Emergency Rescue Path Planning for Urban Emergencies Based on Improved GA and PSO

Zhisong Wu
School of Public Administration, Guangxi Police College, Nanning, 530028, China
E-mail: zswu7788@outlook.com

Keywords: GA, PSO, tent chaotic mapping, the combination strategy of elite retention and roulette wheel, metropolis criterion

Received: August 11, 2025

With the acceleration of urbanization and the frequent urban emergencies, traditional rescue path planning methods have response delay and low path efficiency. To improve the efficiency of urban emergency rescue and resource scheduling capabilities, a path planning technique for urban emergency rescue based on improved Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) is built. The research introduces Tent chaotic mapping initialization particles, adaptive segmented inertia weights and exponential learning factors for the PSO algorithm to enhance global search capabilities, and adds traction acceleration terms to avoid local optimality. For the GA, a combination strategy of elite retention and roulette is used to improve selection efficiency, and the Metropolis criterion is combined to optimize the cross-mutation operation, and an adaptive variable neighborhood search mechanism is introduced to strengthen local search. In the experimental setting, the Python simulation platform is used to compare with baseline methods such as RRT, D* Lite and MOPSO. The test indicators include response time, planning success rate, path length, number of convergence iterations, etc. Experimental results showed that under six concurrent events, the response time of the research method was 6 seconds, which was significantly better than that of the comparison method. When the dynamic obstacle density was 40/km², the planning success rate reached 90.2%. When the scene complexity was 200 nodes, the single planning calculation time was 150ms. The research method converged at the 100th iteration, and the fitness change rate was reduced to 1.3%, showing faster convergence speed and better stability. The above results show that the proposed method is superior to traditional methods in terms of timeliness, robustness and optimization capabilities, which is suitable for emergency rescue path planning in complex urban scenarios.*

Povzetek: Predlagana je izboljšana hibridna metoda GA–PSO za načrtovanje poti pri urbanem reševanju, ki v simulacijah dosega hitrejši odziv, višjo uspešnost in stabilnejšo konvergenco kot primerjalni algoritmi v zahtevnih, dinamičnih mestnih scenarijih.

1 Introduction

Urbanization has led to high population and building density, frequent emergencies, and serious threats to people's lives and property safety [1]. According to statistics, the average annual growth rate of global urban emergencies from 2020 to 2023 is 12.7%, with casualties caused by delayed emergency response accounting for over 34% [2]. Faced with urban emergencies, the timeliness of emergency rescue determines the effectiveness of disaster reduction, and rescue path planning is the core link of rescue timeliness [3]. However, modern urban transportation networks are highly complex and dynamic, and traditional static path planning methods are difficult to cope with the complex and changing road conditions during disasters, often leading to delays and even accidental entry of rescue vehicles into dangerous areas. The collaborative demand for multiple rescue points and resource scheduling in large-scale events further increases the planning difficulty [4]. Particle Swarm Optimization (PSO) can achieve real-time obstacle avoidance path generation in complex terrain, while Genetic Algorithm (GA) can reduce the spatiotemporal cost of resource coordination [5]. The research innovatively proposes a method based on improved GA and Improved PSO (IGA-IPSO), which is an urban emergency rescue path planning technology. Its main goals are: (1) Optimize the PSO algorithm and improve the global convergence of path search and avoid local optimality through Tent chaotic mapping, adaptive inertia weight and traction acceleration terms; (2) Improve the GA and use elite retention, roulette strategy and Metropolis criterion to enhance the local search capability and robustness of task allocation; (3) Combine path planning with task allocation, and achieve multi-objective collaborative optimization through fuzzy evaluation and game theory weight allocation; (4) Verify the timeliness, safety and economy of the proposed method in a dynamic urban environment. The study first introduces Tent chaotic initialization particles, and the adaptive weights and exponential factors work together to enhance the global search of PSO. The combination strategy of elite retention and roulette wheel improves the selection efficiency and convergence speed of GA, and combines Metropolis criterion to optimize crossover and mutation operations, enhancing the local search ability of GA. Finally, a fuzzy rating system for urgency is constructed, and weights are dynamically allocated based on game theory. The GA and PSO are combined to achieve consistency compromise of conflicting objectives and improve decision-making efficiency. The research mainly focuses on the emergency rescue path planning of Unmanned Aerial Vehicle (UAV). Different from ground vehicle path planning, UAV path planning needs to consider key factors such as three-dimensional flight space, battery life, no-fly zone, minimum turning radius, multi-aircraft collision avoidance and communication constraints. These constraints will be embedded in the objective function as hard constraints or penalties to ensure the feasibility and safety of the path. The research focuses on UAV path planning. Future work will be extended to ground rescue vehicles, taking into account constraints such as turning penalties and traffic flow dynamics. It is expected that the method can offer theoretical support for emergency rescue path planning in different scenarios.

2.1 Related works

Urban emergencies occur frequently. The research on emergency rescue path planning for urban emergencies is of great significance. Jose C et al. built a dynamic path planning scheme to address the increased traffic accidents. The research method had good performance [6]. Zhao X et al. designed an optimization model in view of the insufficient efficiency in global COVID-19 epidemic. The results indicated that the method was helpful for emergency resource scheduling [7]. Liu J et al. established a site selection model for emergency medical facilities in mega cities during sudden public health emergencies. The method had higher performance and computational speed [8]. Xia H et al. built a new large-scale emergency medical material scheduling method to deal with the low turnover efficiency of medical supplies. The method addressed the demand differences at each disaster point that affected fair distribution [9].

Many scholars have explored the application of GA and PSO. El-Shafiey M G et al. proposed a mixed GA and PSO algorithm based on random forest for early prediction of heart disease, demonstrating higher prediction accuracy [10]. Shen L et al. built a combinatorial optimization model for scheduling Seru production with dynamic resource allocation. During the research process, a hybrid GA-PSO with nonlinear inertia weights was built. This method demonstrated satisfactory effectiveness [11]. Bousnina K et al. designed an integrated PSO and GA to predict and optimize the roughness of interactive features, as it was difficult to find appropriate cutting parameters and machining processes for simple machining characteristics of CNC machine tools. The method was found to be efficacious [12]. Sun H et al. designed an improved distance vector hop localization algorithm to make the data obtained by each sensor node in wireless sensor networks meaningful. During the research process, the two-dimensional hyperbolic positioning algorithm and improved adaptive GA were combined to estimate the coordinates of unknown nodes. The method demonstrated adequate stability and accuracy [13]. Zhi Y et al. took the GA-PSO to optimize the parameter values of support vector machines for estimating the health status of retired batteries. The research method exhibited estimation accuracy and convergence speed [14]. In summary, the table is shown below.

Table 1: Comparison of related works in emergency response path planning

Algorithm used	Environment	Success rate (%)	Average path length (km)	Re-planning capabilities	Robustness metrics
Optimization-based	Dynamic	85	12.0	Yes	Medium
Deep Q networks	Static	N/A	N/A	No	Low
GA	Static	N/A	N/A	No	Medium
Scheduling algorithm	Dynamic	N/A	N/A	No	Low
Hybrid GA-PSO	Static	N/A	N/A	No	Medium
Hybrid GA-PSO	Static	N/A	N/A	No	Medium
Improved adaptive GA	Static	N/A	N/A	No	Medium
GA-PSO-SVR	Static	N/A	N/A	No	Medium

Note: N/A indicates that this dimension is not suitable for non-path planning research. The robustness index is based on the ability to handle interference such as dynamic obstacles and noise in the literature.

As shown in Table 1, existing research has a good effect on constructing emergency rescue path planning for urban emergencies, but there are still research gaps and deficiencies in collaborative operation mechanisms and strategies to deal with complex scenarios. GA and PSO can dynamically adjust the computational complexity based on the characteristics of input samples or environmental changes, achieving a balance between model accuracy and efficiency. Therefore, the study proposes an urban emergency rescue path planning based on the IGA-IPSO, hoping that it can meet design requirements, and improve the safety, stability, and operational efficiency.

3 Emergency rescue path planning method for urban emergencies

3.1 Emergency rescue path planning based on IPSO

In recent years, various unexpected events such as fires and car accidents have occurred frequently in various cities in China. The suddenness and uncertainty of these events have caused rescue personnel to be unable to rush to the scene for timely rescue, resulting in significant losses [15]. With the rapid development of the drone industry, large payload, long endurance, and highly intelligent drones have emerged, and many countries have incorporated drones into their emergency rescue systems. Table 2 presents the UAV widely used in emergency rescue missions in China.

Table 2: Drone models and their executable tasks applied to emergency rescue

Unmanned aerial vehicle model	Unmanned aerial vehicle model
Dji Matrice 350 RTK	Thermal imaging life detection, etc.
Zongheng Co., LTD. CW-40 Vertical lift Fixed wing	Emergency communication relay coverage, etc.
Large scale disaster investigation, emergency communication relay coverage, etc	Toxic gas monitoring, etc.
Daotong Dragon Fish series medical drones	Cold chain transportation of blood/vaccines, etc.
The Hava Tomahawk series of special unmanned aerial vehicles	Ruin penetration detection, etc.
Xag Emergency Rescue Drone V50	Fire investigation in high-rise buildings, etc.

As shown in Table 2, various types of drones are widely used in different tasks of emergency rescue based on their performance. During the emergency rescue process of UAVs, PSO is used for path planning. The UAV path planning problem is defined as: finding the optimal path from the starting point to the end point in three-dimensional space, and minimizing the path length and risk-weighted sum. Key constraints include: (1) Battery life: the total length of the path is not greater than the maximum endurance distance; (2) No-fly zone: the path point must not be located within the no-fly zone set. Otherwise, a penalty will be imposed; (3) Minimum turning radius: the path curvature is not greater than the maximum curvature, where the maximum curvature depends on the drone model; (4) Collision avoidance: the distance between multiple drone paths is not less than the safe distance; (4) Communication range: the drone location must meet the communication link budget. The above constraints are implemented through hard bound checks or penalty terms in the fitness function. The equation for calculating curvature using the discrete point three-point method is $\kappa = \frac{2 \cdot |(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)|}{\sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)^3}}$, and the unit is m^{-1}. The smaller the curvature, the smoother the path, the easier it is for the drone to track. It can effectively reduce energy consumption and the risk of loss of control. To enhance the security of path planning, the study introduces Control Barrier Functions (CBF) as a real-time safety filter to ensure that the path always meets the minimum safety distance constraint in a dynamic environment. In addition, the Optimal Reciprocal Collision Avoidance (ORCA) algorithm is used to handle mutual collision avoidance between multiple drones to avoid deadlock and shock. To deal with disturbances in actual flight (such as wind field and sensor noise), the wind field disturbance term and sensor noise model are introduced into the fitness function. To simulate the urban built environment, the study uses voxel maps to represent the probability of obstacle occupancy, and introduces line-of-sight inspection and minimum flight height constraints (such as ≥30m over built-up areas). The no-fly zone acts as a hard boundary that imposes a penalty term in the fitness function. The research is passed, in which the trigonometric functions of different frequencies and directions are superimposed, as shown in equation (1).

$$h(x,y) = \sin(y+a) + b\sin(x) + c\cos(d\sqrt{x^2+y^2}) \\ + e\cos(y) + f\sin(f\sqrt{x^2+y^2}) + g\cos(y) \tag{1}$$

In equation (1), (x,y) signifies the plane coordinates of a point on the map. $h(x,y)$ signifies the terrain elevation value corresponding to the point. a,b,c,d,e,f represent the influence on density, waveform, etc. in various directions. When drones are used for emergency rescue in urban emergencies, they are subject to constraints such as endurance. The PSO algorithm can drive particles to move towards the optimal solution based on changes in drone speed during drone rescue path planning, thereby updating candidate paths [16]. The updated particle velocity and position are shown in equation (2).

$$\begin{cases} \mathrm{v}_{id}^{t+1} = w\mathrm{v}_{id}^{t} + c_1 r_1 (pbest_{id}^{t} - X_{id}^{t}) + c_2 r_2 (\mathrm{g}best_{d}^{t} - X_{id}^{t}) \\ X_{id}^{t+1} = X_{id}^{t} + v_{id}^{t+1} \end{cases} \tag{2}$$

In equation (2), v represents the velocity of the drone particle iteration. w signifies the inertia weight.

c_1 and c_2 signify individual learning factors and social learning factors, and $r_1, r_2 \in [0,1]$. $pbest$ signifies the optimal position of individual particles. $gbest$ signifies the global optimal position of the population. X represents the current location. Figure 1 displays the process of emergency rescue path planning for urban emergencies based on PSO.

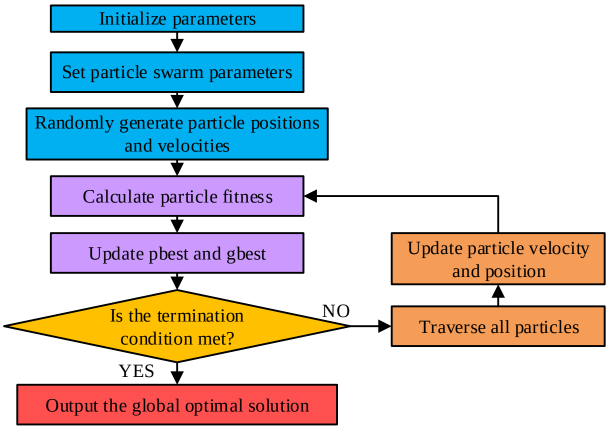

Initialize parameters
Set particle swarm parameters
Randomly generate particle positions and velocities
Calculate particle fitness
Update pbest and gbest
Is the termination condition met?
NO
YES
Traverse all particles
Update particle velocity and position
Output the global optimal solution

Figure 1: Path planning process based on PSO

In Figure 1, the urban emergency rescue based on PSO algorithm first initializes parameters and sets particle swarm parameters, and randomly generates initial population representative candidate paths. Next, the fitness of each path is calculated, and pbest and gbest are updated to dynamically update particle velocity and position to explore better paths. Finally, the termination condition is determined. If it is not satisfied, all particles are traversed and iteratively updated until it is satisfied and the global optimal solution is output, which is the best rescue path. Fitness is the weighted sum of minimizing path length and risk, as shown in equation (3).

$$F_{path} = \alpha \cdot Distance + \beta \cdot Risk \quad (3)$$

In equation (3), F_{path} represents the fitness function. α and β represent weight coefficients. $Distance$ represents the total length of the path. $Risk$ is calculated based on terrain elevation and obstacle density. Although traditional PSO algorithm has strong robustness and other advantages, it cannot adjust the initial position of particles and has local optima. The traditional PSO algorithm is optimized. The study introduces an improved Tent chaotic mapping to optimize the initial position of particles in the PSO algorithm. The study introduces an improved Tent chaotic mapping to solve the low randomness in traditional PSO particle initialization, enhances global search diversity, and avoids falling into local optimality in complex scenes. Its expression is shown in equation (4).

$$x_{i+1} = \begin{cases} 2x_i + r \times \dfrac{1}{N}, 0 \le x_i < \dfrac{1}{2} \\ 2(1-x_i) + r \times \dfrac{1}{N}, \dfrac{1}{2} < x_i \le 1 \end{cases} \quad (4)$$

In equation (4), x_{i+1} signifies the value of the $i+1$-th iteration in the chaotic sequence. r signifies a random number within the $[0,1]$ interval. N represents the population size of the PSO algorithm. The position expression of the optimized particle mapping to the solution space is shown in equation (5).

$$p_i = a + (b-a) \cdot x_i \, (5)$$

In equation (5), x_i signifies the value generated by the improved Tent chaotic mapping. a and b signify the lower and upper bounds of the solution space. p_i signifies the actual position of the mapped particle in the solution space. Aiming at the slow convergence speed of the algorithm in dynamic environments, the global and local searches are balanced by dynamically adjusting weights to improve the planning success rate when the obstacle density changes. The specific expression is shown in equation (6).

$$w = \begin{cases} w_{\max}, F > F_{avg} \\ w_{\min} + \dfrac{(w_{\max} - w_{\min}) \times (F - F_{\min})}{(F_{avg} - F_{\min})}, F \le F_{avg} \end{cases} \quad (6)$$

In equation (6), $w_{\min}, w_{\max}$ represent the preset weight boundary value. F signifies the fitness function value. F_{avg} signifies the average fitness value of all particles in the population. To optimize particle learning behavior and solve the low convergence accuracy of the traditional PSO in multi-objective optimization, the adjusted expression is shown in equation (7).

$$\begin{cases} C_1 = \exp(c_{1\min} + \dfrac{c_{1\max} - c_{1\min}}{t_{\max}} \times t) \\ C_2 = \exp(c_{2\min} + \dfrac{c_{2\max} - c_{2\min}}{t_{\max}} \times t) \end{cases} \quad (7)$$

In equation (7), $c_{1\min}, c_{1\max}$ and $c_{2\min}, c_{2\max}$ represent the boundaries of preset individual learning factors and social learning factors, respectively. $t_{\max}$ represents the maximum iteration. The boundary values of inertia weight and learning factor are shown in Table 3.

Table 3: Variable definition

Variable	Description	Boundary value
$w_{\min}$	Minimum inertia weight	0.4
$w_{\max}$	Maximum inertia weight	0.9
$c_{1\min}$	Lower limit of individual learning factor	0.5
$c_{1\max}$	Upper limit of individual learning	2.0

	factor	
$c_{2\min}$	Lower limit of social learning factor	0.8
$c_{2\max}$	Upper limit of social learning factors	3.0
$t_{\max}$	Maximum number of iterations	100

As shown in Table 3, all the above values are based on experimental calibration. Finally, to force particles to jump out of the local optimum and directly deal with the planning failure under high-density obstacles, the concept of traction acceleration is introduced to optimize the speed update equation of the PSO algorithm. The expression is shown in equation (8).

$$\mathrm{v}_{id}^{t+1} = wv_{id}^{t} + c_1 rand(0,1)(pbest_{id}^{t} - X_{id}^{t}) + c_2 rand(0,1)(gbest_{id}^{t} - X_{id}^{t}) + a_{id}^{t+1}$$
(8)

In equation (8), v_{id}^{t+1} represents the new velocity of particle i in the d -th dimension at the $t+1$ -th iteration. wv_{id}^{t} signifies the inertia term. $rand(0,1)$ signifies a random number of $[0,1]$. a_{id}^{t+1} is the traction acceleration, and its specific form is $a_{id}^{t+1} = \gamma \cdot (gbest_{d}^{t} - X_{d}^{t})$, where γ is the traction coefficient (default value 0.1). This item is activated when the particle fitness has not improved for many consecutive times to force the particle to jump out of the local optimum. Stability is ensured by limiting the γ range. The traction acceleration draws on the speed clamping mechanism in the PSO to avoid divergence by limiting the maximum speed of particles. In addition, its form is similar to the shrinkage factor method, balancing exploration and exploitation by adjusting the value of γ. Stability analysis shows that when $\gamma \in [0, 0.2]$, the particle trajectory converges and there is no oscillation. In summary, the emergency rescue path planning process for urban emergencies based on the IPSO is displayed in Figure 2.

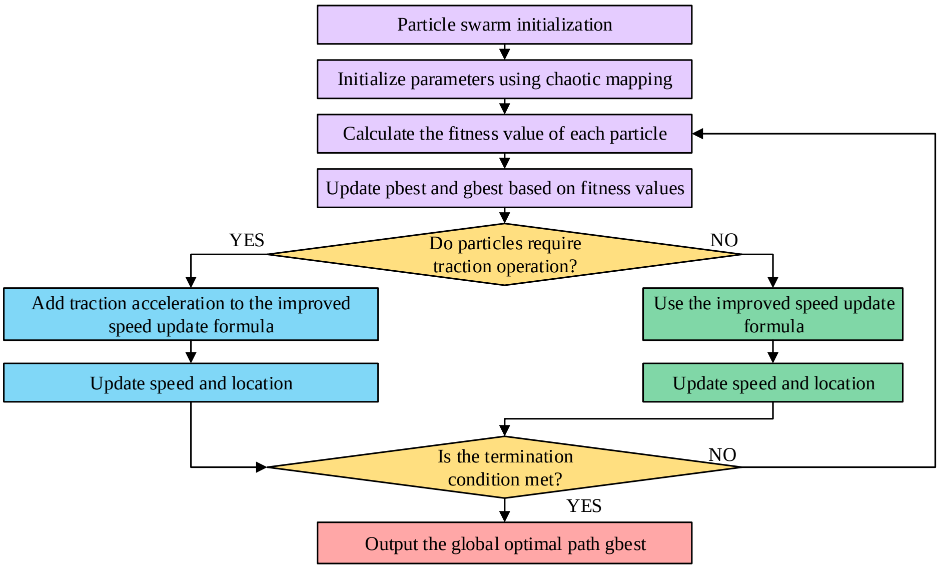

Particle swarm initialization
Initialize parameters using chaotic mapping
Calculate the fitness value of each particle
Update pbest and gbest based on fitness values
YES
Do particles require traction operation?
NO
Add traction acceleration to the improved speed update formula
Use the improved speed update formula
Update speed and location
Update speed and location
Is the termination condition met?
NO
YES
Output the global optimal path gbest

Figure 2: Emergency rescue path planning process based on improved PSO

In Figure 2, the process first uses chaotic mapping to initialize particle swarm parameters and improve search diversity. Next, the fitness function evaluates path distance and risk, and dynamically updates individual and global optimal solutions. Subsequently, a traction operation decision mechanism is introduced, which adds traction acceleration to the velocity update equation when particles fall into local optima, forcing them to move towards the global optimum direction. Finally, after iterative optimization to meet the termination condition, the global optimal rescue path is output. The optimal rescue path primarily involves: finding a collision-free route from the starting point to the destination, ensuring the path adheres to the drone's dynamic constraints such as minimum turning radius and maximum climb rate, and completing the planning within the maximum computation time. The trigger condition of the re-planning mechanism is that the distance between the new obstacle and the path is less than the safe distance or the path is blocked. The planning range is through rolling time-domain planning, the hot start uses the pbest/gbest of the previous round of planning as the initial particle position, and the historical information reuse accelerates convergence by retaining elite particles.

3.2 Emergency rescue path planning based on IGA-IPSO

The path planning for urban emergencies based on IPSO can quickly obtain the shortest and smoothest path for drones to the location of urban emergencies [17]. However, emergency rescue in urban areas not only requires planning the optimal rescue path, but also reasonable task allocation planning for emergency rescue material distribution and other tasks. Multi-drone task allocation: Under the premise of optimal path, to match the best task plan for multiple drones, drone payload and task urgency are considered [18]. The task allocation problem aims to allocate appropriate tasks to multiple UAVs, with the goal of minimizing the total cost. Constraints include: (1) Load constraint: the total mission load of the UAV is not greater than the maximum mission load; (2) Urgency constraint: the urgency of the task affects the allocation priority; (3) Time window: the task must be completed before the end of the task. These constraints are enforced through the GA encoding and decoding process, or incorporated into the fitness function as penalty terms. However, multi-UAV task allocation technology needs to consider factors such as the load capacity of the UAV and the urgency of the mission target. Multi-drone task allocation technology needs to consider drone payload capacity and the urgency of task objectives. In the multi-UAV task allocation, in addition to load and time window constraints, the research introduces a distributed safety collaboration protocol to ensure that the minimum safe distance between UAVs is always met during task execution. This protocol implements dynamic obstacle avoidance based on the ORCA and combines the communication delay model (OPC UA + MQTT protocol) for real-time collaborative verification. In addition, the task allocation plan needs to pass formal verification tools (such as UPPAAL) for deadlock and conflict detection before execution to ensure its feasibility under high-density dynamic obstacles. The GA can simulate the natural selection and genetic mechanisms of biological evolution. Therefore, the study introduces the GA to assign distribution tasks to each drone in the distribution center, efficiently executing emergency rescue tasks for urban emergencies [19]. The GA framework for multi-drone task allocation is shown in Figure 3.

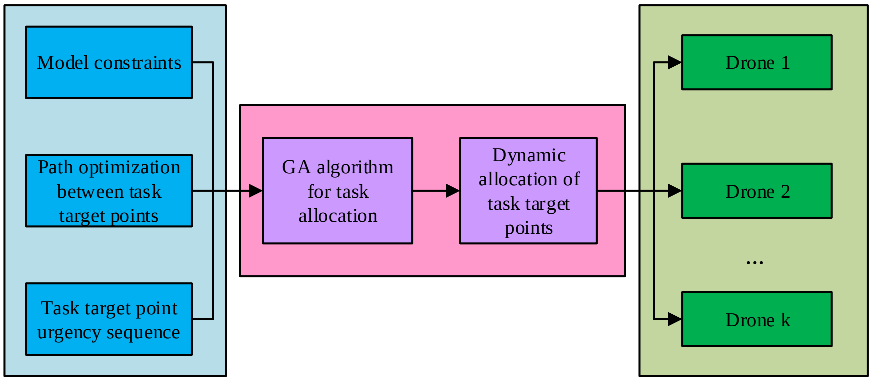

Model constraints
Path optimization between task target points
Task target point urgency sequence
GA algorithm for task allocation
Dynamic allocation of task target points
Drone 1
Drone 2
...
Drone k

Figure 3: Task allocation structure

In Figure 3, when using the GA to allocate UAV emergency rescue tasks in urban emergencies, it is necessary to first consider model constraints and target point path optimization, and construct emergency sequences to prioritize high-risk points. Then, the GA is used to dynamically allocate tasks and respond to sudden changes, ultimately achieving collaborative execution among multiple drones. Due to the numerous parameters involved in task allocation for multiple drones, GA is difficult to control accurately and has insufficient local optimization performance. Therefore, to optimize the robustness and global search capability of GA during the search process, the GA is optimized. The research introduces a combination strategy of elite retention and roulette to solve the lost excellent solutions in traditional GA during task allocation, and improve the convergence speed and resource allocation efficiency. The expression is shown in equation (9).

$$N_{el} = \lfloor g \times PopulationSize \rfloor \ (9)$$

In equation (9), N_{el} represents the number of elite individuals. g is the proportionality coefficient. $PopulationSize$ represents the population size. $\lfloor \cdot \rfloor$ is a down rounding function. After the elite retention strategy, the roulette wheel selects the remaining individuals for crossover and mutation operations, and the selection probability of the roulette wheel is shown in equation (10).

$$P(x_i) = \frac{fit(x_i)}{F} \ (10)$$

In equation (10), $P(\cdot)$ represents the probability of being selected to enter the mating pool. $fit(\cdot)$ signifies the fitness value. F signifies the total fitness of the population. Next, to obtain the cumulative probability distribution interval, the corresponding individuals are selected and the cumulative probability of all individuals being selected by roulette wheel is calculated. The calculation is shown in equation (11).

$$C(x_i) = \sum_{i=1}^{i} P(x_i) \ (11)$$

In equation (11), $C(\cdot)$ signifies the sum of the selection probabilities of all individuals. $P(\cdot)$ signifies the individual's selection probability. To enhance the local search capability of the GA, the Metropolis criterion is introduced to enhance the local search capability and avoid premature convergence in multi-UAV task allocation. The expression is shown in equation (12).

$$P\left(\Delta E,T\right)=\begin{cases}1,\Delta E<0\\ \exp(-\dfrac{\Delta E}{k_bT}),\Delta E\geq 0\end{cases}(12)$$

In equation (12), P signifies the probability of the algorithm accepting a new solution. ΔE represents the amount of energy change. T represents the current temperature. k_b represents the Boltzmann constant. The GA chromosome uses real number encoding to represent the UAV task allocation sequence. The chromosome length is the number of tasks, and each gene corresponds to a task ID. The crossover operation uses two-point crossover, and the mutation operation is implemented by randomly exchanging genes. Specifically, the crossover probability is 0.8 and the mutation probability is 0.1, which are dynamically adjusted based on the Metropolis criterion. To solve the poor adaptability of fixed search times in complex terrain, the search depth is dynamically adjusted to improve path smoothness and task completion rate. The study introduces an adaptive variable neighborhood search time scheme to deeply optimize the GA. The specific expression is shown in equation (13).

$$IterGen=\rho_1+\left\lfloor \rho_2\times\frac{Gen}{GenMax}\right\rfloor(13)$$

In equation (13), $IterGen$ signifies the total times variable neighborhood searches are performed on elite individuals in the current iteration round. ρ_1 represents the minimum number of guaranteed times for variable neighborhood search. ρ_2 represents the upper limit span of adaptive search times. Gen represents the current iteration count. The GA chromosome uses extended real number encoding and includes fields such as task ID, payload quality, and time window. The reconnaissance task constraint is that the load is no more than 2kg, and the service time is no more than 5min. The material delivery constraint is that the load is no more than 10kg, and the delivery time window is [5,30] min. The types of multi-drone task allocation based on GA are shown in Figure 4.

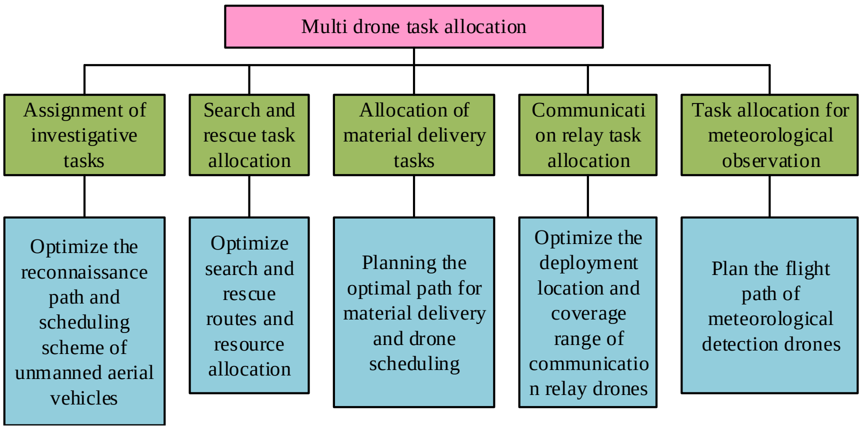

Multi drone task allocation
Assignment of investigative tasks
Search and rescue task allocation
Allocation of material delivery tasks
Communication relay task allocation
Task allocation for meteorological observation
Optimize the reconnaissance path and scheduling scheme of unmanned aerial vehicles
Optimize search and rescue routes and resource allocation
Planning the optimal path for material delivery and drone scheduling
Optimize the deployment location and coverage range of communication relay drones
Plan the flight path of meteorological detection drones

Figure 4: Classification types of urban emergency rescue drone tasks based on GA

As shown in Figure 4, the multi-drone task allocation for urban emergency rescue mainly targets five types of tasks: reconnaissance, search and rescue, material delivery, communication relay, and meteorological detection. GA is used to achieve path planning, resource allocation, scheduling scheme optimization, or deployment location and coverage adjustment. GA is used to improve the scientificity and emergency rescue efficiency of multi-drone task allocation. To determine the urgency sequence of urban emergency rescue needs, the study first introduces a triangular membership function to establish a fuzzy evaluation system, transforming multi-objective conflicts into single-objective optimization, and solving the uneven weight distribution in collaborative decision-making. The expression is shown in equation (14).

$$\mu_{\tilde{a}}(x)=\begin{cases}0,x<a_1 or x>a_3\\ \dfrac{x-a_1}{a_2-a_1},a_1\leq x\leq a_2\\ \dfrac{a_3-x}{a_3-a_2},a_2\leq x\leq a_3\end{cases}(14)$$

In equation (14), $\mu_{\tilde{a}}(x)$ represents the degree to which the evaluation object x belongs to the fuzzy concept a. x signifies the actual value of the object to be evaluated. a_1,a_2,a_3 are vertex parameters of the triangle. To further enhance the objectivity of urgency rating, the study clarifies the specific variables and scales of the fuzzy evaluation system.

The set evaluation variables include the number of casualties (unit: people), the impact scope of the event (unit: km^2), and the expected response time (unit: minutes). Each variable is independently rated by experts using a 5-level Likert scale (1-5 points). The parameters of the triangular membership function are calibrated based on historical data: a, b, and c correspond to the minimum, median, and maximum values of each variable score, respectively. To verify the consistency of ratings, the Kappa coefficient is used to evaluate the consistency of ratings among experts, and the ROC curve is used to verify the effectiveness of the ratings in determining actual rescue priority. To determine the urgency weight of emergency rescue in urban emergencies, this study is based on game theory to achieve consistency or compromise between the weights obtained by different weighting methods. The combination weight is presented in equation (15).

$$w = \sum_{k=1}^{n} \alpha_k^* u_k^T \quad (15)$$

In equation (15), w represents the final demand urgency combination weight vector. n signifies the quantity of methods. α_k^* signifies the combination coefficient corresponding to the k-th original weight calculation method. u_k represents the weight vector obtained by the k-th original weight calculation method. This study transforms the path length cost and urgency matching cost of two conflicting objectives into a flexible single objective problem through normalization and weighted summation, providing decision-makers with multiple delivery strategies. Equation (16) presents the objective function.

$$\min F = \theta_1 \frac{f_1 - \min f_1}{\max f_1 - \min f_1} + \theta_2 \frac{\max f_2 - f_2}{\max f_2 - \min f_2} \quad (16)$$

In equation (16), $\min F$ represents the total cost that needs to be minimized. θ_1 and θ_2 represent the importance of balancing the path length cost f_1 with the urgency matching cost f_2, satisfying f_2. Weights θ_1 and θ_2 are dynamically assigned through game theory. The default value θ_1 is 0.6 (path length priority) and θ_2 is 0.4 (urgency priority). Normalization uses Min-Max scaling to ensure that each target value is in the range [0,1], where $\min f_1$ and $\max f_1$ are estimated based on historical data. To eliminate the dimensional difference between the path length cost and the urgency matching cost, the Min-Max normalization is used to scale the two to the [0,1] interval, and its specific expression is shown in equation (17).

$$\begin{cases} C'_{\text{path}} = \dfrac{C_{\text{path}} - C_{\text{path,min}}}{C_{\text{path,max}} - C_{\text{path,min}}} \\ C'_{\text{urgency}} = \dfrac{C_{\text{urgency}} - C_{\text{urgency,min}}}{C_{\text{urgency,max}} - C_{\text{urgency,min}}} \end{cases} \quad (17)$$

In equation (17), C'_{path} and C'_{urgency} are set based on historical extreme scenarios (such as the shortest/longest path, the lowest/highest urgency), and their final objective function is $J = \theta_1 C'_{\text{path}} + \theta_2 C'_{\text{urgency}}$. In summary, the technical roadmap based on the IGA-IPSO is shown in Figure 5.

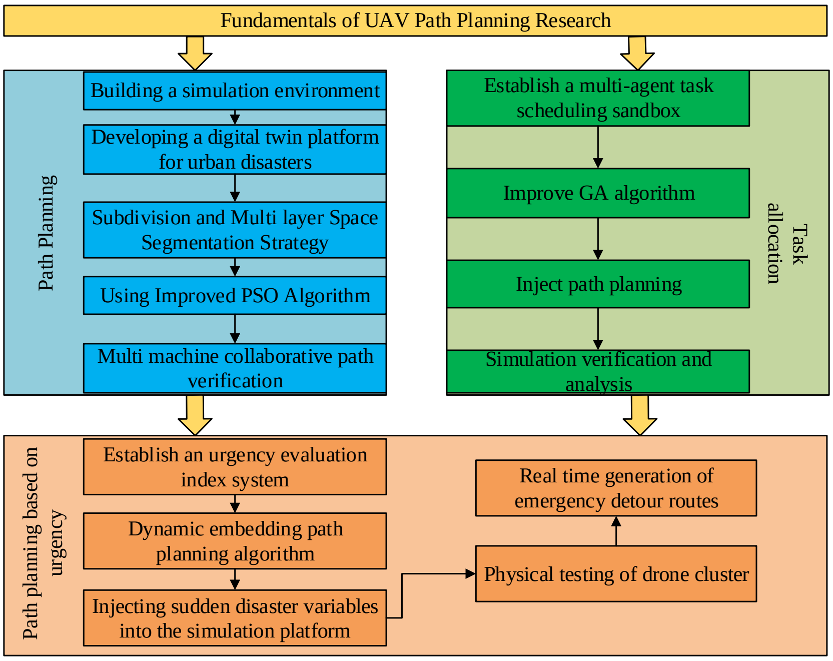

Fundamentals of UAV Path Planning Research
Path Planning
Building a simulation environment
Developing a digital twin platform for urban disasters
Subdivision and Multi layer Space Segmentation Strategy
Using Improved PSO Algorithm
Multi machine collaborative path verification
Establish a multi-agent task scheduling sandbox
Improve GA algorithm
Inject path planning
Simulation verification and analysis
Task allocation
Path planning based on urgency
Establish an urgency evaluation index system
Dynamic embedding path planning algorithm
Injecting sudden disaster variables into the simulation platform
Real time generation of emergency detour routes
Physical testing of drone cluster

Figure 5: Route planning technology based on improved GA and PSO

In Figure 5, the process based on IGA-IPSO optimizes conventional path planning and multi-machine collaborative verification using hierarchical spatial strategy and IPSO algorithm. In addition, focusing on time critical scenarios, an emergency assessment system is established to coordinate multi-agent task scheduling through IGA, dynamically embed path planning to respond to sudden variables, and achieve "virtual-physical" dual loop verification with drone clusters. The collaboration of two paths enhances the timeliness and adaptability of path planning, providing effective support for emergency rescue in complex urban scenarios. The key parameters involved in the study (including the initial temperature and Boltzmann constant in the Metropolis criterion, as well as the basic number and upper limit span in the adaptive variable neighborhood search) are determined through grid search and pre-experimental calibration. The Metropolis criterion parameters: The initial temperature of 100 is a typical setting based on simulated annealing algorithm combined with exponential cooling strategy. The Boltzmann constant is set to 1 as a standard normalization factor to ensure that the acceptance probability is within a reasonable range. Adaptive variable neighborhood search parameters: Based on sensitivity analysis under different scene complexities (50~200 nodes), the base number is determined to be 5 and the upper limit span is 100 to ensure a balance between convergence speed and search depth. All hyperparameters are tuned through grid search, and the search range is determined based on preliminary experiments. Results are averaged based on 30 independent runs. The hyperparameter settings are as follows:

Table 4: Hyperparameter setting and tuning protocol

Algorithm component	Parameter	Value/Range	Dispatching mode
PSO	Population size	50	Fixed
	Inertia weight	[0.4,0.9]	Adaptive segmentation
	Learning factor c_1, c_2	[0.5,2.0], [0.8,3.0]	Adaptive index
GA	Population size	100	Fixed
	Cross rate	0.8	The Metropolis criteria are dynamically adjusted
	Mutation rate	0.1	The Metropolis criteria are dynamically adjusted
	Selection strategy	Elite reserve + roulette	The proportionality coefficient k=0.1
Variable Neighborhood Search (VNS)	Minimum number of times	5	Adaptive
	Upper limit span	10	Adaptive
Metropolis Criteria	Initial temperature	100	Exponential cooling
Fuzzy evaluation system	Vertex of a triangle (a,b,c)	(0, 0.5, 1)	Calibration based on historical data
Combination coefficients in game theory	λ_i	[0.4,0.6]	Normalized weighting

The specific steps of the IGA-IPSO algorithm are shown in Table 5.

Table 5: Pseudo-code of IGA-IPSO algorithm

```
Algorithm 1: IGA-IPSO Algorithm
Input: Particle swarm size, maximum number of iterations, fitness function
Output: Optimal path and task allocation
Step 1: Initialization: Generate particle positions using the Tent chaotic map
Step 2: For t = 1 to $t_{max}$
Step 3: Calculate the fitness
Step 4: Update pbest and gbest
Step 5: If the particle falls into a local optimum, add the traction acceleration
Step 6: Use GA for task allocation
Step 7: If convergence occurs (fitness change <0.01), exit
Step 8: End For
Step 9: Return the optimal solution
```

As shown in Table 4, the IGA-IPSO algorithm has been described in detail. The research adopts a two-layer loop optimization framework: The outer layer is the IGA task allocation loop, and the inner layer is the IPSO path planning loop. The specific scheduling is as follows: Outer loop (IGA): Each iteration generates a set of task allocation plans. Inner loop (IPSO): Based on the current task plan, calculates the cost of each path. Feedback mechanism: The path cost is fed back to the IGA for updating the fitness function. Stopping criterion: Terminates when the fitness change rate of 10 consecutive rounds of iterations is less than 1% or reaches the maximum number of iterations. This coupling mechanism ensures dynamic coordination between task allocation and path planning.

4 Verification of emergency rescue path planning for urban emergencies based on IGA-IPSO

4.1 Performance testing of emergency rescue path planning based on IGA-IPSO

To verify the performance of the designed method, a simulation model is constructed, and its experimental environment and specific configuration are shown in Table 6.

Table 6: Test environment and specific configuration

Testing environment	Specific configuration
Simulation environment	Python + NumPy
Computing equipment	Intel Core i7-8550U CPU @1.80GHz, 16GB RAM
Environmental monitoring	Distributed gas sensor, and thermal imaging camera
Dynamic obstacle detection	The unmanned aerial vehicle is equipped with a dual-mode visible light/infrared camera
Environmental modeling tool	Three-dimensional topological map
Heterogeneous system integration	The OPC UA + MQTT hybrid protocol is adopted

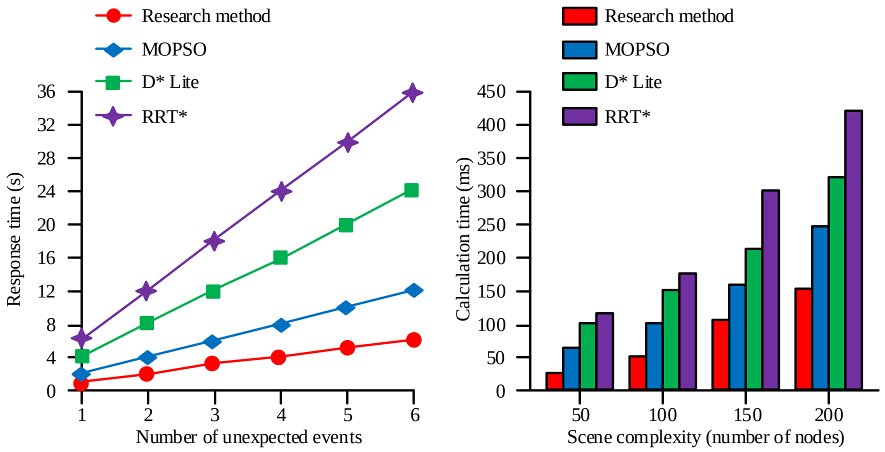
In Table 6, the research uses the experimental environment and specific configuration in the table for performance testing. In this experiment, the performance indicators are clearly defined: The planning success rate is the proportion of the algorithm that successfully generates paths that meet all constraints (such as battery life, no-fly zones, collision avoidance, etc.) in 30 independent runs. Convergence is when the fitness change rate for 10 consecutive iterations is less than 1%, and the algorithm is deemed to have converged. All experimental results are based on 30 independent runs, initialized with different random seeds. Some results are expressed as mean ± standard deviation, and statistical significance is verified by paired t-test (α=0.05). To comprehensively evaluate the robustness of the algorithm in uncertain and dynamic environments, this study injects the following disturbance factors into the simulation: Wind field disturbance: simulated gust model, wind speed 5-10 m/s, random direction; Positioning noise: injected Gaussian noise (σ=0.5m) in position perception; Dynamic obstacles: used random walk model, speed 0-5 m/s, acceleration ±1m/s^2; Communication delay: simulated OPC UA + Communication delay under MQTT protocol (average 100ms, standard deviation 20ms); Map error: randomly offset obstacle position (offset obeys N(0,1)m). Evaluation indicators include: Feasibility rate: the proportion of successfully generating feasible paths in 30 runs; Number of safety constraint violations: such as collisions, no-fly zone intrusions, etc.; Mission deadline miss rate: the proportion of task completion time exceeding the preset time window. To ensure the repeatability of the experiment, the study uses the public road network data set (OpenStreetMap, a city center area, area 10km×10km) as the basic map. The map grid resolution is 100m×100m, the elevation data comes from SRTM DEM (spatial resolution 30m), and the no-fly layer (such as high-rise buildings, sensitive areas) is synthesized. The dynamic obstacle generation is based on the random walk model, the density unit is "person/km^2", the initial position of the obstacles is randomly distributed, the movement speed range is 0-5m/s, and the update frequency is 1Hz. Scenario complexity is expressed as the number of nodes (50-200) in the path planning graph, where nodes represent road intersections or task points. Based on map density, number of obstacles and task complexity, the study defined three difficulty levels. The details are shown in Table 7.

Table 7: Definition of difficulty levels

Difficulty level	Map type	Obstacle density (units /km²)	Number of tasks	The number of risk areas
Tier 1	Low-density area	20	1-2	0-1
Tier 2	Low-density area	40	3-4	2-3
Tier 3	High-density area	60	5-6	4-5

In Table 7, 30 independent experiments were run at each difficulty level, and indicators such as planning success rate, response time, and path length were recorded. The scene generation and parameter setting in the test mainly include: (1) Concurrent event generation: Randomly generate 1-6 emergency event points in the map. Event types include fire, car accident, drowning, etc., and the event locations obey uniform distribution; (2) Dynamic obstacle density: Simulate urban dynamic traffic by controlling the number of obstacles per unit area (such as 40/km^2), and the obstacle motion model is random direction + constant speed; (3) Scene complexity: Control the scale of the problem by adjusting the number of nodes in the path planning diagram (50, 100, 150, and 200), where nodes represent accessible path points; (4) Risk map synthesis: Construct a risk weight map based on elevation data, obstacle density and historical event data, and calculate the risk value; (5) Task urgency and time window: Task urgency is generated based on the triangle membership function, and the time window is randomly assigned through a uniform distribution (5-30 minutes). The research method is compared with the

Rapid Exploration of Random Tree Stars (RRT*), D* Lite and the Multi-objective Particle Swarm Optimization Algorithm (MOPSO). The RRT* algorithm is a sampling-based path planning algorithm, suitable for high-dimensional space and dynamic environments. The D* Lite algorithm is an incremental search algorithm, suitable for dynamic obstacle environments. The MOPSO algorithm is a representative of multi-objective optimization algorithms, used to compare the performance of multi-UAV task allocation. All baseline methods use the same cost function (path length and risk-weighted sum) and constraints (drone endurance, obstacle density, etc.) for fair comparison. To ensure the fairness of the experiment, the maximum running time budget is set to 300 seconds per task. If the algorithm does not converge within the budgeted time, it is considered a failure. All algorithms are executed in single-threaded mode to ensure fair comparison. The experimental platform does not have multi-threading or GPU acceleration enabled. The response time of the four methods under different numbers of emergencies and the single planning calculation time under different scene complexities are compared, as presented in Figure 6.

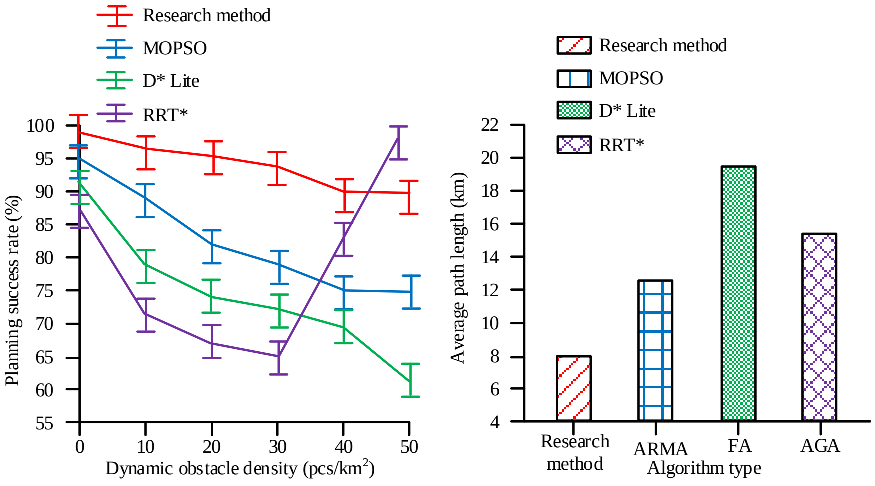


(a) Dynamic programming response time

(b) Single planning calculation time

Figure 6: Dynamic programming response time and single planning calculation time

In Figure 6 (a), the response time of path planning for all four methods gradually increased with the increase of unexpected events, with the research method showing the slowest increasing trend. When the number of emergencies was 1, the response time of the research method was 1 second. When the emergencies increased to 6, the response time increased linearly from 1 to 6 seconds. However, the response time of the three comparison methods was higher than. In Figure 6 (b), the single planning calculation time of the four methods varied under different scene complexities. When the scene complexity was 50 nodes, the computation time of the research method was 25±2ms, and the computation time of the three comparison methods was not obviously different from that of the research method. When the scene complexity was 200 nodes, the single computation time of the research method was 150±12ms, while the computation time of the other three methods was much longer. Overall, compared to comparison methods, the research method has better stability and planning efficiency. The success rates of four methods in planning under different dynamic obstacle densities, as well as the average path lengths planned by the four methods, are compared, as presented in Figure 7.



(a) Algorithm success rate

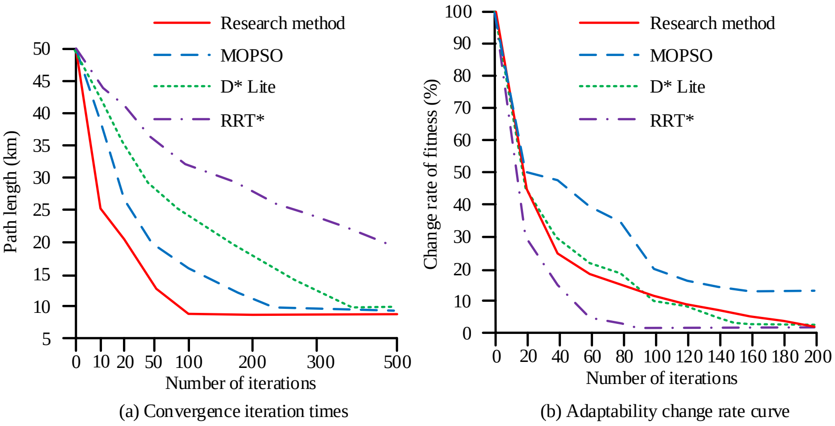
(b) Average path length

Figure 7: Algorithm success rate and average path length

In Figure 7 (a), the planning success rates of the four methods decrease as the density of dynamic obstacles increases. In 30 independent runs, when the dynamic obstacle density was 40/km², the planning success rate of the research method (that is, the proportion of feasible paths successfully generated) reached 90.2%. However, the planning success rates of the other three methods were significantly lower, with the RRT* showing an abnormal increase. In Figure 7 (b), the obvious difference existed in the average path length of the rescue routes planned by the four methods. The average path length of the research method was 8.0±1.2km. Among the three comparison methods, the MOPSO model had an average path length of 12.5±1.5km, while the D* Lite and RRT* had average path lengths of 19.7±2.2km and 15.4±1.6km, respectively, which were significantly larger than that of the research method. Overall, compared to comparison methods, the research method has stronger robustness and optimization capabilities. To verify the impact of constraints on algorithm performance, the study tested the planning success rate under different battery life (from 10km to 50km) and no-fly zone density (from 5/km² to 30/km²). The results show that when the battery life is less than 20km, the planning success rate drops to 70%. When the no-fly zone density exceeds 20/km², the path length increases by an average of 15%. This highlights the importance of constraint handling in complex scenarios. All experimental results are based on 30 independent runs, initialized with different random seeds. Some results are expressed as mean ± standard deviation, and statistical significance is verified by paired t-test (α=0.05). The details of instance generation are: (1) Emergency event types include fires, car accidents, drowning, etc., and the locations are uniformly distributed; (2) Dynamic obstacles are generated based on the random walk model, with a movement speed of 0-5m/s, and an update frequency of 1Hz; (3) The complexity of the scene is controlled by the number of nodes (50-200), and the nodes represent passable path points; (4) The risk map is synthesized based on elevation, obstacle density, and historical event data. Moreover, to verify the robustness of the algorithm under extreme conditions, safety-critical stress test scenarios are designed: (1) Sensing noise: Injecting Gaussian noise (σ=0.5m) into position sensing; (2) Execution noise: Adding random deviation (±5%) to the control instructions; (3) Wind field disturbance: Simulating gust model (wind speed 5-10m/s, random direction). The test results show that under the dynamic obstacle density of 40/km² and wind field disturbance, the planning success rate of the research method is 87.5%, which is only 2.7% lower than that of the non-disturbance scenario. The success rate of the comparison methods (RRT*, D* Lite) dropped by more than 15%. In addition, after introducing CBF and ORCA, the number of collisions dropped from an average of 1.2 times per task to 0.1 times, verifying the effectiveness of the safety filter. The IGA-IPSO has the best planning efficiency, stability, and robustness.

4.2 The practical application effect of emergency rescue path planning based on IGA-IPSO

On the basis of verifying the performance of the IGA-IPSO, the practical application value is verified. The research adopts the urban operation basic dataset and builds a multi-modal simulation platform. The research method is compared with MOPSO, D* Lite, and RRT*. The path length and fitness value change rate of the four methods are compared at different iteration times, and the results are shown in Figure 8.

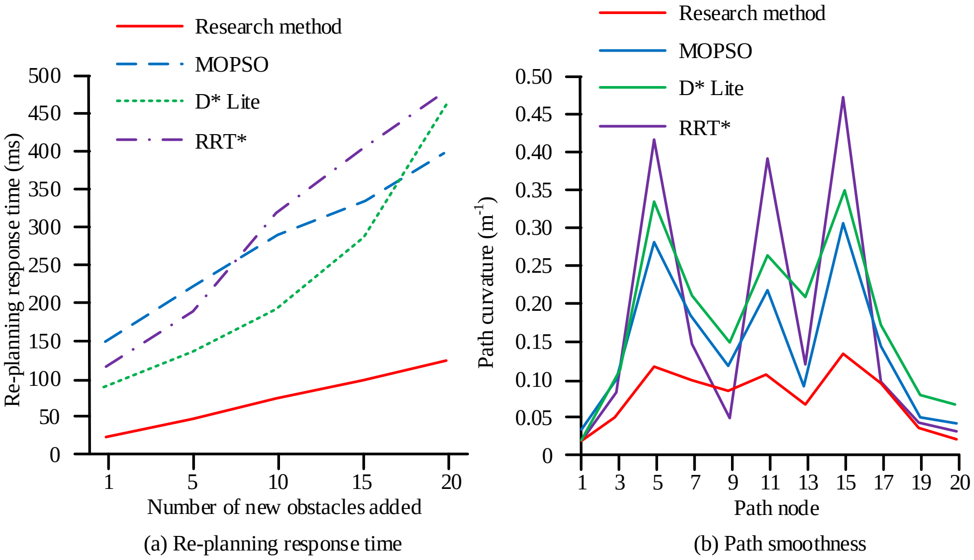


(a) Convergence iteration times

(b) Adaptability change rate curve

Figure 8: Curve of convergence iteration times and fitness change rate

In Figure 8 (a), the path lengths of the four methods gradually converge as the number of iterations increases. According to the convergence criterion (fitness change rate <1%), the research method completed convergence at the 100th iteration, and the path length reached a stable value of 8km. However, the convergence speed of the other three methods was significantly slower. In Figure 8 (b), the fitness change rate of the four methods gradually decreased with the increase of iteration times. The fitness change rate of the research method decreased gently, reaching 25.1% at the 40th iteration and 1.3% at the 200th iteration, with stable convergence. The other three methods showed poor convergence, quadratic convergence, and fast convergence, respectively. Overall, compared to comparison methods, the research method has better efficiency and optimization capabilities. The response time and path curvature changes of four methods for re-planning are compared, as displayed in Figure 9.



(a) Re-planning response time

(b) Path smoothness

Figure 10: Response time and path smoothness for re-planning

In Figure 9 (a), the re-planning time of the four methods increased with the increase of the number of new obstacles added. The research method had a re-planning response time of 25±2ms when the obstacle was added once, and 125±10ms when the obstacle was added 20 times, with an increase of 100±8ms. The re-planning time of the three comparison methods was significantly greater. In Figure 9 (b), the four methods had different curvatures at different path nodes. The research method had a maximum curvature of 0.13±0.02m^{-1} at node 15, and peaked in curvature at nodes 5 and 11. The overall curvature of the other three methods at 20 nodes was greater. Moreover, in an actual environment with wind fields and sensor noise, the average re-planning time of the research method was 135ms (obstacles are added 20 times), which was only an 8% increase compared to the undisturbed scene. The path curvature remained within 0.15m^{-1} under wind field disturbance, indicating that the path smoothness was not significantly affected. In addition, through CBF real-time correction, the drone successfully avoided all dynamic obstacles and no-fly zone intrusions during flight. Overall, compared to comparison methods, the research method has better reliability and stability. All experimental results are based on 30 independent runs, initialized with different random seeds. Some results are expressed as mean ± standard deviation, and statistical significance is verified by paired t-test (α=0.05). Finally, to evaluate the independent contribution of each improvement component, ablation experiments were designed. The experiment was conducted under the same instance (six concurrent events, dynamic obstacle density 40/km^2) and fixed random seeds, and the following components were gradually added: Tent chaotic mapping (C), adaptive segmented inertia weight (W), exponential learning factor (L), traction acceleration (T), elite retention and roulette (E), Metropolis criterion (M), and adaptive variable neighborhood search (V). The results show that traction acceleration (T) and adaptive variable neighborhood search (V) contribute the most to avoiding local optimality, while chaotic initialization (C) and elite strategy (E) significantly improve the convergence speed. The IGA-IPSO has the best efficiency, performance, reliability, and stability. To verify the impact of weights θ_1 and θ_2 on the planning results, a sensitivity experiment was designed: fix $\theta_1 + \theta_2 = 1$, adjust θ_1 from 0.0 to 1.0 with a step size of 0.1, and evaluate the changes in path length, urgency matching error and total cost. The results show that: when $\theta_1 > 0.7$, path length takes priority, but the urgency matching error increases by more than 15%. When $\theta_1 < 0.3$, urgency matching takes priority, but the path length increases by more than 20%. When $\theta_1 \in [0.4, 0.6]$, the total cost changes slowly (fluctuation <5%), verifying the robustness of the default weight. This analysis provides decision-makers with a basis for weight tuning to adapt to the preferences of different rescue scenarios.

5 Discussions

The research proposes an urban emergency rescue path planning method based on the improved GA and the PSO algorithm (IGA-IPSO), and its performance is verified through simulation experiments. (1) Performance comparison and attribution with existing technology: The experimental results that the average path length of this research method is 8.0 km, while the average path length of the comparison method is significantly longer. This improvement is mainly attributed to several optimization components introduced in the IGA-IPSO algorithm: (i) Tent chaotic mapping. (ii) Traction acceleration term. (iii) Adaptive piecewise inertia weights and exponential learning factors. (iv) Elite Retention and Metropolis Code. (2) Trade-offs and limitations: Although the present research method performs well in path length, convergence speed, and robustness, there are still trade-offs and limitations to consider: (i) Computational complexity. (ii) Parameter sensitivity. (iii) Robustness. (3) Scalability analysis. The research tests up to six concurrent events and dynamic obstacle densities up to 40/km² in experiments. From the results: (i) Event number expansion: Response time increases linearly as the number of events increases. (ii) Map density expansion: When the scene complexity is up to 200 nodes, the research method still maintains low calculation time and high success rate. (iii) Multi-UAV collaboration: Through distributed security protocols and formal verification, the research method supports multi-UAV task allocation and path planning. (4) Summary and outlook: In summary, this research method is superior to the existing technology in path length, convergence speed and robustness by integrating improved PSO and GA components. Its main advantage is that it dynamically balances multi-objective optimization and constraint processing, and is suitable for complex urban emergency rescue scenarios. Future work will focus on path planning under extreme weather, large-scale event expansion, and real-time learning mechanisms.

6 Reproducibility and experimental materials

To ensure the repeatability and transparency of this research experiment, complete code, data, configuration and experimental materials are provided and publicly available in the GitHub repository (repository address: https://github.com/[username]/IGA-IPSO-Emergency-Path-Planning). The repository contains the following content:

(i) Deterministic Seeds: All experiments use fixed random seeds (such as seed=42) to ensure consistent results for each run. (ii) Scenario Generators and Maps provide road network data (in .osm or .graphml format) based on OpenStreetMap, and the dynamic obstacle generation script scenario_generator.py. The script supports custom event points, obstacle density and motion models. (iii) Configuration Files (Configuration Files): All algorithm hyperparameters, environment configurations and experimental parameters are stored in the config/ directory in YAML format, including PSO algorithm parameters (such as inertia weights, learning factors, etc.), GA parameters (such as crossover rate, mutation rate, etc.), and simulation environment configuration (such as map range, resolution, obstacle attributes, etc.). (iv) Reproduction Scripts provides a complete Python script for reproducing all charts and tables: The main experimental script performs performance testing and practical application verification, generates all result graphs, and outputs tabular data. (v) README file: README.md in the root directory of the warehouse contains detailed environment configuration instructions, including Python 3.8+ environment setup, dependent library list (such as NumPy, Matplotlib, NetworkX, etc.), data preprocessing steps, and step-by-step operation guide. All materials have been tested and can reproduce all experimental results in this article under the same hardware and software environment.

7 Conclusion

In response to the frequent occurrence of urban emergencies, the low efficiency of traditional emergency rescue path planning, and the insufficient ability to coordinate multiple tasks, an innovative urban emergency rescue path planning method based on the IGA-IPSO was proposed. Compared with traditional Tent chaotic mapping, this study uses parameter adaptation and scene calibration mechanisms to make the initialization process more in line with the dynamic needs of urban emergency rescue, effectively improving the convergence speed and robustness of the algorithm. The research method incorporated the traction acceleration term to optimize the PSO, designed an adaptive variable neighborhood search mechanism to optimize the GA, and constructed a fuzzy evaluation system for urgency. The matching cost between path length and urgency was transformed into a single objective optimization problem. When the scene complexity was 50 nodes, the computation time was 25±2ms. The average path length was 8.0km. In practical application testing, the fitness change rate of the research method decreased gently, from 25.1% in the 40th iteration to 1.3% in the 200th iteration, with stable convergence. The research method showed that the rescue cost for car accidents was around CNY 2,000±12. The rescue cost during a drowning incident was around CNY 500±8. The rescue costs for fire and disease incidents were around CNY 3,500±20 and CNY 2,500±15, respectively. When an obstacle was added once, the response time for re-planning was 25±2ms. When an obstacle was added 20 times, the response time for re-planning was 125±10ms, with an increase of 100±8ms. The research significantly improves the robustness of path planning in safety-critical scenarios by introducing the Control Barrier Function (CBF), ORCA obstacle avoidance algorithm and wind field disturbance model. The stress test shows that the research method still maintains a high success rate and low collision rate under sensing noise, execution deviation and wind field interference, which is suitable for complex dynamic environments in actual urban emergency rescue. Overall, the proposed method enhances the timeliness, robustness, and economy of emergency rescue in complex urban conditions. The research method significantly improves the feasibility and safety of path planning by formalizing key constraints such as battery life, no-fly zones, and collision avoidance. Experiments show that the constraint processing mechanism still maintains a planning success rate of more than 90% under dynamic obstacles and high-density no-fly zones. The current research does not consider the interference of extreme weather on the UAV dynamic model. In future research, Reinforcement Learning (RL) can be introduced for dynamic policy learning and environmental adaptive decision-making to further improve the robustness of the path planning system in uncertain environments. Specifically, RL can be used to: (1) Dynamically adjust the inertia weight and learning factor in IPSO so that it changes adaptively according to real-time wind field, visibility and other meteorological conditions; (2) Optimize the task allocation strategy in IGA and dynamically adjust task priority and resource scheduling according to the evolution of real-time events; (3) Collaborate with the existing IGA-IPSO framework to form a two-layer decision-making mechanism of "offline optimization + online learning": IGA-IPSO is responsible for global path and mission planning, and RL is responsible for local dynamic adjustment and disturbance compensation. By embedding RL into the existing framework and combining it with the meteorological-terrain coupling constraint model, the research method can better adapt to dynamic uncertain scenarios such as extreme weather, and improve the real-time performance and adaptability of the emergency rescue system.

8 Deployment considerations

The main deployments in the study include: (1) Regulatory compliance. BVLOS (beyond visual line-of-sight) approval: Apply for a BVLOS flight permit from the Civil Aviation Administration in actual deployment, and submit a risk assessment report, emergency response plan, and other materials; Dynamic update of no-fly zones: The system should access real-time no-fly zone data released by civil aviation and the military, and respond dynamically in path planning; Airspace hierarchical management: Design an airspace priority usage mechanism based on the urgency of the task to ensure priority for high-urgency tasks. (2) Safety interlocking mechanism. Real-time health monitoring: The drone needs to be equipped with battery voltage, GPS signal, communication link and other status monitoring. Once abnormal, it will trigger automatic return or backup; Human intervention interface: The system should support manual takeover by the command center, with functions such as one-click pause, path re-planning, and mission abort; Fault emergency protocol: Define the emergency handling process for extreme situations such as the drone losing contact, crashing, and being hijacked. (3) System integration and command and dispatch. API interface specifications: Provide RESTful API or MQTT themes to support docking with urban emergency command platforms (such as 110, 120, and 119 systems); Task status synchronization: Push task progress, path deviations, remaining resources and other information to the command center in real time; Logs and auditing: Record all planning decisions, path execution, abnormal events, and support post-event review and responsibility tracing. (4) Multi-modal collaboration and handover protocol. UAV-ground vehicle collaboration: (i) UAVs are responsible for rapid reconnaissance and preliminary material delivery, and ground vehicles are responsible for large-scale material transportation or personnel transfers; (ii) Handover points are usually located in safe areas (such as squares and school playgrounds), and UAVs guide ground vehicles to the target point. Mission handover process: (i) After the drone completes the mission, it sends "mission completion + location information" to the command center; (ii) The command center assigns the nearest ground vehicle to respond; (iii) The drone hovers or lands at the handover point, waiting for confirmation before returning.

Funding

The research is supported by Special Project of Guangxi Public Security Department in2023: Research on Prevention and Control Strategies for the New Business Model of Homestay +"Accommodation in Guangxi Border Areas (2023GAYB048); 2024 Guangxi Higher Education Undergraduate Teaching Reform Project: "Student centered" New Era Public Security College General Education Upgrade Path, Reform and Practice of Dual wheel Driven Curriculum System(2024JGB426); 2024 Guangxi Higher Education Undergraduate Teaching Reform Project: Teaching Reform and Practice of "Specialized Creation Integration" in the Course of Public Management under the Background of New Liberal Arts(2024JGA400).

References

[1] Xu W, Xu J, Proverbs D, Zhang Y. A hybrid decision-making approach for locating rescue materials storage points under public emergencies. Kybernetes, 2024, 53(1): 293-313. DOI:10.1108/k-07-2022-1060

[2] Rhodes C, Liu C, Westoby P, Chen W H. Autonomous search of an airborne release in urban environments using informed tree planning. Autonomous Robots, 2023, 47(1): 1-18. DOI:10.1007/s10514-022-10063-8

[3] Peixoto J P J, Costa D G, Portugal P, Vasques F. Flood-resilient smart cities: a data-driven risk assessment approach based on geographical risks and emergency response infrastructure. Smart Cities, 2024, 7(1): 662-679. DOI:10.3390/smartcities7010027

[4] Sun B, Zhou Y. Bayesian network structure learning with improved genetic algorithm. International Journal of Intelligent Systems, 2022, 37(9): 6023-6047. DOI:10.1002/int.22833

[5] Yun Y, Gen M, Erdene T N. Applying GA-PSO-TLBO approach to engineering optimization problems. Math. Biosci. Eng, 2023, 20(1): 552-571. DOI:10.3934/mbe.2023025

[6] Jose C, Vijula Grace K S. Optimization based

routing model for the dynamic path planning of emergency vehicles. Evolutionary Intelligence, 2022, 15(2): 1425-1439. DOI:10.1007/s12065-020-00448-y

[7] Zhao X, Wang G. Deep Q networks-based optimization of emergency resource scheduling for urban public health events. Neural Computing and Applications, 2023, 35(12): 8823-8832. DOI:10.1007/s00521-022-07696-2

[8] Liu J, Li Y, Li Y, Zibo C, Lian X, Zhang Y. Location optimization of emergency medical facilities for public health emergencies in megacities based on genetic algorithm. Engineering, construction and architectural management, 2022, 30(8): 3330-3356. DOI:10.1108/ecam-07-2021-0637

[9] Xia H, Sun Z, Wang Y, Zhang J Z, Kamal M M, Jasimuddin S M, et al. Emergency medical supplies scheduling during public health emergencies: algorithm design based on AI techniques. International Journal of Production Research, 2025, 63(2): 628-650. DOI:10.1080/00207543.2023.2267680

[10] El-Shafiey M G, Hagag A, El-Dahshan E S A, Ismail M A. A hybrid GA and PSO optimized approach for heart-disease prediction based on random forest. Multimedia Tools and Applications, 2022, 81(13): 18155-18179. DOI:10.1007/s11042-022-12425-x

[11] Shen L, Zhang Z, Song X, Yin Y. A hybrid GA-PSO algorithm for seru scheduling problem with dynamic resource allocation. International Journal of Manufacturing Research, 2023, 18(1): 100-124. DOI:10.1504/ijmr.2023.10040833

[12] Bousnina K, Hamza A, Ben Yahia N. A combination of PSO-ANN hybrid algorithm and
genetic algorithm to optimize technological parameters during milling 2017A alloy. Journal of Industrial and Production Engineering, 2023, 40(7): 554-571. DOI:10.1080/21681015.2023.2243312

[13] Sun H, Li H, Meng Z, Wang D. An improvement of dv-hop localization algorithm based on improved adaptive genetic algorithm for wireless sensor networks. Wireless Personal Communications, 2023, 130(3): 2149-2173. DOI:10.1007/s11277-023-10376-6

[14] Zhi Y, Wang H, Wang L. A state of health estimation method for electric vehicle Li-ion batteries using GA-PSO-SVR. Complex & Intelligent Systems, 2022, 8(3): 2167-2182. DOI:10.1007/s40747-021-00639-9

[15] Aloqaily M, Elayan H, Guizani M. C-healthier: A cooperative health intelligent emergency response system for c-its. IEEE Transactions on Intelligent Transportation Systems, 2022, 24(2): 2111-2121. DOI:10.1109/tits.2022.3141018

[16] Pillai B M, Suthakorn J, Sivaraman D, Nakdhamabhorn S, Nillahoot N, Ongwattanakul S, et al. A heterogeneous robots collaboration for safety, security, and rescue robotics: e-ASIA joint research program for disaster risk and reduction management. Advanced Robotics, 2024, 38(3): 129-151. DOI:10.1080/01691864.2024.2309622

[17] KA A, Subramaniam U. A systematic literature review on multi-robot task allocation[J]. ACM Computing Surveys, 2024, 57(3): 1-28. DOI:10.1145/3700591

[18] Abujabal N, Fareh R, Sinan S, Baziyad M, Bettayeb M. A comprehensive review of the latest path planning developments for multi-robot formation systems[J]. Robotica, 2023, 41(7): 2079-2104. DOI:10.1017/s0263574723000322

[19] Singh G, Chaturvedi A K. Hybrid modified particle swarm optimization with genetic algorithm (GA) based workflow scheduling in cloud-fog environment for multi-objective optimization. Cluster Computing, 2024, 27(2): 1947-1964. DOI:10.1007/s10586-023-04071-1