Journal of Systems Engineering and Electronics
Vol. 37, No. 3, June 2026, pp.964 – 973

Multi-stage forest UAV route design based on multi-strategy GA

XU Wangying and XIE Naiming*

College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 210006, China

Abstract: Forest fires are characterized by their abrupt onset and highly destructive nature, resulting in significant annual property losses. Hence, regular surveillance is imperative for forest fire prevention and mitigation. The fundamental challenge in patrolling is akin to the problem of helicopter route planning. Conventional unmanned aerial vehicle (UAV) path planning commonly entails single-trip missions. Considering the extensive and complex forest environments, we advocate a multi-stage UAV reconnaissance strategy to address the daily inspection route planning conundrum. This approach facilitates UAVs to conduct round-trip flights between designated surveillance points and the base station at diverse time intervals, effectively satisfying the requirements for multi-tiered, hierarchical reconnaissance. Furthermore, we develop an advanced multi-strategy genetic algorithm (MSGA) to optimize the multi-stage reconnaissance model. Experimental outcomes underscore the superior performance of the enhanced MSGA, achieving a reduction of nearly 20% in total flight path length relative to the traditional genetic algorithm. This methodology significantly enhances the efficacy of daily forest patrols.

Keywords: unmanned aerial vehicle (UAV), path planning, forest scouting, genetic algorithm, optimization strategy.

DOI: 10.23919/JSEE.2026.000119

1. Introduction

In recent years, the occurrence of forest fires has aroused widespread concern around the world [1,2]. However, such incidents cannot be completely resolved. As an emerging transportation equipment, unmanned aerial vehicles (UAVs), with the advantages of flexible operation and reliable communication, occupy a unique advantage in forest fire prevention [3–5]. UAVs not only reduce the society's resources, but also prevent personnel close to the dangerous area in the daily inspection, thus guaranteeing the safety of people's property [6]. UAV path planning entails leveraging satellite imagery and related tools to comprehensively survey the topography of a mountain or grassland. Subsequently, a route is meticulously devised based on the desired exploration direction, optimizing for the minimal cost criteria. In the context of UAV operations, the cost metric may encompass factors such as the shortest flight path length or overall flight expenditure [7].

Currently, several methods are presented to solve the UAV path planning problems. Firstly, greedy algorithms. Common greedy algorithms include A* algorithm [8–10], Dijkstra's algorithm [11,12] and so on. These algorithms utilize the greedy idea of an algorithm, the number of all combinations of journey routes is $(n-1)!/2$ [13]. For example, Prasad et al. [14] proposed a 3-D deployment and trajectory planning algorithm. It uses power savings, reduced latency, and power savings through optimal trajectories to evaluate some of the performance metrics of the proposed algorithm. Secondly, based on computationally assisted approaches, such as artificial potential field (APF) [15,16]. Chen et al. [17] derived the whole transformation process in detail based on a discrete UAV dynamics model, and applied the functional optimization method to transform the problem into an optimal control problem based on APF. Shi et al. [18] proposed a learning-based heuristic algorithm, the population-based adaptive large neighborhood search (P-ALNS) algorithm. In P-ALNS, seven neighborhood structures are designed and adaptively utilized based on their historical performance. Thirdly, Optimization algorithms. For example, genetic algorithm (GA) [19,20], artificial bee colony (ABC) algorithm [21,22] and so on. Li et al. proposed a heterogeneous UAV collaborative multitasking model based on area segmentation using an improved GA (IGA) algorithm for collaborative human task optimization [23]. Yu et al. [24] developed a hybrid grey wolf optimization and

Manuscript received April 12, 2024.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (72571138; T2441003), the National Key R&D Program (2026YFE0153500), the State Grid Zhejiang Electric Power Co., Ltd. Technology Project (B311DS25Z004), and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX25_0650).

differential evolution (HGWODE) algorithm to solve the UAV path planning problem. A ranking-based variation strategy is implemented in the differential evolution (DE) algorithm to facilitate development while maintaining exploration capability. Zou et al. proposed an accurate path planning method for weeding UAVs based on UAV images, using an improved mutation operator GA (IMO-GA) to optimize the operation path [25].

All the above methods have some shortcomings. For example, the search space of the greedy algorithm increases with the increase of the number of populations n. When n is large, it is usually difficult for the greedy algorithm to find the shortest path by traversing [26,27]. The APF method relies on the interplay between attraction and repulsion forces, which is easy to fall into local optimization [15]. On the other hand, certain platform-assisted algorithms necessitate a substantial foundation in machine learning, posing challenges in acquiring extensive learning datasets. Optimization algorithms show the advantage of such algorithms lies in their fast-solving speed [28]. However, they may only converge to a local optimum and fail to reach the global optimal solution of the stochastic nature of the search process [29].

Based on the above discussion, we propose three innovations in this paper. Firstly, based on the value difference of different protected areas, a hierarchical route planning is creatively carried out. A multi-stage UAV inspection hierarchical model is established, which requires key protected areas to be inspected twice a day and general protected areas to be inspected once a day. Secondly, a multi-strategy GA (MSGA) is proposed. To prevent the prematurity phenomenon in the iteration, we propose a multi-strategy crossover approach along with an enhanced mutation operator in MSGA. Compared with the traditional GA, the accuracy and convergence of the MSGA are improved. Finally, three different dimensions problems of the cases are introduced, and a sensitive test is validated by the MSGA on the model. The discussion is deductively used to validate the effectiveness of the MSGA compared to other heuristic algorithms.

The remaining sections of this paper will be unfolded sequentially. Section 2 describes a single UAV multi-stage forest exploration trajectory planning model. Section 3 introduces the MSGA. Section 4 comprises experimental cases of varying dimensions. To validate the effectiveness of the algorithm enhancements, sensitivity analysis will be conducted on each strategy. Section 5 provides a comprehensive summary of the entire paper.

2. UAV multi-stage inspection model

2.1 Description of the problem

Preventing forest fires through scheduled forest patrols is a crucial measure for fire prevention. UAVs have become essential tools for forest patrols due to their small size, ease of operation, and cost-effectiveness compared to manual methods [30].

It is assumed that there are several important conservation sites within the jurisdiction of a forest farm, which are regarded as patrolling points and a single UAV is dispatched to perform the patrolling work. High-level inspections need to be patrolled regardless of day or night. Low-level inspections need to be patrolled only once on the same day, regardless of day or night.

Fig. 1 illustrates the operation of the UAV. There are 13 low-level inspections and four high-level inspections. The first-round and second-round routes influence each other. It is proposed to assign a value of “1” to all low-level inspections visited by the UAV in the first traversal and apply the complement code method to assign a value of “0” to the low-level inspections visited by the UAV in the second traversal. As shown in Table 1, there are currently 13 low-level inspections that need to be patrolled, and the system randomly designates points 1, 2, 5, 6, 7, 9, 10, and 13 as the points to be visited by the UAV in the first traversal. Consequently, the points 3, 4, 8, 11, and 12 need to be visited in the second traversal.

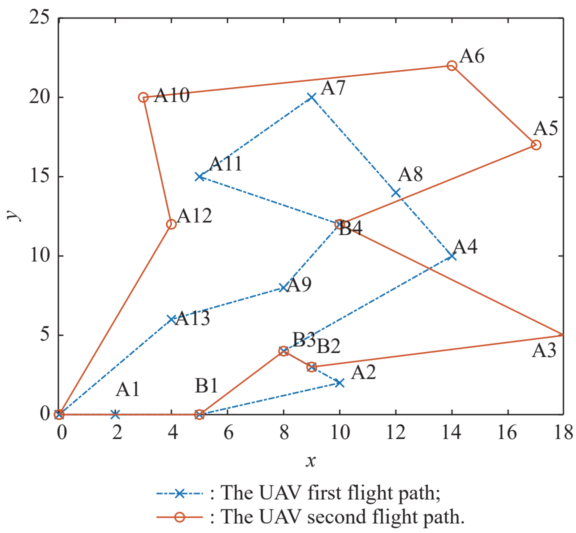


Fig. 1 Schematic diagram of UAV patrol scattering points

Table 1 Randomized categorization of single UAV multi-stage inspections

Inspection point	1	2	3	4	5	6	7	8	9	10	11	12	13
Encoding	1	1	0	0	1	1	1	0	1	1	0	1	0

2.2 Mathematical model

The mathematical problem can be abstracted as an improved multi-travelling salesman problem (MTSP) problem [31]. TSP is an NP-hard problem for which no polynomial time algorithm exists. There is no efficient algorithm for solving the TSP exactly for large networks, only algorithms that derive better solutions. Since the TSP problem is solved classically and the solutions are diversified, the UAV path planning is often transformed into a TSP problem by eliminating the subtle influences. The common model of the TSP problem is

$$\min_{j_1,j_2,\cdots,j_n\in\pi(n)}\sum_{i=1}^{n}d_{j_i,j_{i+1}}+d_{j_n,j_1} \tag{1}$$

where $\pi(n)$ is denoted as the set consisting of all full permutations of $1,2,\cdots,n$. In summary, the parameters of the path planning method for the UAV are set as Table 2.

Table 2 Parameter description

Parameter	Description
A_i	The ith low-level inspections
B_j	The jth high-level inspections
ξ	Number of x_i if $x_i = 1$
η	Number of x_i if $x_i = 0$
$\boldsymbol{P}$	Binary vectors for flight groups
M	Number of high-level inspections
N	Number of low-level inspections
d_{A_m,A_n}	The distance between A_m and A_n
d_{A_m,B_n}	The distance between A_m and B_n
$x_{m,n,k}$	1, the UAV passes through A_m and A_n during the kth round of inspection; 0, else
$y_{i,j,k}$	1, the UAV passes through A_i and B_j during the kth round of inspection; 0, else

The model is modeled as follows:

$$\min f(x)=f_1(x)+f_2(x) \tag{2}$$

$$f_1(x)=\min\sum_{k=1}^{2}\sum_{m=1}^{\xi}d_{x_{m,m+1,k}}+d_{x_{\xi,1,k}}+\sum_{k=1}^{2}\sum_{m=1}^{\eta}d_{x_{m,m+1,k}}+d_{x_{\xi,1,k}} \tag{3}$$

$$f_2(x)=\min\sum_{k=1}^{2}\sum_{j=1}^{M}\sum_{i=1}^{N}d_{A_i,B_j}\cdot y_{i,j,k} \tag{4}$$

s.t.

$$\xi+\eta=2M+N \tag{5}$$

$$0\leqslant d_{A_m,A_n}\leqslant 100 \tag{6}$$

$$0\leqslant d_{A_m,B_n}\leqslant 100 \tag{7}$$

$$\sum_{k=1}^{2}x_{m,n,k}=1 \tag{8}$$

where (2) represents the fitness function composed of two parts; (3) denotes the distance flown between low-level inspections; (4) represents the sum of distances flown from high-level inspections to low-level inspections; (5)−(8) are the constraints; (5) states that the sum of decision variables equaling 0 and 1 should be equal to the sum of twice visiting high-level inspections and once visiting low-level inspections; (6) and (7) indicate that the flying distance between inspection points should be less than the UAV’s maximum flight distance of 100 km; (8) specifies that all low-level inspections must be visited exactly once.

3. MSGA

In this section, based on the shortcomings of the traditional GA in the crossover and mutation phases, MSGA is proposed, and two strategies are proposed for these two phases respectively to improve the convergence effect of the algorithm. A multi-strategies crossover is proposed for the impact that the choice of crossover probability in the crossover process will have on the offspring. An adaptive mutation operator F is proposed which improves the efficiency of global iteration.

3.1 Improvement of the mutation operator *F*

The mutation operation involves selecting an individual in the population and randomly changing the gene segments at one or more positions of the current stain according to the mutation probability. As shown in Fig. 2, a schematic diagram of the chromosomal gene mutation process is shown.

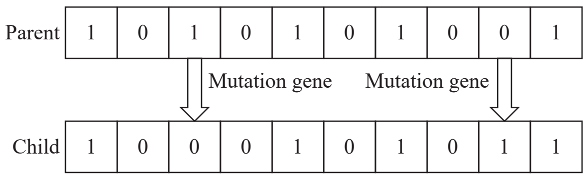


Fig. 2 Process of mutation of chromosomal gene segments

The mutation process imitates the mutation process of gene chromosomes, which is an improvement of the GA to repair and replenish certain genetic genes that may be lost in the crossover process. This strategy helps prevent rapid convergence to local optima. Similar to crossover probability, an inappropriate mutation rate can significantly impact the convergence behavior of the algorithm and the quality of the final solution. A low mutation rate may trap the algorithm in local optima, while a high mutation rate can destabilize the population. Therefore, the selection of the appropriate variation probability needs to take into account the complexity of the problem and the demand for the global optimal solution [32]. Generally, a lower mutation probability can be explored to a certain extent while achieving stability. A higher mutation probability can increase the diversity of the population, but it may also lead to slower convergence of the algorithm. In practical applications, repeated experiments and adjustments need to be made according to specific problems to find the optimal mutation probability.

This paper proposes an adaptive improvement mutation strategy to solve the above problems. At the initial state, given the value of $F_0, F_{i,G}$ can be expressed as

$$F_{i,G} = F_0 2^{\lambda},\ \ \lambda = \mathrm{e}^{1-\frac{G_m}{G_m+1-G}} \tag{9}$$

where F_0 is fixed and takes the value 0.4, a λ is the exponential term, and G_m denotes the total number of generations. G denotes the current evolutionary generation. In the early stage, the starting mutation operator is about $2F_0$.

In the early stage, the mutation operator is larger at this time, which can maintain the diversity of populations and avoid early maturity. From the second generation onwards, the algorithm is updated. We design (9) to confirm the availability of the variational operator. However, as the probability of variability increases at a later stage, it is possible to increase the time for the solution space to converge, affecting the efficiency of the iteration. Therefore, (10) is designed to set to monitor the $F_{i,G}$ operator. The F variation operator expression is as follows:

$$F_{i,G} = \begin{cases} F_l + kF_u, & 0 \leqslant c < 0.1 \\ F_{i,G}, & \text{others} \end{cases}. \tag{10}$$

Generally, the largest possible variance operator difference is taken for the diversity of variance operator changes in the first, middle, and late generations, therefore, the F_l is set to 0.1, and F_u is set to 0.9. k is a random number which belongs to (0,1). c is a regulator which belongs to [0,1]. When the value of c is less than 0.1, a mutation is applied to the mutation operator.

3.2 Improvements in the crossing strategy

As indicated in Section 2, binary encoding is often employed to represent individual solutions within the population when solving the TSP problem. In contrast to conventional GA crossover procedures, the crossover operation for solving the TSP involves a non-continuous integer transformation, rendering the use of standard continuous crossover strategies inadequate for optimal performance.

In the conventional TSP, the crossover strategy often involves randomly selecting a crossover point and performing a chromosome transformation based on that point. This simplistic approach lacks diversity in the population. The flexibility of algorithms is routinely enhanced using multiple crossover strategies. Therefore, this section proposes an enhanced multi-strategy crossover approach to address this limitation. The crossover process introduces three strategies. And a random number is used to specify exactly which crossover strategy is used. The specific three strategies are as follows:

(i) Strategy 1: randomly take the intersection nodes and exchange the solution sets of the left and right sides of the intersection nodes, as shown in Fig. 3. m is the total number of tour points for the first tour; n is the total number of tour points for the second tour.

Parent	$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{1,m}$	$x_{1,m+1}$	$\cdots$	$x_{1,n}$
Child	$x_{1,m+1}$	$\cdots$	$x_{1,n}$	$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{1,m}$

Fig. 3 Schematic diagram of cross-strategy 1

(ii) Strategy 2: randomly take two parents $f1$ with $f2$, since the overall solution is composed of the low-level inspections from the first and second inspections, as shown in Fig. 4.

Parent1	$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{1,m}$	$x_{1,m+1}$	$\cdots$	$x_{1,n}$
Parent2	$x_{2,1}$	$x_{2,2}$	$\cdots$	$x_{2,m}$	$x_{2,m+1}$	$\cdots$	$x_{2,n}$
Child	$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{2,m}$	$x_{2,m+1}$	$\cdots$	$x_{2,n}$

Fig. 4 Schematic diagram of cross-strategy 2

(iii) Strategy 3: keep the original parent generation unchanged and invert the original parent solution with probability 1/2, as shown in Fig. 5.

Parent	$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{1,m}$	$x_{1,m+1}$	$\cdots$	$x_{1,n}$
Child1	$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{1,m}$	$x_{1,m+1}$	$\cdots$	$x_{1,n}$
Child2	$x_{1,n}$	$\cdots$	$x_{1,m+1}$	$x_{1,m}$	$\cdots$	$x_{1,2}$	$x_{1,1}$

Fig. 5 Schematic diagram of cross-strategy 3

The three crossover strategies are performed in the MSGA as following:

$$\mathrm{CR} = \begin{cases} \text{strategy 1}, & 0 < m < \dfrac{1}{3} \\ \text{strategy 2}, & \dfrac{1}{3} \leqslant m \leqslant \dfrac{2}{3} \\ \text{strategy 3}, & \dfrac{2}{3} \leqslant m < 1 \end{cases} \tag{11}$$

where m is a random number belonging to (0, 1). The three crossover-strategies improves the diversity of the algorithm's direction of optimization in the iterative process, thus increasing the probability of convergence to the global optimum point. The pseudo-code of the MSGA is shown as Algorithm 1.

Algorithm 1

Input Number of populations NP, population dimensions dim, variation operator coefficient F_0, number of runs, max generation, binary vector $\boldsymbol{P}$ of the same size as the number of basic points

1. **For** i = 1: runs
2. Based on the vector $\boldsymbol{P}$, the UAV route is divided into the sets a and b
3. **For** ii = 1:NP
4. Initialize each population according to set a and set b, and calculate the total distance traveled by the UAV
5. **End**
6. **while** t <= Max generation
7. Use roulette to randomly select the parents x_1 and x_2, and calculate the fitness value;
8. Multi-strategy crossover operation on the parent based on (11);
9. Mutational manipulation of the parent generation based on (3);
10. By incorporating an elite strategy, the comparison of fitness values between offspring and parents is conducted to determine retention or replacement.
11. $t = t+1$
12. **End**
13. Use external storage to retain the smallest individual and its fitness value from each iteration
14. **End**

Combined with the MSGA, the multi-stage forest scouting trajectory planning method for the UAV is shown in Fig. 6.

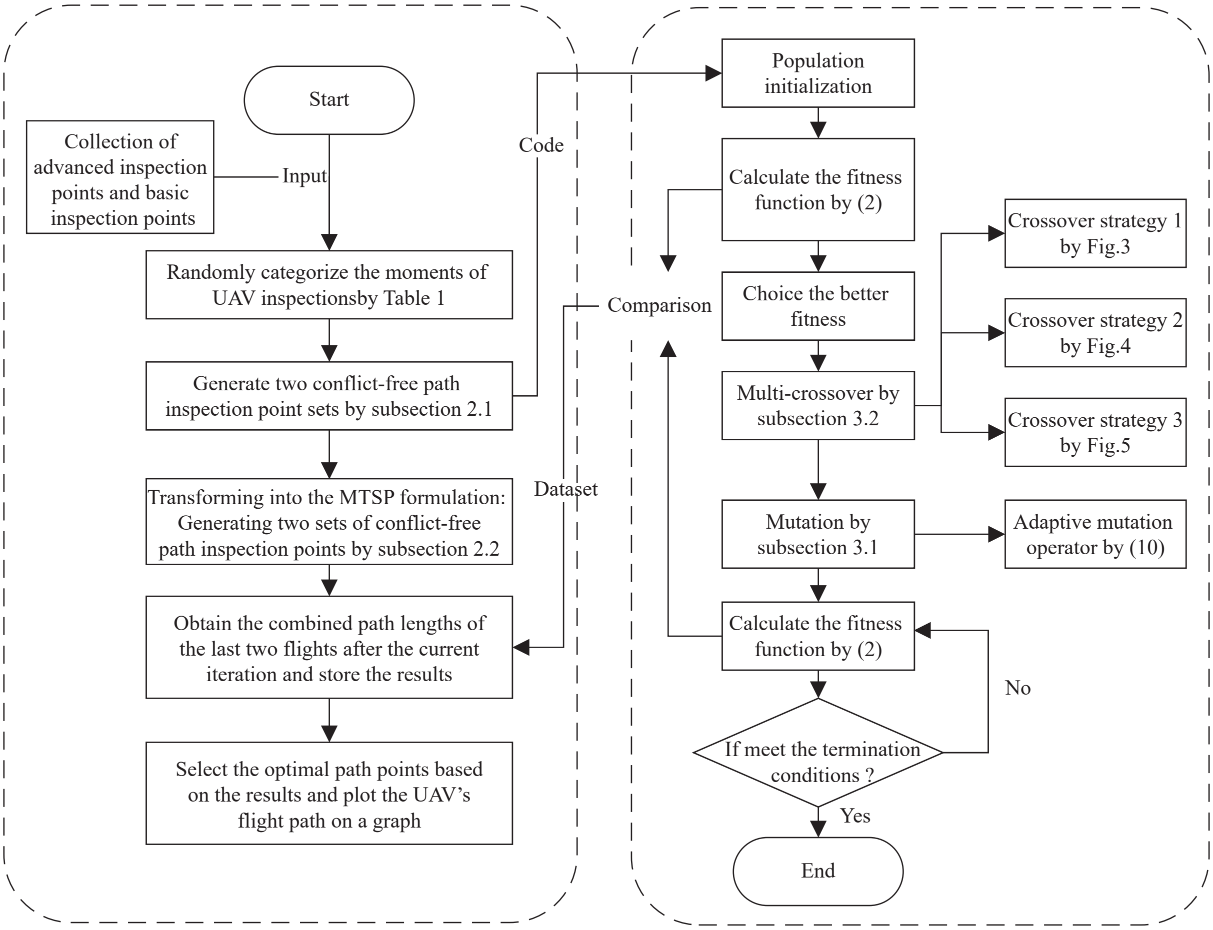


Fig. 6 Two-stage forest scouting trajectory planning method for a single UAV

4. Experiments

To verify the effectiveness of the MSGA, the traditional GA and the improved MSGA are compared. The simulation experiment environment is a computer with a win11 64-bit operating system, i5-8500U processor, and 8GB RAM.

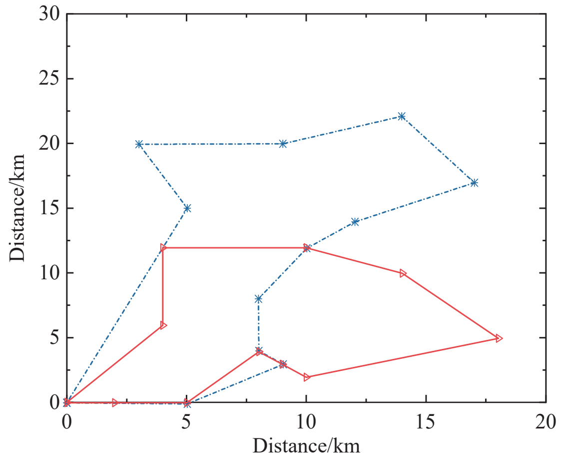
To comprehensively depict the performance of the algorithms, three different-dimension cases are proposed in the experiment. Each case has two different inspection point levels. In this section, the MSGA is used to solve the UAV path planning to verify the effectiveness of the MSGA. We take the total flight distance as the performance index of the algorithm to judge whether the improvement of the MSGA is effective. Table 3 shows the specific parameter of the three experiments. The environments are modeled as follows: The starting point is [0,0]. Set the population size as 20, the number of iterations as 1 000, run 50 times, and record the adaptation value of each generation, respectively.

Table 3 Parameters of the experiments

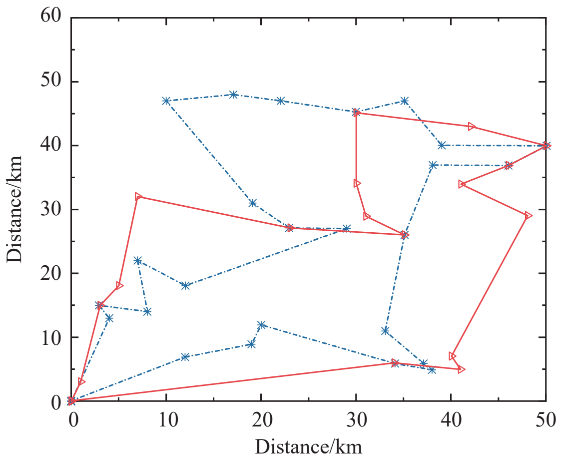
Experiment	Number of high-level inspection points	Number of low-level inspection points	Inspection space
Case 1	4	13	10 km×10 km
Case 2	6	30	50 km×50 km
Case 3	10	50	50 km×50 km

4.1 Experimental results

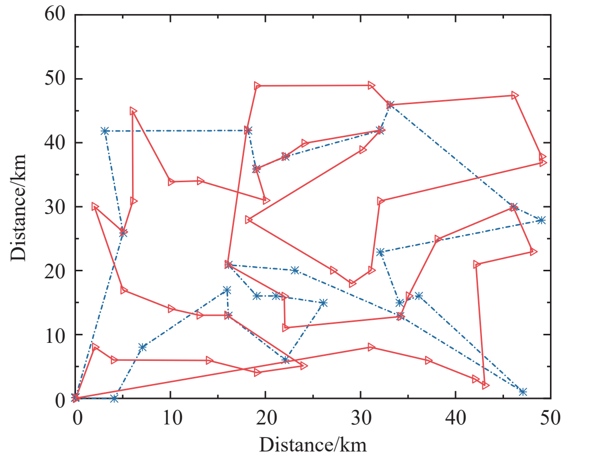
During the actual inspection process, the number of inspection points will vary depending on the size of the forest. Fig. 7 is the map of routes for scheduling multi-stage forest inspections.



(a) Case 1



(b) Case 2



(c) Case 3

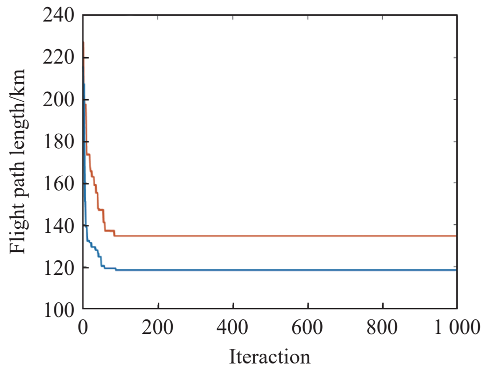
: The UAV first flight path; : The UAV second flight path.

Fig. 7 Routes for scheduling multi-stage forest inspections

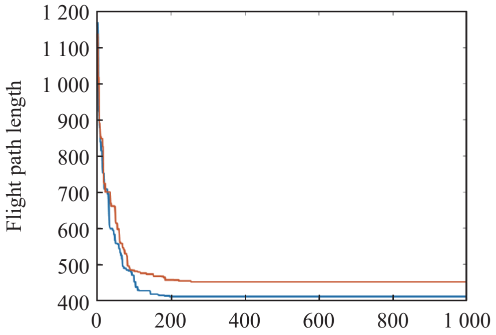
Table 4 describes the comparison of the flight distances calculated by different algorithms. Fig. 8 denotes the iteration of GA and MSGA. As depicted in Fig. 8(c), the iterative performance of the enhanced MSGA remains notably superior. According to the Table 4, the total flight path length for the MSGA is approximately 410 km, whereas the total path length of traditional GA is around 467 km, representing a 12.2% increase over the flight path length of the MSGA algorithm. Moreover, in the iterations of this case, the flight distances for the MSGA are consistently shorter than those of the GA.

Table 4 Comparison of the flight distances calculated by different algorithms km

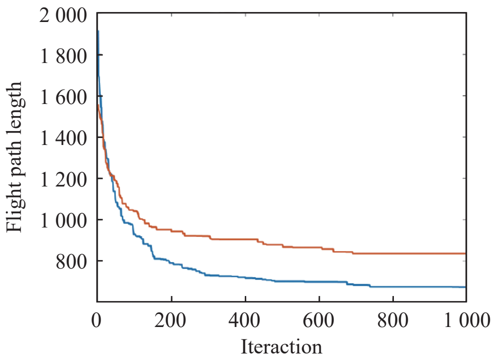
Experiment	Algorithm	First flight distance	Second flight distance	Total distance
Case 1	Traditional GA	62.039 7	72.536 5	134.576 2
	MSGA	67.413 5	50.896 8	118.310 3
Case 2	Traditional GA	258.091 1	209.117 1	467.208 2
	MSGA	220.556 8	190.363 9	410.920 7
Case 3	Traditional GA	346.823 1	480.924 5	827.747 6
	MSGA	252.278 7	394.702 1	646.980 8



(a) Case 1



(b) Case 2



(c) Case 3

: MSGA; : GA.

Fig. 8 Iteration of the flight distance

Fig. 8 is the iteration of the flight distance. As seen in Fig. 8(a), since the 100th generation, both the GA and the MSGA have reached a stable iteration to find the optimal objective. In terms of the final fitness value of the two algorithms, the MSGA has the shortest total path length. Therefore, the MSGA improves and enhances the traditional GA significantly.

4.2 Sensitivity analysis

To compare whether each strategy of the proposed MSGA algorithm is effective or not, this section analyzes the sensitivity of each improved strategy. The experiment adopts the control variable method to compare the four algorithms, namely, the traditional GA, GA with the mutation improvement, GA with the multi-crossover strategy, and the MSGA, respectively. The four algorithms are run independently fifty times, and the results of each run are recorded. To highlight the performance of the algorithms, we select case 3 as a calculation example of sensitivity test.

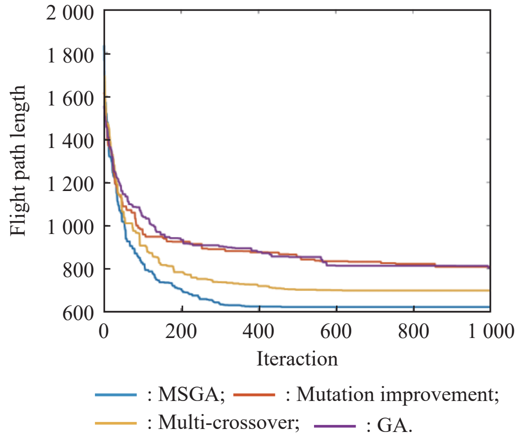


Fig. 9 Comparison of 60-dimensional MSGA algorithm sensitivity analysis iterations

As shown in Table 5, in the 60-dimensional case, the total flight distances obtained by the four algorithms are 807 km, 699 km, 810 km, and 616 km, respectively. The MSGA demonstrates the shortest scheduling route through iterations in Fig.9. Notably, in the later stages, there is a considerable gap in the stable iteration values among the algorithms. The MSGA achieves stability around the 400th generation, followed by the sole improvement of the F operator strategy stabilizing around the 450th generation, and lastly, the enhancements in the multi-crossover and the traditional GA stabilize from the 400th generation onwards. In this case, the crossover strategy appears to play a more significant role in enhancing the optimization performance of the MSGA, as indicated by the convergence patterns observed.

Table 5 Comparison of sensitivity analysis of the distance km

Algorithm	First flight distance	Second flight distance	Total distance
Traditional GA	362.6364	445.1062	807.7426
Mutation improvement	338.5164	361.1602	699.6766
Multi-crossover	480.9235	329.8601	810.7836
MSGA	252.2787	394.7021	646.9808

Fig. 10 shows the flight paths of the four algorithms. Since case 3 belongs to the high-dimensional arithmetic cases, a certain degree of difficulty is existed in finding the optimal point. As depicted in Fig. 10(a), the GA struggles to find the global optimum during iterations of high-dimensional problems due to its utilization of fixed crossover and mutation operators. Upon comparing Fig. 10(a), Fig. 10(b), Fig. 10(c), and Fig. 10(d), it is evident that the flight routes generated by the improved MSGA exhibit fewer detours and clearer paths compared to the other three algorithms. Particularly in Fig. 10(a), the traditional GA shows multiple back-and-forth routes, increasing the flight distance and leading to unnecessary wastage during the UAV flight process.

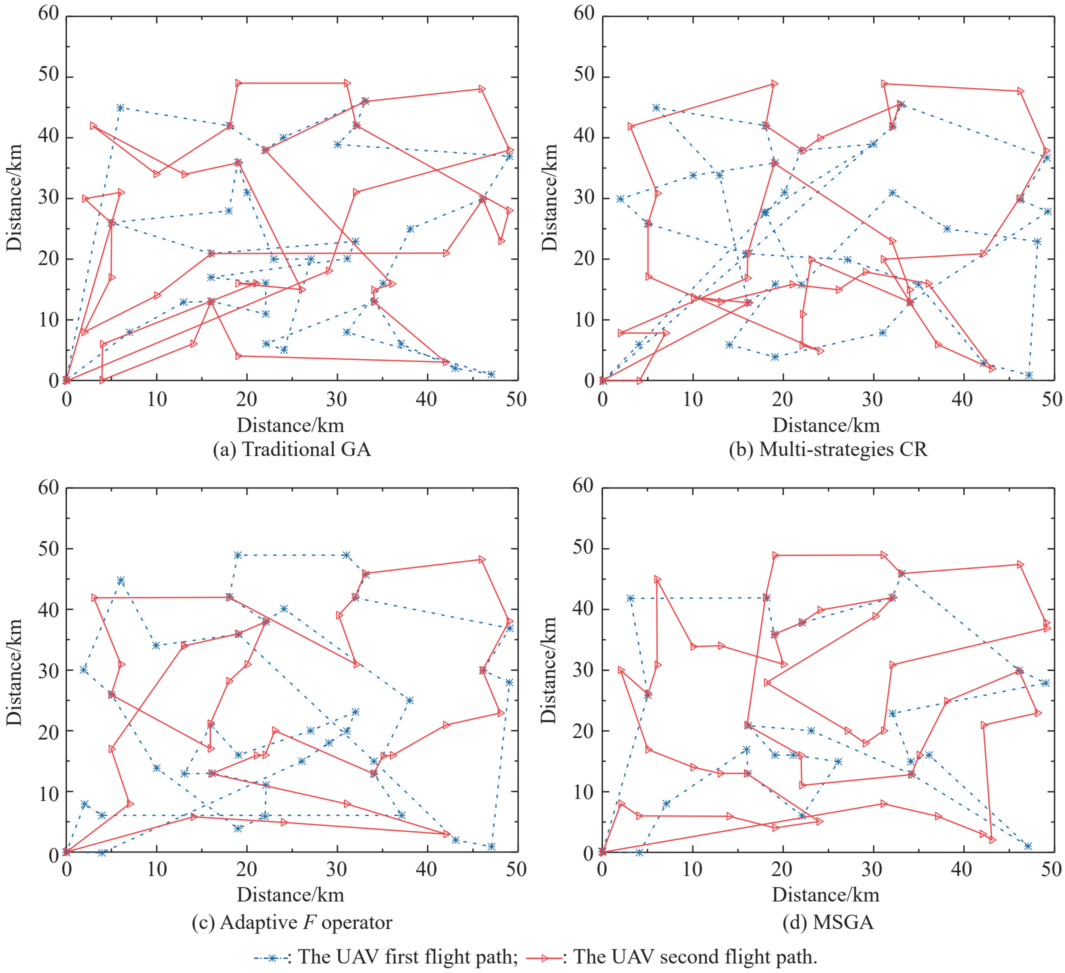


Fig. 10 Flight schematic of the four algorithms UAV path scheduling

Fig. 11 shows a comprehensive comparison of the effects of the improved strategies on the fitness values in 60 dimensions. The overall maximum, minimum, and average values of the four algorithms are plotted for comparison. Traditional GA approaches struggle to reach the global optimum solely based on fixed mutation and crossover probabilities, leading to significant disparities in algorithm outcomes. This observation suggests that enhancing the F operator yields a greater enhancement in fitness values for the MSGA.

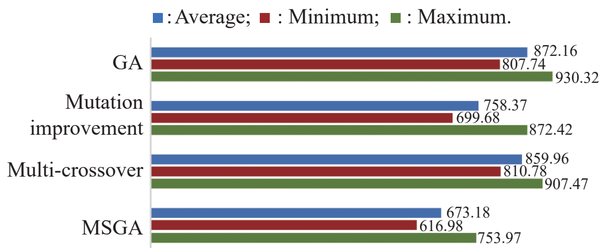


Fig. 11 Comparison of the effect of the improvement strategy on the fitness value

4.3 Discussion

To verify the generalizability and effectiveness of the proposed algorithms, three common heuristic algorithms, including P-ALNS [18], IGA [23], HGWODE [24], IMO-GA [25] and traditional GA are compared with the MSGA. The 60-dimensional case with the highest optimization difficulty is selected for comparison. Set the population size as 20, the number of iterations as 1 000, run 20 times, and record the adaptation value of each generation, respectively.

The performance of the algorithm can be shown by the optimization results. Five conventional algorithms are compared with MSGA in Table 6. As shown in Fig. 12, MSGA performs the best in 60 dimensions experimental test followed by IGA and P-ALNS algorithm. IMO-GA and GA performs poorly. This fully demonstrates the effectiveness of our proposed algorithm in solving the multi-stage forest probing problem.

Table 6 Distance comparison of different algorithms km

Algorithm	First flight distance	Second flight distance	Total distance
P-ALNS	292.53	458.10	750.63
IGA	295.64	404.92	700.56
HGWODE	316.65	447.08	763.73
IMO-GA	364.62	431.11	795.73
GA	346.82	480.92	827.74
MSGA	252.28	394.70	646.98

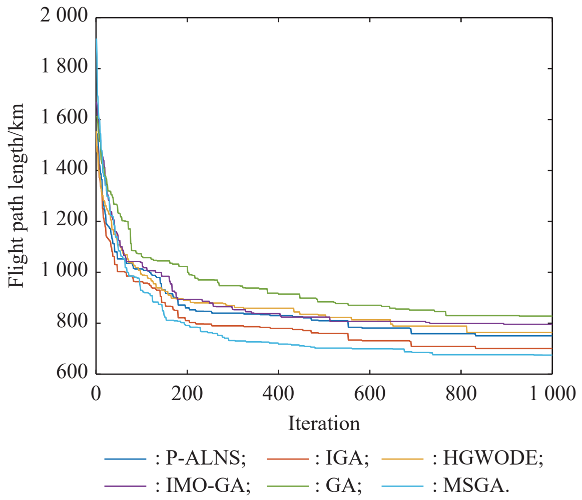


Fig. 12 Iterations of different algorithms 60-dimensional multi-stage problem

Quality analysis is a very crucial part of the empirical aspect of the algorithm. It is mainly used to assess the performance and effectiveness of the algorithm. In this section, we use *t*-test, which is a common method to compare the overall performance of algorithms. The t-test is used to verify the significant difference between the proposed MSGA algorithm and its rivals. The test is based on a level of 0.05. The comparison results are listed in Table 7. "–" signifies that the algorithms perform worse than the MSGA. Overall, the proposed MSGA outperforms most of the opposing adversaries.

Table 7 t-test results of MSGA

Dimension	Comparison algorithm
17D	P-ALNS(−); IGA(−); HGWODE(−); IMO-GA(−); GA(−);
36D	P-ALNS(−); IGA(−); HGWODE(−); IMO-GA(−); GA(−);
60D	P-ALNS(−); IGA(−); HGWODE(−); IMO-GA(−); GA(−);

5. Conclusions

Daily inspection is fundamental to forest fire prevention. The heterogeneous nature of forest properties necessitates differentiated scheduling tasks based on their distinct values. This paper applies UAV technology to daily forest fire detection across various periods. Three key innovations are presented:

(i) Considering the timely nature of daily forest fire prevention inspections, we innovatively construct a multi-stage forest inspection model. A hierarchical multi-stage inspection methodology is proposed.

(ii) To address the issue where traditional GA are prone to falling into local optima, we have designed the MSGA algorithm, featuring multi-strategy crossover and adaptive mutation operators.

(iii) Experimental results from three case studies demonstrate that the MSGA algorithm yields more reasonable outcomes in UAV path planning. The discussion verifies the superiority of this algorithm over others.

This paper is devoted to solving the multi-stage inspection of a single UAV. In the future, we will consider multiple UAV patrol missions in phases with different goals. Combined with online communication equipment, dynamic UAV inspection process will be set up.

References

[1] REIS H C, TURK V. Detection of forest fire using deep convolutional neural networks with transfer learning approach. Applied Soft Computing, 2023, 143: 110362.

[2] YANG X B, HUA Z C, ZHANG L, et al. Preferred vector machine for forest fire detection. Pattern Recognition, 2023, 143: 109722.

[3] SANKEY T T, TANGO L, TATUM J, et al. Forest fire, thinning, and flood in wildland-urban interface: UAV and lidar-based estimate of natural disaster impacts. Landscape Ecology, 2024, 39(3): 58.

[4] SUDHAKAR S, VIJAYAKUMAR V, KUMAR C S, et al. Unmanned aerial vehicle (UAV) based forest fire detection and monitoring for reducing false alarms in forest-fires. Computer Communications, 2020, 149: 1–16.

[5] WANG H, FENG Y, HUANG X, et al. An AI-based action detection UAV system to improve firefighter safety. Proc. of the International Conference on Human-Computer Interaction, 2023: 632−641.

[6] MOSHREF-JAVADI M, HEMMATI A, WINKENBACH M. A truck and drones model for last-mile delivery: a mathematical model and heuristic approach. Applied Mathematical Modelling, 2019, 80: 256–267.

[7] CHEN R, CHEN J, XU Q Y. Joint channel access and power control optimization in large-scale UAV networks: a hierarchical mean field game approach. IEEE Trans. on Vehicular Technology, 2023, 72(2): 1982–1996.

[8] GUO Y C, LIU X X, LIU X H, et al. FC-RRT*: an improved path planning algorithm for UAV in 3D complex environment. ISPRS International Journal of Geo-Information, 2022, 11: 524–533.

[9] LIU L X, WANG X, YANG X, et al. Path planning techniques for mobile robots: review and prospect. Expert Systems with Applications, 2023, 227: 120254.

[10] ZHANG W, WANG N X, WU W H. A hybrid path planning algorithm considering AUV dynamic constraints based on improved A* algorithm and APF algorithm. Ocean Engineering, 2023, 285: 115333.

[11] FAN D K, SHI P. Improvement of Dijkstra's algorithm and its application in route planning. Proc. of the 7th International Conference on Fuzzy Systems and Knowledge Discovery, 2010: 1901−1904.

[12] KHAKZAD N. A methodology based on Dijkstra's algorithm and mathematical programming for optimal evacuation in process plants in the event of major tank fires. Reliability Engineering & System Safety, 2023, 236: 109291.

[13] KNUTH D E. A Generalization of Dijkstra's Algorithm. Information Processing Letters, 1977, 6: 1–5.

[14] PRASAD N L, RAMKUMAR B. 3-D deployment and trajectory planning for relay based uav assisted cooperative communication for emergency scenarios using Dijkstra's algorithm. IEEE Trans. on Vehicular Technology, 2022, 72(4): 5049–5063.

[15] VADAKKEPAT P, TAN K C, WANG M L. Evolutionary artificial potential fields and their application in real time robot path planning. Proc. of the Congress on Evolutionary Computation, 2000: 256−263.

[16] CHANG H, CHEN Y C, ZHANG B C, et al. Multi-UAV mobile edge computing and path planning platform based on reinforcement learning. IEEE Trans. on Emerging Topics in Computational Intelligence, 2021, 6(3): 489–498.

[17] CHEN Y B, LUO G C, MEI Y S, et al. UAV path planning using artificial potential field method updated by optimal control theory. International Journal of Systems Science, 2016, 47(6): 1407–1420.

[18] SHI J M, ZHANG J M, LEI H T, et al. Joint mission and route planning of unmanned air vehicles via a learning-based heuristic. Journal of Systems Engineering and Electronics, 2023, 34(1): 81–98.

[19] NGUYEN H D, YOSHIHARA I, YAMAMORI K, et al. Implementation of an effective hybrid GA for large-scale traveling salesman problems. IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics), 2007, 37(1): 92–99.

[20] ZHENG J Z, ZHONG J L, CHEN M L, et al. A reinforced hybrid genetic algorithm for the traveling salesman problem. Computers & Operations Research, 2023, 157: 106249.

[21] CHOONG S S, WONG L P, LIM C P. An artificial bee colony algorithm with a Modified Choice Function for the traveling salesman problem. Swarm and Evolutionary Computation, 2019, 44: 622–635.

[22] SAHIN M. Solving TSP by using combinatorial Bees algorithm with nearest neighbor method. Neural Computing and Applications, 2023, 35(2): 1863–1879.

[23] LI J X, YANG X R, YANG Y J, et al. Cooperative mapping task assignment of heterogeneous multi-UAV using an improved genetic algorithm. Knowledge-Based Systems, 2024, 296: 111830.

[24] YU X B, JIANG N J, WANG X M, et al. A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning. Expert Systems with Applications, 2023, 215: 119327.

[25] ZOU K L, WANG F, ZHANG F, et al. Precision route planning method based on UAV remote sensing and genetic algorithm for weeding machine. Applied Intelligence, 2023(53): 11203–11213.

[26] ZHAO Z Y, ZHOU M C, LIU S X. Iterated greedy algorithms for flow-shop scheduling problems: a tutorial. IEEE Trans. on Automation Science and Engineering, 2021, 19(3): 1941–1959.

[27] DIMIC G, SIDIROPOULOS N D. On downlink beamforming with greedy user selection: performance analysis and a simple new algorithm. IEEE Trans. on Signal processing, 2005, 53(10): 3857–3868.

[28] LIU H J, LEE A, LEE W S, et al. DAACO: adaptive dynamic quantity of ant ACO algorithm to solve the traveling salesman problem. Complex & Intelligent Systems, 2023, 9(4): 4317–4330.

[29] GHALI M, ELGHALI S, AIFAOUI N. Genetic algorithm optimization based on manufacturing prediction for an efficient tolerance allocation approach. Journal of Intelligent Manufacturing, 2024, 35(4): 1649–1670.

[30] TANG L N, SHAO G F. Drone remote sensing for forestry research and practices. Journal of Forestry Research, 2015, 26: 791–797.

[31] FURCHì A, LIPPI M, CARPIO R F, et al. Route optimization in precision agriculture settings: a multi-steiner TSP formulation. IEEE Trans. on Automation Science and Engineering, 2022, 9(6): 643.

[32] LIU J H, LAMPINEN J. A fuzzy adaptive differential evolution algorithm. Soft Computing, 2005, 9: 448–462.