Journal of Engineering, Project, and Production Management
2026, 16(4), 2026-90

An Emergency UAV Delivery Scheduling Method Combining MOPSO and Improved RRT Algorithms

Bo Li
Lecturer, School of Huahu Airport Economics, Changjiang Polytechnic, Wuhan, 430074, China, E-mail: lib.libo@outlook.com

Project Management
Received January 21, 2026; revised May 6, 2026; accepted May 7, 2026
Available online May 29, 2026

Abstract: Emergency logistics scenarios pose dual demands for scheduling efficiency and system robustness in collaborative scheduling of multiple Unmanned Aerial Vehicles (UAVs). Traditional integer programming and heuristic methods have high computational complexity when applied to large-scale problems, making it difficult to meet real-time requirements. Therefore, this study constructs a UAV delivery scheduling method that integrates improved Multi-Objective Particle Swarm Optimization (MOPSO) and improved rapidly-exploring random tree. This method enhances the rationality of task allocation and the efficiency of path planning. In terms of methodology, the task allocation stage introduces dynamic weight adjustment. A partition-based fast non-dominated sorting mechanism is further employed to ensure convergence speed and distribution balance of the solution set. The path planning process combines predictive-guided sampling and cost-based extension functions. Smoothing processing is applied to further enhance trajectory quality and improve the success rate of obstacle avoidance. Experiments show that the UAV task allocation method reduces the total completion time to less than 100 minutes during the task allocation phase, with a load balancing degree exceeding 93%. The UAV path planning method maintains an obstacle avoidance success rate of around 92% in dense obstacle environments, and the path smoothness is close to 0.9, both of which are superior to those of the comparison algorithms. In the fusion experiment, the proposed emergency UAV distribution scheduling method achieves a task success rate of 96.3% in low disturbance environments, with an average delay of only twelve minutes. It still maintains a success rate of 94.8% in high disturbance environments, with the lowest energy consumption of 111.8 kJ (kJ is a standard unit of energy in the International System of Units (SI). One kilojoule equals 1,000 joules). Research shows that this fusion method exhibits significant advantages in efficiency, trajectory quality, and system robustness, providing a practical and feasible technical path for UAV intelligent scheduling in complex emergency scenarios.

Keywords: UAV scheduling, emergency logistics distribution, multi-objective particle swarm optimization, rapidly-exploring random tree, task allocation, path planning.

DOI 10.32738/JEPPM-2026-90

---

1. Introduction

In recent years, sudden disasters and public events have occurred frequently. Whether it is natural disasters or man-made accidents, fast and efficient material distribution is crucial for saving lives and reducing property losses. As an emerging delivery tool, Unmanned Aerial Vehicles (UAVs) have shown great potential in emergency delivery due to their high flexibility, fast response speed, and ability to operate in complex terrain and harsh environments (Zhang et al., 2025; Wei et al., 2023). However, emergency UAV distribution scheduling is a complex optimization problem. It needs to consider multiple factors, including UAV flight path planning, task allocation, flight time constraints, payload capacity, and priority of delivery points. In practical scenarios, there may be obstacles in the delivery area, such as buildings, hills, trees, which can restrict the UAV's flight path, making path planning particularly critical. At the same time, the number of UAVs is limited, and there are numerous materials and locations that need to be delivered. How to allocate tasks effectively to maximize rescue efficiency is an urgent problem (Wan et al., 2024). Yin et al. (2025) proposed an accurate branch pricing cut algorithm that combines a novel column and cut generation scheme with a bidirectional labeling algorithm to optimize the truck-UAV collaborative delivery solution, reduce costs, and improve service satisfaction. The truck-UAV collaborative mode has reduced costs by 10.32% and service failure rate by 3.43% compared to the single-truck mode. Wu et al. (2023) proposed a hybrid strategy that integrates self-rescue, deployment, and cancellation to reduce the loss of delivery delay caused by logistics UAV interference. Compared with the global rescheduling strategy, this strategy reduced the average cost by 11.5%, improved the delivery completion rate by 9.5%, and increased customer satisfaction by 7%, effectively reducing the impact of interference.

However, in emergency scenarios with multiple UAVs and task points, task allocation and path planning problems are intertwined. Traditional integer programming and heuristic methods have high computational complexity in large-scale scenarios (e.g., involving dozens of UAVs and hundreds of task points), making it difficult to meet the dual requirements of real-time performance and robustness (Wan et al., 2024). Multi-Objective Particle Swarm Optimization (MOPSO) is a swarm intelligence optimization method developed on the basis of traditional PSO, which can simultaneously handle optimization problems of multiple objective functions. MOPSO is widely used in complex problems such as scheduling optimization, path allocation, and resource allocation (Hu et al., 2023). Li et al. (2024) proposed a mathematical model that integrates transportation difficulty, UAV capacity, and hub distribution to optimize UAV logistics hub location, reduce costs, and improve distribution efficiency. They also designed a diverse mixed PSO algorithm for solving the model. This algorithm has improved efficiency by 42.58% compared to other intelligent algorithms, effectively reducing logistics costs and improving consumer experience. Shu and Li (2023) proposed a joint unloading strategy for mobile edge computing Internet of Vehicles based on quantum PSO to meet the energy consumption and high cost challenges brought by computing-intensive applications in the Internet of Vehicles and ensure the quality of user experience. This strategy effectively reduced system overhead and task completion delay. The Rapidly-Exploring Random Tree (RRT) algorithm is a typical sampling path planning method that can quickly search for feasible paths in high-dimensional space. Wang et al. (2024) proposed a comprehensive RRT algorithm for UAV path planning, which integrates an improved artificial potential field method, pruning strategy, and Bezier curve optimization, to solve the problem of difficult and inefficient manual inspection caused by the numerous equipment and complex faults in the fracturing well site. The path generated by this algorithm was more in line with the motion characteristics of UAVs, with the advantages of a short path and less time consumption, and could efficiently complete obstacle avoidance inspection tasks. Lu et al. (2025) proposed an Improved RRT (IRRT) algorithm based on search rules and cross-entropy optimization to overcome the low computational efficiency and lack of asymptotic optimality of the RRT algorithm. This algorithm significantly improved path quality while ensuring computational efficiency. However, these methods still suffer from limitations in convergence efficiency, solution diversity, and adaptability when applied to large-scale emergency scenarios involving multiple UAVs and task points.

In summary, existing research task allocation methods have shown good performance in small-scale problems, they are difficult to extend to complex environments where multiple UAVs and task points coexist. MOPSO has attracted attention due to its parallel search and nonlinear adaptation capabilities. Although the traditional form has shortcomings, such as slow convergence speed and uneven distribution of solution sets. In terms of path planning, RRT produces long paths and insufficient smoothness due to its random nature, and its sampling efficiency decreases in obstacle-dense environments (Fang et al., 2024). In response to the above shortcomings, this study proposes a UAV scheduling method that integrates Improved MOPSO (IMOPSO) and IRRT, aiming to achieve efficient task allocation and reliable path planning. The research innovation lies in improving MOPSO by introducing a dynamic weight adjustment and a fast non-inferiority sorting mechanism. IRRT is designed for the path planning stage and achieves dual-optimization of task allocation and path planning through predictive-guided sampling, cost-based extension, and path smoothing.

In summary, existing methods show limited scalability and robustness in large-scale multi-UAV emergency scenarios. Traditional MOPSO suffers from slow convergence and uneven distribution of solutions, while RRT often generates long, unsmooth paths in dense environments. To address these issues, an integrated scheduling framework combining IMOPSO and IRRT is developed. IMOPSO improves convergence and solution diversity through dynamic weight adjustment and partition-based sorting, while IRRT enhances path feasibility via guided sampling and cost-based extension. The proposed method jointly optimizes task allocation and path planning, providing a scalable and robust solution for emergency UAV logistics.

2. Methodologies

2.1. UAV Task Allocation Method based on IMOPSO

In emergency logistics distribution scenarios, UAV scheduling problems include task allocation and path planning. In task allocation research, traditional integer programming and heuristic algorithms can handle small-scale problems, though when multiple UAVs and task points coexist, the computational complexity is often too high to meet the requirements of real-time and robustness in emergency scenarios. Multi-objective optimization algorithms, especially MOPSO, have become increasingly important tools for solving UAV scheduling problems due to their parallel search capabilities and adaptability to nonlinear objective functions (PinaPardo et al., 2024; Sorbelli et al., 2025). Assuming there is a UAV set I and a task point set J, the task allocation problem can be abstracted as the following multi-objective optimization model, as shown in Eq. (1).

$$\min F(x) = (f_1(x), f_2(x), f_3(x)) \tag{1}$$

In Eq. (1), $f_1(x)$ represents the total task completion time. $f_2(x)$ is the load balance degree. $f_3(x)$ task priority completion degree. Subject to UAV payload capacity, maximum flight range, and task assignment uniqueness constraints, each task point is assigned to at most one UAV. To visually present the spatial distribution and constraints of UAVs and task points in emergency scenarios, this study constructs a UAV emergency distribution task scenario modeling diagram, as shown in Fig. 1.

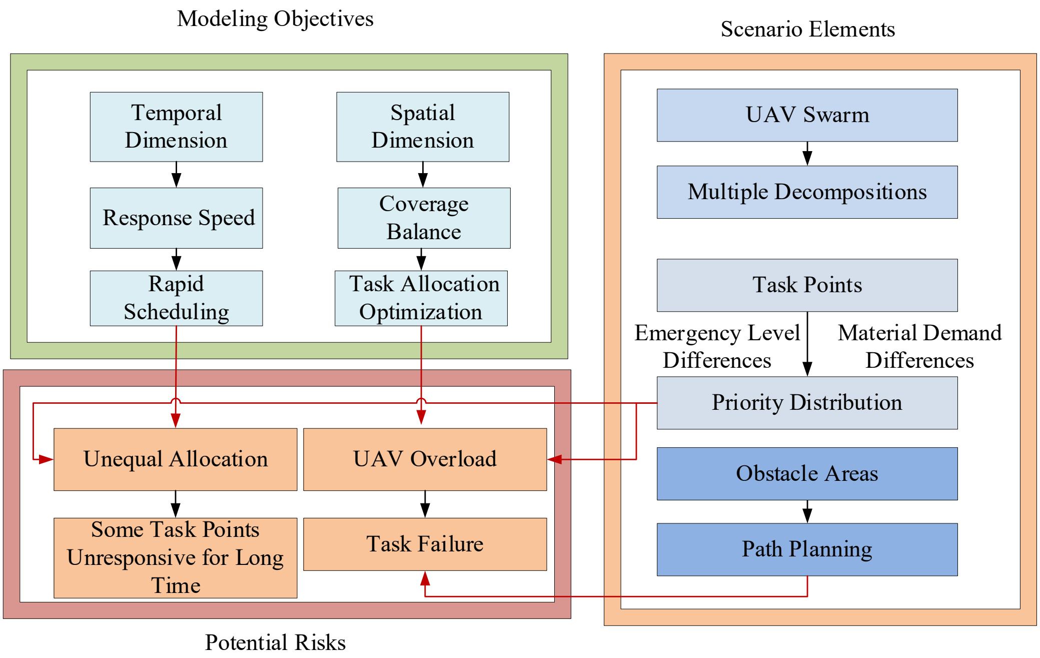


Fig. 1. Modeling diagram of the emergency delivery task scenario for UAVs

In Fig. 1, green frames represent UAVs, pink frames denote task points, and orange frames indicate obstacles or restricted regions. These elements describe the spatial relationships and constraints in the emergency delivery scenario. In Fig. 1, each UAV must deliver materials under limited payload and range constraints, with significant differences in urgency and material requirements at different task points. The priority distribution of task points directly determines the rationality of UAV scheduling schemes. If the allocation is uneven, some task points may go unaddressed for a long time, or individual UAVs may become overloaded, leading to task failures. After clarifying the spatial relationship between task points and UAVs, it is necessary to further demonstrate the solution steps for task allocation in a procedural manner. The task allocation process based on MOPSO is shown in Fig. 2.

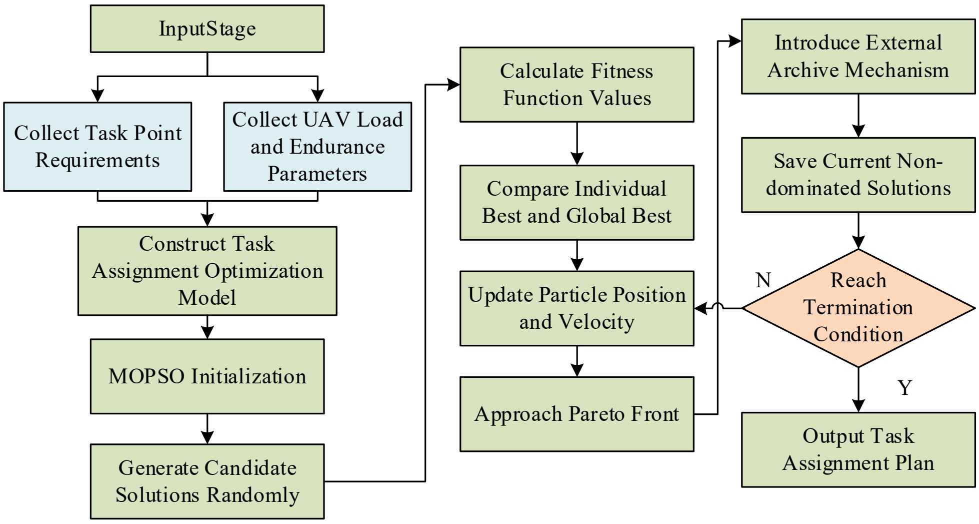


Fig. 2. Task allocation process based on MOPSO

In Fig. 2, during the input stage, the demand information for task points, the load, and endurance parameters of UAVs are collected to first construct a task allocation optimization model. Subsequently, it will enter the initialization phase of MOPSO, which randomly generates candidate solutions and evaluates their fitness values to assess their performance in terms of completion time, load balancing, and task priority. Next, the algorithm updates the position and velocity of particles by comparing individual historical optima with global optima, gradually approaching the Pareto front in the solution space (Chen et al., 2023). To maintain the diversity of the solution set, an external archive mechanism needs to be introduced to preserve the current optimal non-inferior solution. Finally, after the iteration termination condition is reached, the allocation scheme is selected from the external archive and output. The speed update of MOPSO is shown in Eq. (2).

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (P_i^{best} - X_i^t) + c_2 r_2 (G^{best} - X_i^t)\ X_i^{t+1} \tag{2}$$

In Eq. (2), V_i^t / V_i^{t+1} and X_i^t / X_i^{t+1} represent the velocity and position of particle i in the t and $t+1$ iterations. ω is the inertia weight. c_1 and c_2 are learning factors. r_1 and r_2 are random numbers. P_i^{best} and G^{best} are the optimal positions found so far for particle i and the entire population. The position update formula is shown in Eq. (3).

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{3}$$

However, traditional MOPSO is prone to local convergence in high-dimensional multi-UAV scheduling problems, leading to low search efficiency. Then, in the distribution of Pareto solution sets, traditional non-dominated sorting mechanisms often result in solution sets clustering in a few regions, reducing the diversity and selectivity of allocation schemes. In response to these issues, this study introduces a dynamic weight adjustment strategy that adapts the search direction based on the urgency of the task and the UAV's remaining payload. In addition, a partition-based fast non-dominated sorting mechanism is designed to divide the solution space into several sub-regions and independently screen Pareto solutions to improve the distribution uniformity of the solution set (Ben et al., 2023). The solution space is partitioned according to the ranges of objective values, such as task completion time and load balance. Within each partition, non-dominated sorting is performed independently, and a local archive is maintained to preserve representative Pareto solutions before they are merged into the global archive. The specific dynamic weight adjustment is shown in Eq. (4).

$$\omega(t) = \omega_{\max} - \left(\omega_{\max} - \omega_{\min}\right)\frac{t}{T} \tag{4}$$

In Eq. (4), $\omega(t)$ is the inertia weight of t. $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum values of inertia weights. t is the current iteration count. T is the total number of iterations. It makes the algorithm focus on global search in the early stage and enhances local convergence later. Therefore, this study proposes an IMOPSO algorithm, and the structure of the UAV task allocation method based on this algorithm is shown in Fig. 3.

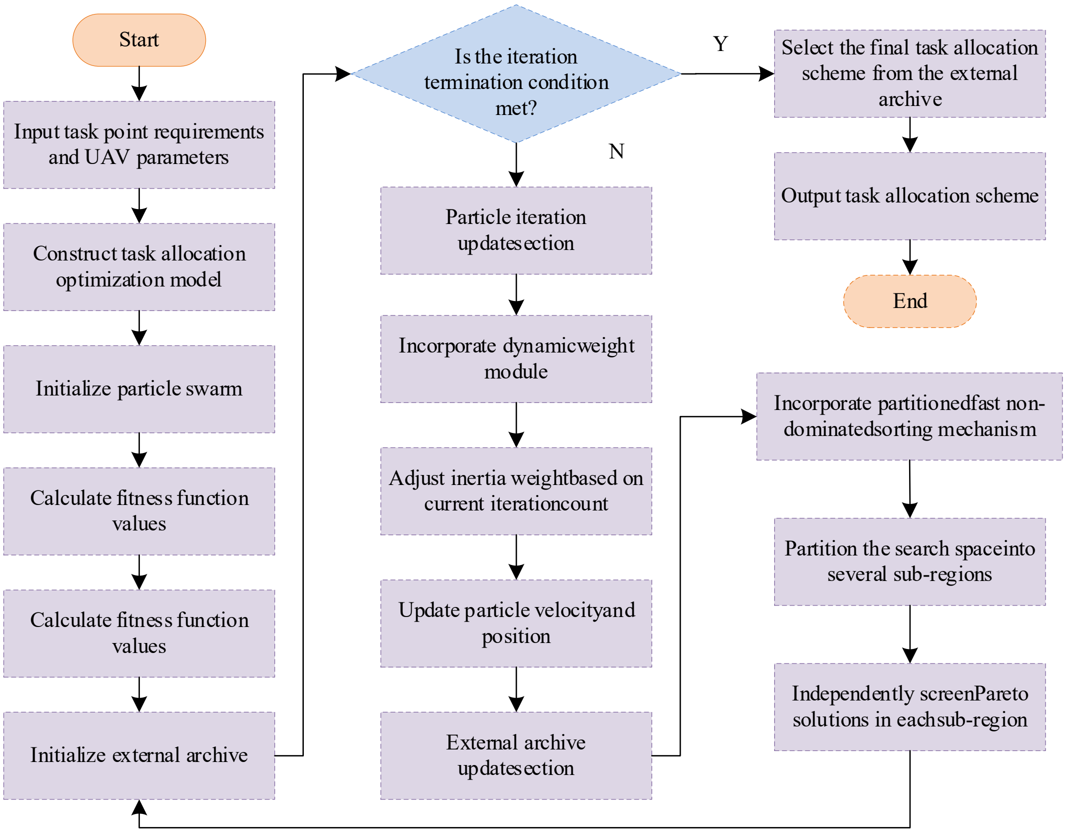


Fig. 3 UAV task allocation based on IMOPSO algorithm

In Fig. 3, the improved process introduces a dynamic weight module into the particle iteration update stage, enabling the algorithm to maintain strong global search ability in the early iterations and gradually enhance local convergence in later iterations, effectively avoiding falling into local optima. Additionally, the external archive update section has added a partition fast non-dominated sorting mechanism, which divides the search space into several subdomains and independently maintains the Pareto solution set in each subdomain. Finally, after several iterations, the IMOPSO outputs the optimal task allocation scheme. Specifically, the IMOPSO operates as follows. First, particles are initialized based on UAV capacity and task-priority constraints. Second, dynamic inertia weights are updated at each iteration to balance global exploration and local exploitation. Third, candidate solutions are evaluated using multiple objectives, including completion time and load balance. Fourth, a partition-based fast non-dominated sorting mechanism is applied to maintain solution diversity across sub-regions. Finally, the external archive is updated, and the iteration continues until convergence.

2.2. UAV Path Planning Method based on IRRT

To address the task allocation problem in UAV emergency distribution, this study improves MOPSO to achieve efficient matching between task points and UAVs. However, completing task allocation does not necessarily mean the UAV can reach the target smoothly. UAVs still need to fly in complex environments during actual mission execution, avoiding static obstacles such as mountains and buildings, while dealing with dynamic interference such as sudden weather or moving targets (Zhao et al., 2023). Therefore, path planning is a core component in ensuring mission success. To model the UAV's path, the trajectory length is defined as shown in Eq. (5).

$$L_{\text{path}} = \sum_{k=1}^{n-1} \| P_{k+1} - P_k \| \tag{5}$$

In Eq. (5), L_{path} is the total trajectory length of the UAV from the starting point to the target point. The total number of n path points. The coordinates of the k-th trajectory point on the P_k-path. The specific calculation of planning time is shown in Eq. (6).

$$T_{\text{plan}} = T_{\text{end}} - T_{\text{start}} \tag{6}$$

In Eq. (6), T_{plan} represents the total time spent on path planning. T_{end} and T_{start} plan the start and end times. The specific calculation of obstacle avoidance success rate is shown in Eq. (7).

$$S_{\text{succ}} = \frac{N_{\text{succ}}}{N_{\text{total}}} \times 100\% \tag{7}$$

In Eq. (7), N_{succ} represents the number of successful obstacle avoidance attempts. N_{total} is the total number of attempts. The optimization objective of UAV path planning is to minimize while ensuring safety. In path planning, RRT is a typical sampling algorithm that can quickly search and find a feasible path in high-dimensional space and is widely used in motion planning problems such as UAV's and robots (Bello and Oladipo, 2024). Therefore, this study uses RRT to plan the UAV path, as shown in Fig. 4.

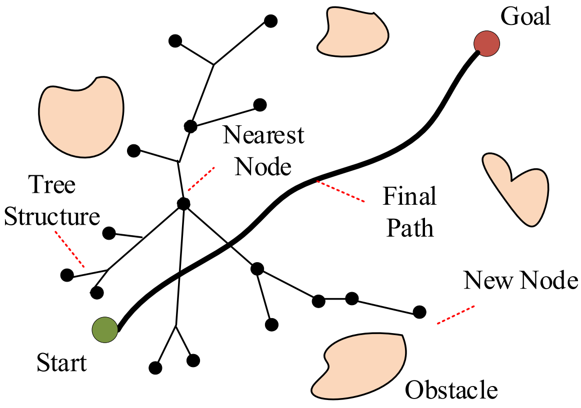


Fig. 4. Schematic diagram of the RRT algorithm principles

In Fig. 4, the algorithm first randomly samples a point in the search space, then finds the nearest node to that point in the existing tree structure, and expands in the direction of the sampling point by the preset step size to generate new nodes. Through continuous iteration, the tree structure gradually covers the entire feasible space, ultimately yielding a workable path from the starting point to the target. However, due to sampling randomness, the paths generated by RRT are often long and not smooth enough, resulting in low sampling efficiency in environments with dense obstacles. At the same time, they are highly sensitive to step size and sampling strategy, making it difficult to meet the dual requirements of timeliness and safety in emergency distribution. RRT extends the tree structure via random sampling, and its node update calculation is given by Eq. (8).

$$X_{\text{new}} = X_{\text{near}} + \Delta t \frac{X_{\text{rand}} - X_{\text{near}}}{\| X_{\text{rand}} - X_{\text{near}} \|} \tag{8}$$

In Eq. (8), X_{rand} is a random sampling point. X_{near} is the nearest neighbor. Δt is the step size (Wu et al., 2024). Therefore, this study improves RRT by introducing a prediction-oriented sampling mechanism and a cost-based path selection strategy, combined with pruning and smoothing. The structure of the IRRT path planning method is shown in Fig. 5.

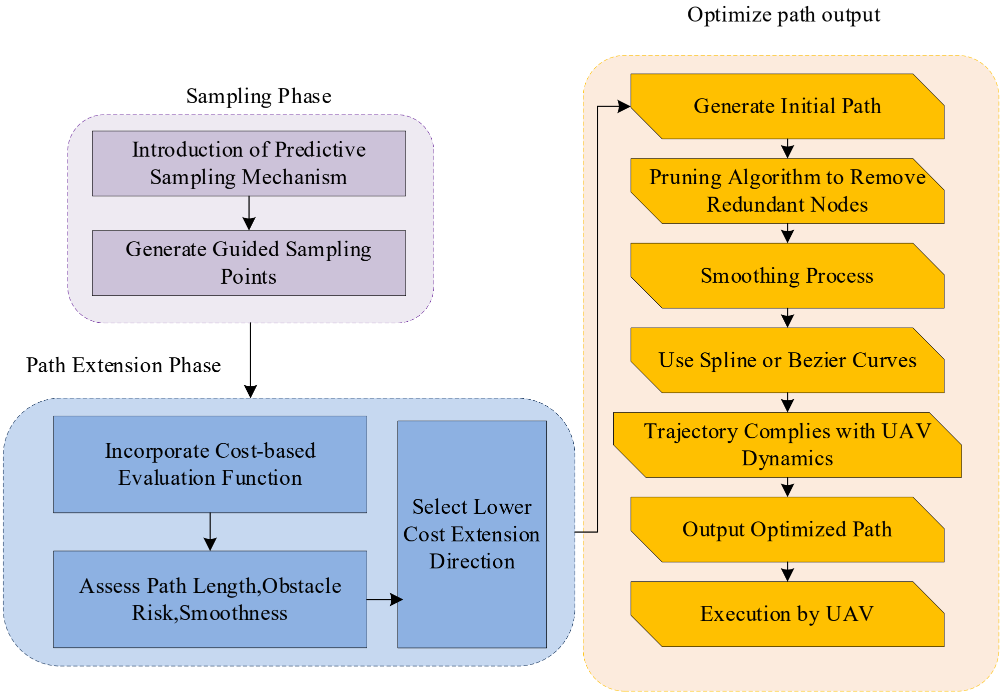


Fig. 5. Structure diagram of IRRT path planning method

In Fig. 5, a predictive-guided sampling mechanism is introduced at the sampling stage, which utilizes the task target direction and obstacle distribution information to generate more guided sampling points, reducing the efficiency loss caused by random sampling. The prediction guidance is based on the relative direction between the UAV's current position and the target, combined with information on obstacle distribution, to bias sampling toward achievable, low-risk areas. Furthermore, during path extension, a cost-based evaluation function is added to weight and comprehensively evaluate path length, obstacle-avoidance risk, and smoothness, thereby prioritizing the low-cost extension direction. After generating the initial path, the process further uses a pruning algorithm to remove redundant nodes, applies spline- or Bezier-based smoothing to reduce sharp turns, and checks for collisions to ensure the smoothed path does not intersect with obstacles. Lastly, the algorithm outputs the optimized path for UAV execution. The path cost function is shown in Eq. (9).

$$C(P)=\alpha L_{\text{path}}+\beta R_{\text{obs}}+\gamma S_{\text{smooth}} \tag{9}$$

In Eq. (9), R_{obs} represents the cost of obstacle avoidance risk. S_{smooth} is the smoothness index. α, β, and γ are weight factors. The overall structure of the emergency UAV distribution scheduling method combining MOPSO and IRRT algorithms is shown in Fig. 6.

In Fig. 6, the dispatch center first receives task requirements and UAV status information in emergency scenarios, then uses an IMOPSO algorithm to perform global allocation of UAV task points and then outputs the allocation matrix. After obtaining the allocation results, each UAV enters the path-planning stage, where the IRRT algorithm generates practicable trajectories from the starting point to the mission point. Thus, after dual optimization of task allocation and path planning, the scheduling system produces a complete set of UAV scheduling schemes to ensure that tasks are completed with the highest security in the shortest time.

3. Results

3.1. Experimental Setup

To verify the effectiveness and superiority of the UAV emergency distribution scheduling method combining IMOPSO and IRRT, this study conducts multiple simulation experiments on a unified software and hardware platform. In terms of hardware, the experimental platform uses an Intel Xeon Gold 6248 CPU, paired with 128 GB of memory, and an NVIDIA RTX 3090 GPU, providing sufficient computing power for large-scale task allocation and path-planning simulation. In terms of software, the experimental operating system is Ubuntu 20.04 LTS, and the main algorithm implementation is based on Python 3.9 and MATLAB R2022b. At the same time, partial 3D scene verification is performed in a ROS (Robot Operating System) + Gazebo environment to ensure the algorithm's feasibility at both theoretical and engineering levels. In terms of parameter settings, the IMOPSO and IRRT have been finely configured. The specific environmental parameters are shown in Table 1. This study considers medium-scale multi-rotor UAV's used in urban delivery. The payload is set to three to eight kg, and flight endurance is limited to 30 to 40 minutes, corresponding to a range of ten to fifteen km. Energy consumption depends on flight distance, payload, and trajectory complexity. All tasks are constrained by payload capacity and maximum flight range.

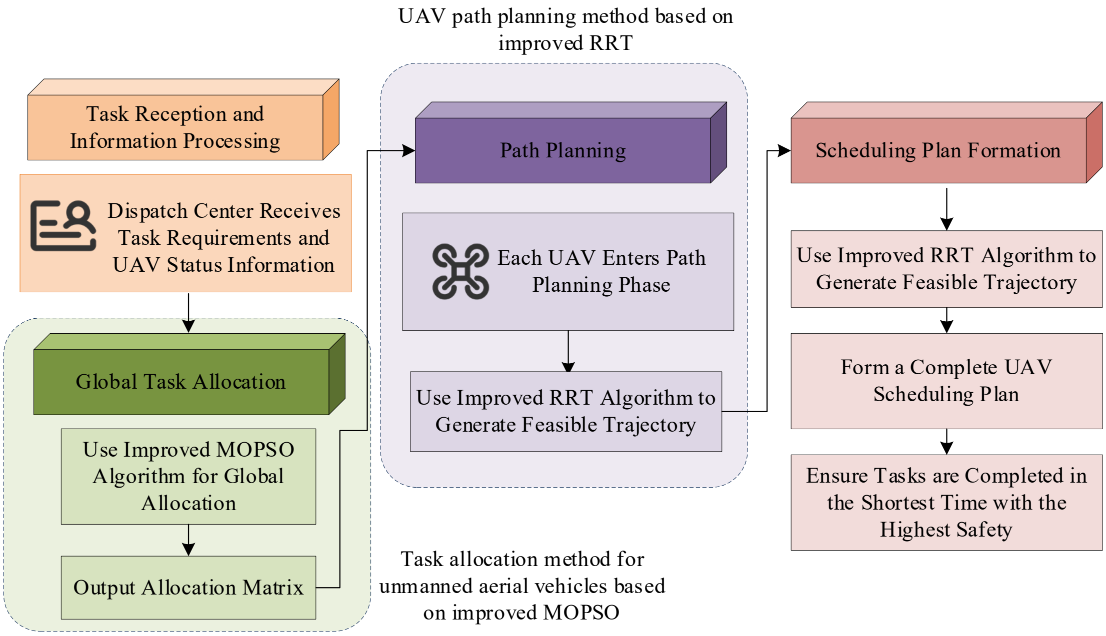


Fig. 6. Overall framework diagram of emergency UAV delivery scheduling method

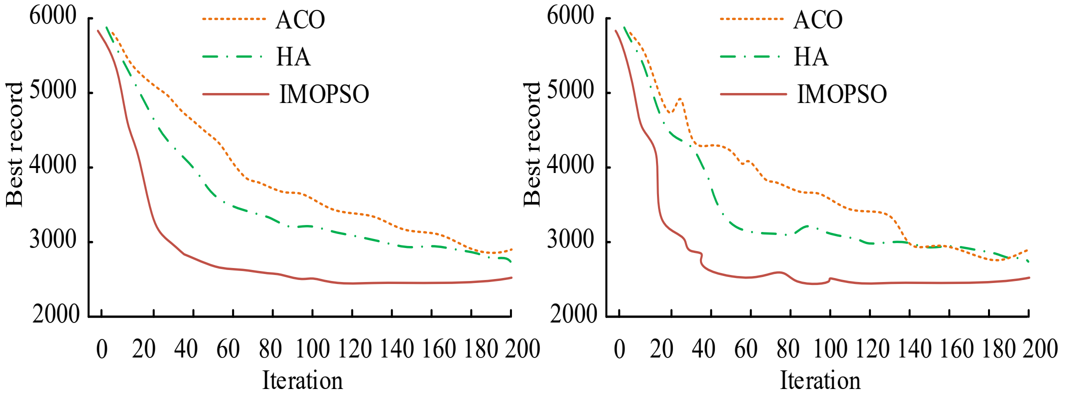
Table 1. Experimental environment configuration

Category	Item	Description
Hardware	CPU	Intel Xeon Gold 6248 @ 2.5 GHz
	RAM	128 GB
	GPU	NVIDIA RTX 3090, 24 GB
Software	Operating System	Ubuntu 20.04 LTS
	Simulation/Programming	Python 3.9, MATLAB R2022b, ROS + Gazebo
MOPSO Params	Population Size	50
	Max Iterations	200
	Inertia Weight	0.4–0.9
	Learning Factors	$c1=1.5,c2=1.7$
	External Archive Size	100
RRT Params	Max Sampling Number	5000
	Step Size	5m
	Guided Sampling Ratio	0.7
	Cost Function Weights	$\alpha=0.5,\beta=0.3,\gamma=0.2$

To ensure reproducibility, this study uses publicly available datasets for simulation verification. The task points and requirement data are sourced from MIT's UAV Delivery Dataset (UDD), which includes various types of emergency material delivery tasks in urban areas. Environmental barriers and flight constraints are used in the NASA Urban Air Mobility Simulation Dataset (NASA UAM Dataset) to construct flight environments for complex scenarios. The MIT UDD dataset provides urban emergency delivery scenarios with dozens of task points and heterogeneous priority requirements, while the NASA UAM dataset offers dense obstacle layouts and realistic airspace constraints. These characteristics are highly consistent with the multi-UAV, multi-task, and obstacle-rich emergency scheduling scenarios addressed in this study. For illustration, the UDD dataset includes delivery task records with target coordinates, task priority levels, material demand, and service time constraints, which are used to construct multi-UAV task allocation scenarios. The NASA UAM dataset provides urban airspace information such as obstacle distributions, no-fly regions, and flight corridor constraints, which are used to model complex path planning environments. In this study, task nodes from UDD are mapped onto obstacle-constrained scenes generated from NASA UAM to form integrated emergency scheduling test cases.

3.2. Performance Verification of IMOPSO UAV Task Allocation Method

In fairness, all comparison algorithms adopt parameter settings recommended in their original studies, and key parameters are adjusted under the same tuning principles and iteration budgets. In all performance curves, each line represents the average result over multiple simulation runs under the same parameter settings, ensuring result stability and comparability. To verify the optimization performance of the proposed IMOPSO in UAV task allocation challenges, this study compares the Ant Colony Optimization (ACO) with the Hungarian Algorithm (HA). The obtained results are shown in Fig. 7. In this study, the simple model refers to scenarios with fewer UAVs, task points, and limited environmental constraints, while the complex model involves a larger number of UAVs, denser task points, and stricter load and scheduling constraints.

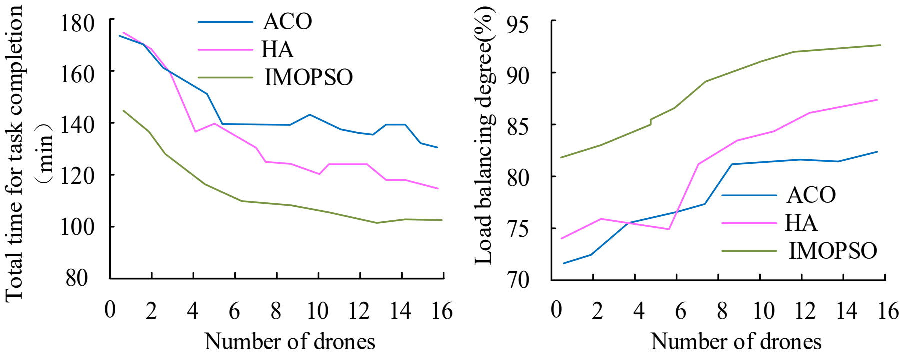


(a) Comparison of Algorithm Convergence Curve under Simple Model

(b) Comparison of convergence curves of algorithms under complex models

Fig. 7. Comparison of convergence performance of various algorithms under simple and complex models

In Fig. 7(a), under the simple model, IMOPSO converges significantly faster and achieves a lower final optimal value. In the first 40 iterations, the convergence speed of ACO and HA is slow, and the optimal fitness value remains around 4,000, while IMOPSO has decreased to around 3,000. As the number of iterations increases, the decrease in ACO and HA gradually slows. At 200 iterations, their optimal values remain around 2,800 and 2,600, while IMOPSO eventually converges to about 2,300, achieving better task allocation performance. This indicates that IMOPSO can effectively avoid getting stuck in local optima and quickly approach the global optimum in low complexity scenarios. In Fig. 7(b), under complex models, the convergence performance gap between the algorithms widens further. ACO exhibits fluctuations in the early stages of iteration, with convergence values still hovering between 4,000 and 5,000 within the first 60 generations. Although HA improves in the middle and later stages, the overall convergence curve is high and eventually stabilizes at around 3,000. In contrast, IMOPSO can quickly decrease to around 3,500 in the first 30 generations of iteration, and converges to about 2,500 after 100 generations, which is overall better than the compared algorithms. This performance gain is attributed to the dynamic weight adjustment strategy, which accelerates early global exploration while preventing premature convergence in later iterations. To further verify IMOPSO's comprehensive performance in UAV task allocation, the experiment compares the performance of various algorithms from the perspectives of total task completion time and load-balancing degree, as shown in Fig. 8.



(a) Curve of total task completion time varying with the number of drones

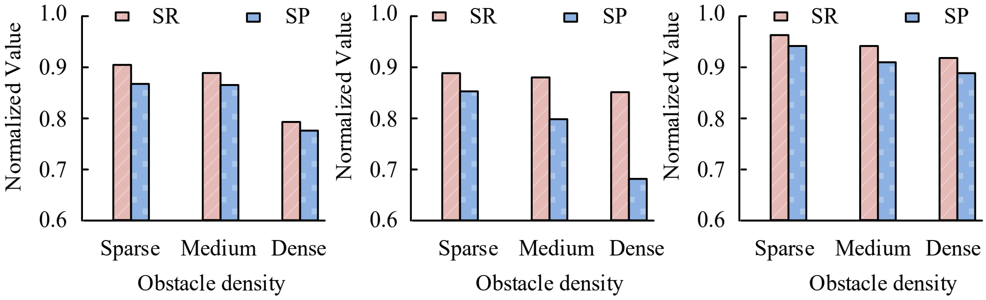
(b) The curve of load balancing degree changing with the number of drones

Fig. 8. Comparison of task completion time and load balancing across drone swarm sizes

When the number of UAVs reaches twelve, the completion time of ACO is still higher than 120 minutes, and HA converges around 110 minutes. In contrast, during the task allocation stage, IMOPSO stabilizes below 100 minutes, which is about 20% lower than ACO and about 10% lower than HA. This indicates that IMOPSO can better shorten the overall scheduling time in the allocation strategy, reflecting the advantages of global optimization and a dynamic weighting mechanism. In Fig. 8(b), as the number of UAVs increases, the load balancing degree gradually improves, and IMOPSO maintains optimal performance in all stages. When there are 6 UAVs, the load balancing degrees of ACO and HA are 74% and 77%, while IMOPSO has reached 83%. When the number of UAVs increases to fourteen, ACO and HA increase to about 82% and 86%, but IMOPSO reaches over 93%. Overall, IMOPSO can effectively shorten task completion time and significantly improve the balance and robustness of multi-UAV task allocation, making it more suitable for complex emergency distribution scenarios. The improvement in completion time and load balance is primarily due to the partition-based non-dominated sorting mechanism, which reduces solution clustering and enables more balanced task distribution among UAVs.

3.3. Performance Verification of UAV Path Planning Method

To verify the path planning performance of IRRT in different obstacle environments, this study compares the Probabilistic Roadmap Method (PRM) and the A-star Algorithm (A*). The experiment evaluates the obstacle avoidance Success Rate (SR) and Path Smoothness (SP) along two dimensions, as shown in Fig. 9. The path smoothness index ranges from zero to one, with higher values indicating fewer sharp turns and smoother trajectories. It is calculated based on the cumulative change in heading angles along the planned path.



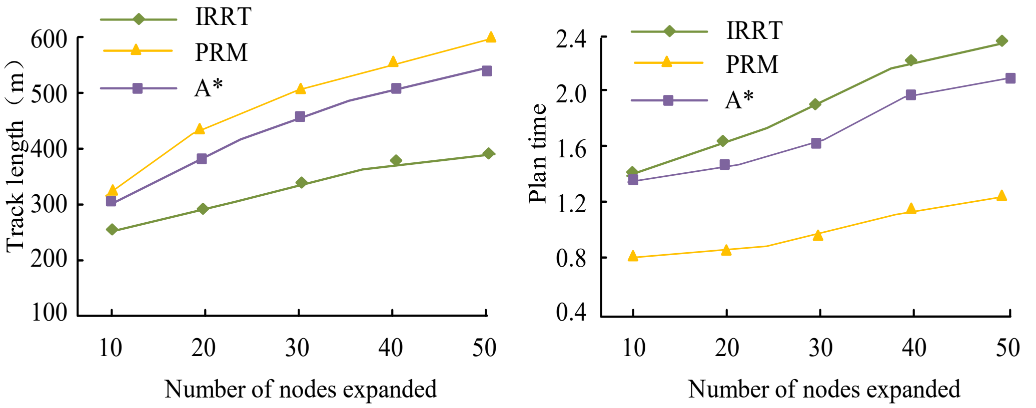
(a) PRM (b) A* (c) IRRT

Fig. 9. Performance comparison of different path planning algorithms under three types of obstacle densities

In Fig. 9(a), the obstacle avoidance success rate of PRM remains above 0.9 in sparse and moderate environments, but decreases to about 0.85 in dense environments. Its path smoothness is close to 0.88 in sparse scenes, but drops to about 0.8 in dense scenes. In Fig. 9(b), the obstacle avoidance success rate of A* is approximately 0.88, 0.85, and 0.82 in the three environments, overall lower than PRM. The smoothness of the path decreases more significantly, from 0.83 in sparse scenes to about 0.7 in dense scenes. In Fig. 9(c), IRRT maintains optimal performance in all three types of environments. In sparse environments, the obstacle avoidance success rate is as high as 0.97, and the path smoothness is close to 0.95. Even in dense environments, SR remains around 0.92, and SP remains around 0.9. Compared to PRM and A*, IRRT has the smallest decrease, mainly due to reduced ineffective exploration in predictive-guided sampling and improved path viability and continuity. The results in Fig. 9 indicate that IRRT maintains the most stable path planning performance across different obstacle densities. In dense environments, its obstacle avoidance success rate remains around 0.92, and its path smoothness stays close to 0.90, whereas PRM declines to about 0.85 and 0.80, and A* decreases further to about 0.82 and 0.70. These results show that IRRT can generate safer, smoother, achievable paths under complex spatial constraints. Further analysis of the efficiency and path quality differences among different path planning algorithms is presented as the number of task points increases, as shown in Fig. 10.

In Fig. 10(a), as the number of task points increases, the trajectory lengths of the three algorithms all show an upward trend. When the task point is 10, the trajectory lengths of the three algorithms are all around 300 meters, with IRRT being the shortest, only about 280 meters. When the task point is expanded to 50, the trajectory lengths of PRM and A* reach about 580 meters and 540 meters, while IRRT remains at around 400 meters, significantly reducing the path length. In Fig. 10(b), as the number of task points increases, the planning time of the three algorithms increases as well, but the differences are quite significant. The planning time of PRM is the lowest, remaining around 1.2s at task point 50, reflecting its high efficiency in static environments. However, PRM sacrifices path quality for speed, leading to longer trajectories. The planning time of A* is moderate, increasing from 1.0s to 2.0s, demonstrating good scalability. In contrast, the planning time of IRRT increases from 1.2s to 2.4s with the number of task points, slightly higher than A*. As shown in Fig. 10, when the number of task points increases to 50, the path generated by IRRT is about 400m, shorter than PRM at about 580m and A* at about 540m. Although its planning time rises to 2.4s, slightly higher than PRM and A*, the reduced trajectory length and improved path quality provide better support for practical UAV execution.

Based on the verification of task allocation and path planning methods separately, this study further integrates the two to construct three scheduling schemes, including ACO-PRM, HA-A*, and IMOPSO-IRRT. Table 2 compares its comprehensive performance in different environments.



(a) The curve of trajectory length changing with the number of task points

(b) The curve of the change in planning time with the number of task points

Fig. 10. Comparison of track length and planning time for various algorithms at different task points

Table 2. Performance comparison of various fusion scheduling algorithms in different environments

Environment	Method	Total completion time (min)	Average delay(min)	Task success rate (%)	Energy consumption (kJ)
Low disturbance environment	ACO-PRM	149.1	15.3	90.4	121.2
	HA-A*	143	14.1	92.8	115.7
	IMOPSO-IRRT	131.8	12.0	96.3	107.3
High disturbance environment	ACO-PRM	155.6	17.5	86.7	128.1
	HA-A*	150.3	16.3	89.5	121.0
	IMOPSO-IRRT	137.5	13.6	94.8	111.8

In Table 2, under low disturbance conditions, IMOPSO-IRRT performs the best in all indicators. The total completion time is only 131.8 minutes, which is about 17.3 minutes and 11.2 minutes shorter than ACO-PRM and HA-A*. The average delay is only 12.0 min, significantly better than ACO-PRM's 15.3 minutes and HA-A*'s 14.1 minutes. In terms of task success rate, IMOPSO-IRRT reaches 96.3%, while its energy consumption is the lowest at 107.3 kJ. In high-disturbance environments, the performance of each algorithm decreases slightly, but the gap widens. The task success rate of ACO-PRM has decreased to 86.7%, HA-A* is only 89.5%, while IMOPSO-IRRT remains at 94.8%. Its energy consumption is 111.8 kJ, a reduction of approximately 16.3 kJ and 9.2 kJ compared to ACO-PRM and HA-A*. The reason is that IMOPSO-IRRT combines the global task allocation capability of IMOPSO with the efficient path planning of IRRT, enabling it to achieve better task scheduling and energy consumption control in complex environments, thereby ensuring high task completion rates and low latency. This advantage can be attributed to the complementary mechanisms of IMOPSO and IRRT. IMOPSO reduces redundant task assignments through global load balancing, thereby avoiding unnecessary UAV flights. Meanwhile, IRRT generates shorter and smoother trajectories with lower maneuvering costs, which helps maintain low energy consumption under both low and high disturbance environments. The integrated scheduling results further verify the practical effectiveness of the proposed framework. In low-disturbance environments, IMOPSO-IRRT achieves a task success rate of 96.3% with an average delay of 12.0 minutes. In high-disturbance environments, it still maintains 94.8% task success and the lowest energy consumption of 111.8 kJ. These results confirm that the proposed method can coordinate task allocation and path planning more effectively than ACO-PRM and HA-A* in realistic emergency delivery scenarios. In emergency scenarios, additional constraints are considered, including task urgency levels, limited UAV availability, strict time windows, and dynamic environmental disturbances such as obstacles and interference. These factors directly affect task prioritization and path workability. Although the proposed method is designed for emergency logistics, it can be extended to non-emergency scenarios by relaxing urgency and time constraints, making it applicable to general UAV delivery optimization problems.

4. Conclusion

In response to the challenges of task allocation and path planning separation, as well as insufficient overall robustness in emergency UAV distribution, this study proposed a scheduling method that integrates IMOPSO and IRRT. It aimed to improve the scheduling efficiency, task completion rate, and path feasibility of multiple UAVs in complex environments. At the methodological level, the task allocation section introduced dynamic weight adjustment and a fast non-dominated sorting mechanism for partitioning. The path planning part used predictive-guided sampling, cost-based extension functions, and smoothing post-processing. In the experiment, in terms of task allocation, IMOPSO achieved improvements in both completion time and balance compared to traditional methods. When the number of UAVs was 12, its completion time was shortened by about 18% compared to ACO and about 9% compared to HA, while the overall load balancing was improved by more than 10%. In the path planning experiment, IRRT maintained an obstacle avoidance success rate of over 90% in dense obstacle environments and improved path smoothness by an average of 15% to 20% compared to PRM and A*, significantly reducing redundant trajectories while ensuring safety. In the fusion scheduling experiment, the energy consumption of IMOPSO-IRRT in high-disturbance environments decreased by about 13% compared to ACO-PRM and about 8% compared to HA-A*, and showed a more stable delay-control trend. Research shows that the proposed UAV delivery scheduling method outperforms comparative algorithms in terms of efficiency, robustness, and resource utilization, and can effectively improve the overall performance of UAV emergency delivery. Beyond experimental performance, this study contributes to the theoretical understanding of multi-UAV emergency scheduling by formulating task allocation and path planning as a coupled optimization problem. The proposed IMOPSO-IRRT framework enhances convergence efficiency, solution diversity, and trajectory feasibility under disturbance-prone conditions. These characteristics extend existing UAV scheduling approaches toward more robust and scalable collaborative systems. The results indicate that integrated optimization is essential for complex emergency logistics. From a managerial perspective, the proposed method supports decision-makers in dynamically assigning delivery tasks based on urgency and resource availability, rather than relying on static or experience-based scheduling. It also enables more efficient route planning under constraints, helping managers reduce delays, improve resource utilization, and enhance response efficiency in emergency operations. However, as the task scale increases, the algorithm's computational burden grows, and path-smoothing processing may lack sufficient accuracy in extreme environments. Despite its effectiveness, the proposed method still faces limitations in computational cost as the task scale further increases, and the accuracy of path smoothing may degrade in extreme environments. Future work will focus on improving computational efficiency through distributed or parallel optimization strategies. In addition, incorporating real-time environment perception and uncertainty modeling will be explored to further enhance robustness in highly dynamic emergency scenarios. Beyond performance improvements, the results highlight the importance of integrating task allocation and path planning into a unified optimization framework. This integrated strategy provides a new perspective for solving large-scale multi-UAV scheduling problems under uncertainty. Future research may explore real-time adaptive scheduling, distributed optimization, and integration with real-world sensing systems to enhance applicability in highly dynamic environments.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

Declaration of Artificial Intelligence (AI) Tools

The author used DeepSeek solely for language editing and readability improvement. The author has reviewed and verified all content and takes full responsibility for the accuracy and integrity of the manuscript.

References

Bello, R. W., and Oladipo, M. A. (2024). Mask YOLOv7-based drone vision system for automated cattle detection and counting. *Artificial Intelligence and Applications,* 2(2). doi: 10.47852/bonviewAIA42021603.

Ben, N., Xue, B., and Zhou, T. (2023). Interactive bacterial foraging optimization algorithm for aircraft base maintenance technician scheduling. *Journal of Industrial Engineering and Engineering Management*, 37(5), 247-258. doi: 10.13587/j.cnki.jieem.2023.05.021.

Chen, H., Wei, F., Han, C., and Zhao, L. (2023). UAV path planning strategy based on threat avoidance in multiple extended target tracking optimization*. Journal of Radars*, 12(3), 529-540. doi: 10.12000/JR22116.

Fang, C., Yang, F., and Pan, Q. (2024). Multi-UAV collaborative path planning based on multi-agent soft actor critic. *Scientia Sinica Informationis,* 54(8), 1871-1883. doi: 10.1360/SSI-2024-0050.

Hu, L., Zhao, H., Nan, Y., and Guo, J. (2023). Unmanned aerial vehicle path planning method based on search rule and cross. *Journal of Electronics & Information Technology,* 45(6), 2144-2152. doi: 10.11999/JEIT220579.

Li, Z., Xu, G., and Chen, W. (2024). Using improved particle swarm optimization algorithm for location problem of drone logistics hub. *Computers, Materials & Continua,* 78(1), 935-957. doi: 10.32604/cmc.2023.046006.

Lu, Y., Liu, J., and Zhang, M. (2025). A model-assisted federated reinforcement learning method for multi-UAV path planning. *Journal of Electronics & Information Technology*, 47(5), 1368-1380. doi: 10.11999/JEIT241055.

PinaPardo, J., Silva, D., Smith, A., and Lopez, C. (2024). Fleet resupply by drones for last-mile delivery. *European Journal of Operational Research,* 316(1), 168-182. doi: 10.1016/j.ejor.2024.01.045.

Shu, W., and Li, Y. (2023). Joint offloading strategy based on quantum particle swarm optimization for MEC-enabled vehicular networks. *Digital Communications and Networks*, 9(1), 56-66. doi: 10.1016/j.dcan.2022.03.009.

Sorbelli, F., Coro, F., Das, S., and D’Angelo, G. (2025). Dispatching point selection for a drone-based delivery system operating in a mixed Euclidean-Manhattan grid. *Annals of Operations Research,* 351(1), 203-222. doi: 10.1007/s10479-023-05458-4.

Wan, X., Pu, H., Ran, Y., and Chen, L. (2024). Optimization of railway alignments in regions with multiple constraints based on a 3D rapidly exploring random tree algorithm. *Journal of Railway Science and Engineering,* 21(8), 3142-3152. doi: 10.19713/j.cnki.43-1423/u.T20231814.

Wan, X., Wu, B., Yi, J., and Zhou, H. (2024). UAV path planning method for passive radar transmitter localization. *Journal of Electronics & Information Technology,* 46(5), 2057-2064. doi: 10.11999/JEIT231293.

Wang, S., Zhang, L., and Liu, S. (2024). Safety inspection path planning for unmanned aerial vehicle in oil and gas fracturing unit. *Science Technology and Engineering,* 24(25), 10832-10840. doi: 10.12404/j.issn.1671-1815.2309985.

Wei, M., Sun, Y., Sun, B., and Liu, H. (2023). UAV distribution route and flight path collaborative planning based on deep reinforcement learning. *China Safety Science Journal,* 33(8), 68-76. doi: 10.16265/j.cnki.issn1003-3033.2023.08.0038.

Wu, T., Gou, L., Wang, J., and Zhou, Q. (2023). Research on scheduling of UAV after disturbed delay in urban distribution. *Computer Engineering and Applications,* 59(15), 310-317. doi: 10.3778/j.issn.1002-8331.2205-0118.

Wu, X., Wang, C., Zhao, J., and Li, M. (2024). Improved sparrow algorithm for UAV 3D path planning. *Science Technology and Engineering*, 24(15), 6534-6542. doi: 10.12404/j.issn.1671-1815.2304154.

Yin, Y., Li, D., Wang, D., and Zhang, Q. (2025). Truck-drone pickup and delivery service optimization with availability profiles. *Naval Research Logistics*, 72(4), 534-565. doi: 10.1002/nav.22238.

Zhang, X., Xie, K., Huang, P., and Liu, J. (2025). Onsite spatial perception and UAV acquisition path planning. *Scientia Sinica Informationis,* 55(5), 1051-1072. doi: 10.1360/SSI-2024-0234.

Zhao, J. W., Gao, F. F., Jia, W. M., and Song, S. (2023). Integrated sensing and communications for UAV communications with jittering effect. *IEEE Wireless Communications Letters,* 12(4), 758-762. doi: 10.1109/LWC.2023.3243590

Bo Li graduated in Computer Science and Technology and currently serves as a lecturer in the Intelligent Technology of Postal Express program at the School of Huahu Airport Economics, Changjiang Polytechnic. His research interests include logistics, big data analytics, low-altitude logistics system optimization, UAV scheduling and path planning, and multi-objective optimization algorithms and their applications. He is dedicated to applying big data technologies and artificial intelligence methods to modern logistics, especially intelligent decision-making and resource scheduling in low-altitude logistics. As first or corresponding author, he has published several academic papers in related fields. He has participated in multiple provincial and municipal research projects on logistics informatization and UAV delivery, demonstrating strong interdisciplinary teaching and research capabilities.