She is currently pursuing M. Eng. degree in Artificial Intelligence at the Beijing Institute of Technology.

Jia Zhang received her B.S. and Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2006 and 2015, respectively. She is currently Associate Professor with the School of Automation, Beijing Institute of Technology.

Jia Zhang is the author of over 70 technical publications. Her current research interests include multi-objective optimization and decision-making, unmanned systems and multi-agent systems.

Minggang Gan received his B.E. and Ph.D. degrees in Control Science and Engineering from Beijing Institute of Technology, Beijing, China, in 2002 and 2007, respectively. During 2015–2016, he was Visiting Scholar in New York University. He is Professor with the State Key Laboratory of Autonomous Intelligent Unmanned Systems, School of Automation, Beijing Institute of Technology, Member of IEEE. His main research interest covers intelligent information processing and intelligent control.

Qing Wang received her B.E. and Ph.D. degrees from the University of Science and Technology Beijing, Beijing, China, in 2013 and 2018, respectively. She was Post-Doctoral Researcher with the School of Automation, Beijing Institute of Technology, Beijing, from 2018 to 2020. She is currently with Beijing Institute of Technology. Her current research interests include multi-agent systems, nonlinear systems, intelligent control and distributed optimization.

Bin Xin received his B.S. degree in information engineering and Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2004 and 2012, respectively. He was Academic Visitor with the Decision and Cognitive Sciences Research Centre, The University of Manchester, from 2011 to 2012. He is currently Professor with the School of Automation, Beijing Institute of Technology. His current research interests include evolutionary computation, combinatorial optimization and multi-agent systems.

Jie Chen received his B.Sc., M.Sc. and the Ph.D. degrees in control theory and control engineering from the Beijing Institute of Technology, Beijing, China, in 1986, 1996 and 2001, respectively. From 1989 to 1990, he was Visiting Scholar at the California State University, Long Beach, California, USA. From 1996 to 1997, he was a research fellow in the School of Engineering at the University of Birmingham, Birmingham, UK. He is

Professor with the School of Automation, Beijing Institute of Technology, where he serves as the Director of the National Key Laboratory of Autonomous Intelligent Unmanned Systems. He is also Professor with the Department of Control Science and Engineering, Harbin Institute of Technology, Harbin, Heilongjiang, China.

His current research interests include complex systems, multiagent systems, multiobjective optimization and decision and constrained nonlinear control. Professor Chen is currently Editor-in-Chief of Unmanned Systems and the Journal of Systems Science and Complexity. He is a Fellow of IEEE, IFAC, ACA, and a member of the Chinese Academy of Engineering.