UAV Path Planning Based on Enhanced PSO-GA

Hongbo Xiang[1,2], Xiaobo Liu[1,2](✉), Xinsheng Song[1,2], and Wen Zhou[1,2]

[1] Key Laboratory of Geological Survey and Evaluation of Ministry of Education,
China University of Geosciences, Wuhan 430074, China
{hbxiang,xbliu,xinsheng_song,1202211346}@cug.edu.cn
[2] School of Automation, China University of Geosciences, Wuhan 430074, China

Abstract. Path planning for unmanned aerial vehicles (UAV) is a key technology for UAV intelligent system in the aspect of model construction. In order to improve the rapidity and optimality of UAV path planning, we propose a hybrid approach for UAV path planning in 2D environment. First, an enhanced particle swarm optimization algorithm (EPSO) combine with genetic algorithm (GA) which named as EPSO-GA is utilized to obtain the initial paths of UAV. In EPSO-GA, a hybrid initialization of Q-learning and random initial solutions is adopted to find the better initial paths for the UAV, which improves the quality of initial paths and accelerates the convergence of the EPSO-GA. The acceleration coefficients of EPSO-GA are designed as adaptive ones by the fitness value to make full use of all particles and strengthen the global search ability of the algorithm. Finally, the effectiveness of the proposed algorithm is proved by the experiments of UAV path planning.

Keywords: UAV path planning · PSO-GA · Hybrid initialization · Q-learning

1 Introduction

Unmanned aerial vehicle (UAV) path planning refers to find a path between the starting position and the destination under the conditions of terrain, radar, and other factors [1]. In last decades, many methods have been proposed to solve the issue of UAV path planning in 2D environment, which includes Graph-based algorithm [2,3], Rapidly-exploring Random Trees algorithm (RRT) [4], artificial potential field algorithm (APF) [5,6], and reinforcement learning [7]. Population-based algorithms are the most commonly methods used to plan path for the UAV, such as genetic algorithm (GA) [8,9], particle swarm optimization (PSO) [10], and ant colony algorithm (ACO) [11,12]. This kind of algorithms

This work was supported by the National Natural Science Foundation of China (61973285, 62076226), Hubei Provincial Natural Science Foundation of China (2022CFB438), the Opening Fund of Key Laboratory of Geological Survey and Evaluation of Ministry of Education (Grant No. GLAB 2023ZR08).

L. Fang et al. (Eds.): CICAI 2023, LNAI 14474, pp. 271–282, 2024.
https://doi.org/10.1007/978-981-99-9119-8_25

has the advantages of fast convergence speed, good parallelism and facilitate collaboration among multiple populations.

Compared to other algorithms, PSO is often utilized in the path planning of UAV. However, conventional PSO is confronted with several challenges. Firstly, the initialization process of the particles is complex, leading to suboptimal initial paths. Secondly, these algorithms exhibit slow convergence rates and are prone to getting trapped in local optima. In order to solve the above problems, we propose an EPSO-GA, and furthermore adopted it on the UAV path planning. The main contribution of this paper can be summarized as follows.

(1) A hybrid approach of Q-learning and random initial solutions is applied to find the initial paths for the UAV, which improves the quality of initial paths and accelerates the convergence of the EPSO-GA.

(2) The acceleration coefficients of EPSO-GA are designed as adaptive ones by the fitness value to make full use of all particles and strengthen the global search ability of the algorithm. In addition, a heuristics factor is recommended into mutation to speed up the convergence of the algorithm.

2 The Proposed Method

In this section, we construct the objective function for UAV path planning. To enhance the quality of the initial population in the particle swarm algorithm, we introduce Q-learning for hybrid initialization. Furthermore, we accelerate the convergence speed and improve the global search capability of the algorithm by incorporating mutation and crossover mechanisms.

2.1 Objective Function Construction

In this article, path length is defined as follows:

$$f_L = \sum_{i=1}^{n} \sqrt{\left(x_{i+1} - x_i\right)^2 + \left(y_{i+1} - y_i\right)^2} \tag{1}$$

where x_i and y_i are the coordinates of path point i, and n is the total number of path points.

To simplify the calculation, the path from the starting point to the destination is divided into three segments. For each segment, the 1/4, 2/4 and 3/4 points of the outbound leg are used as reference points. The formula for calculating the threat intensity of the jth radar against the drone is as follows:

$$T_j = \sum_{m=1}^{3} K \left(\frac{1}{d^4_{1/4,m,j}} + \frac{1}{d^4_{2/4,m,j}} + \frac{1}{d^4_{3/4,m,j}} \right) \tag{2}$$

where m represents the path segment, K is the threat intensity coefficient, $\frac{1}{d_{1/4,m,j}}$ represents the distance between the jth radar and a quarter of the m path segment, and T_j represents the threat intensity of the jth radar against the drone. Therefore, the threat intensity of a single drone is described as follows:

$$f_T = \sum_{j=1}^{N} T_j \tag{3}$$

where N is the number of radars.

In order to reduce computational complexity, the path planning problem for UAV is transformed into a constrained multi-objective function. By assigning weights to each objective, transforming the multi-objective problem into a single-objective problem. The objective function can be represented as follows:

$$f_{obj} = w_1 * f_L + w_2 * f_T \tag{4}$$

where w_1 and w_2 are the weights for path length and threat intensity respectively.

2.2 Initialization Method of PSO Particle Swarm Based on Q-Learning

The hybrid initialization method of Q-learning and random initial population is used in this paper to obtain the initial swarm of particles. In order to ensure that the paths generated by Q-learning have lower threat intensity and shorter path length compared to those generated randomly, the reward function of Q-learning is shown in Table 1.

Table 1. Design of reward function.

State of UAV	Reward
Starting point	0
End position	20
Obstacles and radar	-1
Feasible point	r

In Q-learning, all feasible points, obstacles and radars are distinguished in the form of positive and negative values. Each cell represents a different value, where 0 represents the starting point, r represents free space, -1 represents an obstacle or radar occupied cell, and 20 represents the destination. The reward values of all feasible points are adaptively adjusted based on their positions, where r¿0. The reward value of feasible points increases as they approach the destination and decreases as they approach a threat. The reward values of all feasible points are shown below:

$$\begin{aligned} r = r_0 + r_{\max} * \exp & \left[\frac{-\sqrt{\left(x_u - x_{\mathrm{g}}\right)^2 + \left(y_u - y_{\mathrm{g}}\right)^2}}{k_{\mathrm{g}}} \right] \\ -r_{\mathrm{t}} * \exp & \left[\frac{-\sum_{i=1}^{m} \sqrt{\left(x_u - x_{\mathrm{it}}\right)^2 + \left(y_u - y_{\mathrm{it}}\right)^2}}{k_{\mathrm{t}}} \right] \end{aligned} \tag{5}$$

where r_0 represents the basic reward value, $r_{\max}$ represents the maximum reward value without considering threats, r_t is the threat coefficient, (x_u, y_u) and (x_g, y_g) are the coordinates of the UAV and the target point respectively, (x_{it}, y_{it}) is the coordinates of the ith radar, k_g and k_t are the control coefficients of the reward function and m is the number of threats.

The balance between exploration and exploitation is another key part in Q-learning. This paper adopts Boltzmann distribution [13], and its expression is as follows:

$$p(a \mid s) = \frac{e^{Q(s,a)/T}}{\sum_{a_i \in A} e^{Q(s,a_i)/T}} \tag{6}$$

$$T = \lambda^k T_0 \tag{7}$$

where $p(a \mid s)$ represents the probability that action a is selected in state s. λ is a constant satisfying $0 < \lambda < 1$, k is the current iteration number and T is a control parameter. At the beginning of the training process, T has a large value to ensure strong exploration capability, which decreases as the number of iterations increases to ensure that the algorithm focuses on exploitation. The update formula for Q values is as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max Q\left(s', a'\right) - Q(s,a)\right] \tag{8}$$

where α is the learning rate, γ is the discount factor, r is the immediate reward value of the current action and $Q\left(s,a\right)$ is the estimated value of taking the current action.

Using the Q-learning method based on the fitness values of all solutions, m paths are generated, and n paths are obtained by random initialization. The initial population size $S = m + n$.

2.3 PSO Mutation Crossover Strategy

Inspired by the way of multiple UAV task assignment in [14], the updating strategy of EPSO-GA can be defined as follows:

$$x_{i,j}(t+1) = c_2 \bullet f_3\left(c_1 \bullet f_2\left(w \bullet f_1\left(x_{i,j}(t), p_{i,j,\text{ best}}(t)\right), g_{i,j,\text{ best}}(t)\right)\right) \tag{9}$$

where $x_{i,j}(t)$, $p_{i,j,best}(t)$ represent the position and personal optimal value of particle in t iterations respectively, while $g_{i,j,\text{ best}}\left(t\right)$ represents the global best value of the particle in the t-th generation. w is the learning factor of the particle with respect to itself, c_1 and c_2 are the learning factors of the current particle with respect to its individual best value and global best particle, respectively. f_1 is the operation of the particle with respect to itself, while f_2 and f_3 are the operations of the particle based on its individual best value and global best value, respectively.

f_1 is defined as a mutation where the mutation probability of the particle i is w, f_1 is formulated as follows:

$$f_1 = w \bullet f_1 \left(x_{i,j}(t)\right) \tag{10}$$

where w is the inertia weight which has a great influence on performance of the algorithm. In order to accelerate the convergence of the algorithm, we adopt the linear time-varying inertia weight updating strategy in [15], which is expressed in the following formula:

$$w(t) = \frac{T-t}{T}\left(w_{\max} - w_{\min}\right) + w_{\min} \tag{11}$$

where the value of w is linearly decreased from $w_{\max}$ to the final value $w_{\min}$. t is the current iteration of the algorithm and maxiter is the maximum iterations of the algorithm. At the beginning of the iteration, the particle exhibits a strong global search ability, while towards the end of the iteration, it acquires a local search capability.

As shown in the Fig. 1(a), the mutation operation begins by randomly selecting a point in the path (excluding the start and end points). Assuming the previous coordinate is N, there are 8 possible choices for the mutated coordinate. To improve the efficiency of search, the distance D_i between each candidate i and the destination is calculated, and the probability P_i of the i being selected is calculated as follows:

$$P_i = k_i * \frac{1/D_i}{\sum_{i=1}^{8} 1/D_i} \tag{12}$$

where k_i is the adjusted adaptively according to the angle between coordinate i and the destination. In particular, the probability is 0 when i represents an obstacle.

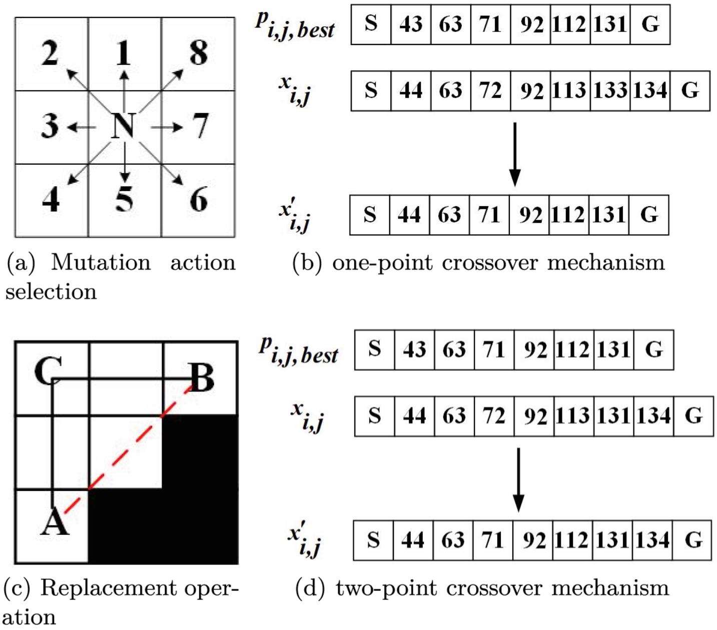
To address the coordinate transformation issue, this paper introduces the crossover mechanism from genetic algorithms while retaining the individual and global learning strategies of the particle swarm.

f_2 is defined as the intersection between particle $x_{i,j}$ and the individual best particle with a probability of c_1. The formula for f_2 is as follows:

$$f_2 = c_1 \bullet f_2 \left(f_1, p_{i,j,\text{ best }}(t)\right) \tag{13}$$

f_3 is defined as the intersection between particle $x_{i,j}$ and the global best particle with a probability of c_2. The formula for f_3 is as follows:

$$f_3 = c_2 \bullet f_3 \left(f_2, g_{i,j,best}(t)\right) \tag{14}$$



(a) Mutation action selection

(b) one-point crossover mechanism

(c) Replacement operation

(d) two-point crossover mechanism

Fig. 1. Crossover and mutation mechanisms.

If there is an overlap between the path points of the current particle and either the individual best particle or the global best particle (excluding the start and end points), a crossover operation is performed. In Fig. 1(b), particle x shares two crossover points (63, 92) with the individual best particle. This operation involves swapping the path point 71 between the two crossover points and swapping the subsequent crossover point with the path points (112, 131) between the individual best particle and particle x's target point. In Fig. 1(d), particle x shares more than two crossover points (63, 92, 131) with the individual best particle. In this case, only the path points 71 and 112 between the single best particle and particle x's crossover point are swapped.

In EPSO-GA, the values of c_1 and c_2 are updated based on their fitness values to fully utilize all particles and enhance the algorithm's ability to escape local optima. In each iteration, all particles are sorted based on their fitness values, with the top half being stored in set A and the bottom half in set B. In the next iteration, particles in set A have a higher probability of crossover. The formulas for updating the learning factors c_1 and c_2 are as follows:

$$\eta = \frac{min\_\text{fitness}}{max\_\text{fitness}} - \frac{min\_\text{fitness}}{\text{fitness}\ (x_{ij})} \tag{15}$$

$$c_1 = a + (1-a)e^{\eta} \tag{16}$$

$$c_2 = b + (1-b)e^{\eta} \tag{17}$$

In the formulas above, $max_$fitness and $min_$fitness represent the maximum and minimum fitness values of all particles in the population, respectively. fitness (x_{ij}) represents the fitness value of particle j after i iterations, and the values of a and b are experimentally determined constants. For particles in set A, higher fitness values result in higher crossover probabilities. The particle with the highest fitness value will always undergo crossover, while the probabilities of the other particles decrease exponentially with their fitness values, but still satisfy $c_1 > a$, $c_2 > b$. Through these operations, the top half of the particles with higher fitness values are given an increased probability of crossover, allowing them to focus more on searching, improving the algorithm's global search capability, and helping to escape local optima.

For the bottom half of the particles with lower fitness values, the values of c_1 and c_2 are designed as follows:

$$c_1 = c_{\max} - \frac{c_{\max} - c_{\min}}{max\_iter} * t \tag{18}$$

$$c_2 = c_{\min} + \frac{c_{\max} - c_{\min}}{max\_iter} * t \tag{19}$$

Here, both $c_{\max}$ and $c_{\min}$ are set to $\left(c_{\max} + c_{\min}\right)/2 = c_1^c = c_2^c$, where c is a constant satisfying $1 > c_{\max} > c_{\min}$. max_iter represents the maximum number of iterations, and t represents the current iteration number. As the number of iterations increases, the value of c_1 gradually increases while the value of c_2 gradually decreases, causing the algorithm to shift from global search to local search.

Additionally, this article simplifies the paths by performing simplification operations when there are no obstacles or threat points blocking a straight line between two non-contiguous positions. As shown in Fig. 1(c), if the UAV can fly directly from the current point to another waypoint in a straight line, the path A-C-B will be replaced by the simpler path of A-B.

3 Simulation

To demonstrate the performance of proposed method, simulations and comparisons are carried out in 2D static environment. They are implemented in the MATLAB environment and compared performance with CIPSO [16] and CIGA [17], the simulation is running on a platform with a 3.2 GHz CPU and 8.0 GB of RAM.

3.1 Parameters Setting

The weight of objectives are set as $w_1 = 0.8$, $w_2 = 0.2$, the size of population is set as $S = 50$ and $max\_iter = 100$. The parameters of the EPSO-GA algorithm are set as $w_{max} = 0.8$, $w_{min} = 0.2$, $c_{max} = a = b = 0.8$, $c_{min} = 0.6$. The parameters of the CIPSO algorithm are set as $w_{max} = 0.9$, $w_{min} = 0.4$, $c_{max} = 3.5$, $c_{min} = 0.5$, $V_1 = 0.5$, $V_2 = 0.1$, $a = 2$, $\mu = 4$. The parameters of the CIGA algorithm are set as $c_r = 0.8$, $pc = 0.15$. Coordinates of starting position, destination and radars in the case are listed in Table 2.

Table 2. The starting position and the destination of UAV and the coordinates of the radars.

Environment type	Start position	End position	Radar1	Radar2	Radar3
Z	381	20	55	356	×
Complex	1	400	131	217	309

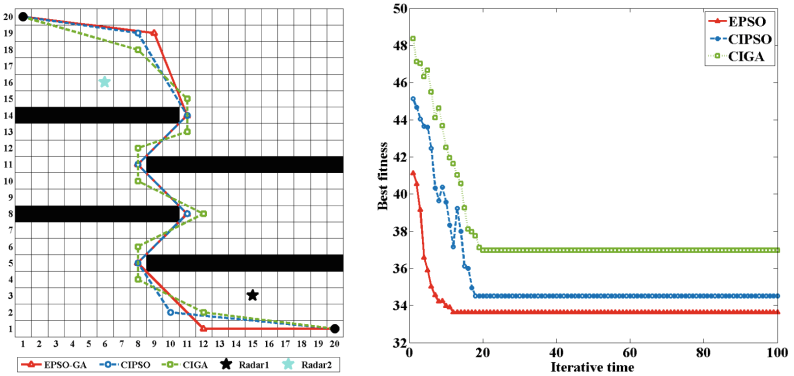
3.2 Result of UAV Path Planning

To compare the effect of the proportion of paths obtained by the Q-learning algorithm on the optimization of the algorithm, the number of paths m obtained by the Q-learning and the number of paths n obtained by random initialization were adjusted respectively, and five groups of experiments were conducted in the complex environment, with each group of experiments running independently for 20 times. The experimental results are shown in Table 3. Where $m = 0$ and $n = 50$ means purely random initialization, the initial best fitness is reduce with the proportion of m increase, and the convergence result of the algorithm is the best when $m = 5$ and $n = 45$. However, the average iteration number increases as m increases when $m > 5$. Therefore, compared with the random initialization, the path obtained by introducing Q-learning reduces the initial best fitness value of the population, which can accelerate the convergence speed of the algorithm. However, if the proportion of m is too large, the algorithm may fall into local optimal and the number of iterations of the algorithm will be increased.

Table 3. Results of EPSO-GA under different m and n values.

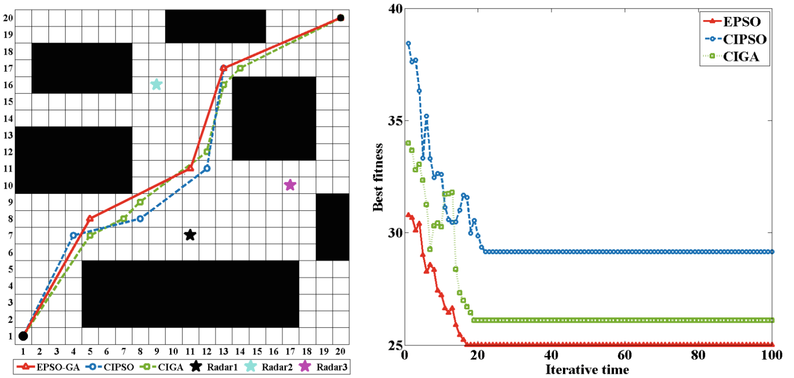
Indicator	$m = 0$,$n = 50$	$m = 3$,$n = 47$	$m = 5$,$n = 45$	$m = 10$,$n = 40$	$m = 15$,$n = 35$
Initial best fitness	33.7546	32.2565	30.7846	30.0512	**28.3654**
Average running time	16.21	15.84	**15.36**	15.38	16.02
Average iterations	21	18	**17**	20	26

The comparative simulations among different algorithms are carried out in two different environments, including Z-type and a complex environment, each method is repeated 50 times independently and the best results are chosen. The generated paths and corresponding convergence curves under different environments are presented in Fig. 2 and Fig. 3. The statistical results of simulation are shown in Table 4. As observed in Fig. 2(a) and Fig. 3(a), CIGA, CIPSO and our proposed method are complete the mission from the staring position to the destination without any collision with obstacles and radars. In the Z-type environment, the path generated by CIGA has a longest path and closest to the radars which the value of path length and threat intensity are 42.1637 and 17.0397 respectively, while CIPSO has a shortest path with 39.2854 and a second highest threat intensity with 16.6979. Our method is slightly longer than CIPSO where the path length is 39.7274, but maintain a relative longer distance



(a) Generated path under Z-type environment. (b) Best fitness curves of different algorithms.

Fig. 2. Simulation results under Z-type environment.



(a) Generated path under complex environment. (b) Best fitness curves of different algorithms.

Fig. 3. Simulation results under complex environment.

from radars which threat intensity is only 9.5938. In the complex environment, our method complete the mission with the shortest path length and the lowest threat intensity.

In the Fig. 2(b) and Fig. 3(b), the best fitness of EPSO-GA are obviously smaller than that of the other algorithms at the initial stage, which are the results of hybrid initialization. Besides, EPSO-GA holds the fastest convergence compare with the other methods, while reach its optimal and remain stable in 12th and 17th iteration under two different environments respectively.

Table 4. Results comparison between the algorithms under different environments.

Indicator	Z-type			Complex type		
	EPSO-GA	CIPSO	CIGA	EPSO-GA	CIPSO	CIGA
Shortest path length	39.7274	**39.2854**	42.1637	**28.1069**	29.5298	28.7108
Threat intensity	**9.5938**	16.6979	17.0397	**13.4810**	27.8362	16.0080
Success rate (%)	**98**	96	98	**100**	98	100
Average running time (s)	6.24	6.87	**5.83**	15.36	18.91	**14.46**

A successful search is defined as finding the optimal solution after 200 iterations. As shown in Table 4, the EPSO-GA algorithm we proposed ranks first in the success rate. In terms of threat intensity, EPSO-GA are 43.7%, 42.5% better than CIGA and CIPSO in the Z-type environment, 15.8%, 51.6% in the complex environment respectively. Although our proposed method consumes slightly more time than the CIGA which due to the adaptive adjustment of parameters and hybrid initialization, the optimality of the solutions we have obtained. Our proposed EPSO-GA can effectively generate optimal path and is more practical in off-line path planing.

Table 5. Results comparison based on different weights under complex environment.

Weights	Algorithms	Best fitness	Average fitnes	Average iterations
$w_1 = 0.9$, $w_2 = 0.4$	EPSO-GA	**26.3028**	**27.0015**	**17**
	CIPSO	30.1475	30.7854	20
	CIGA	29.1311	30.1227	18
$w_1 = 0.8$, $w_2 = 0.2$	EPSO-GA	**25.1817**	**25.4910**	**17**
	CIPSO	29.1911	32.1429	21
	CIGA	26.1702	27.8914	19
$w_1 = 0.6$, $w_2 = 0.4$	EPSO-GA	27.4030	**28.0321**	**19**
	CIPSO	28.9213	29.1231	23
	CIGA	**26.5410**	28.2356	19
$w_1 = 0.5$, $w_2 = 0.5$	EPSO-GA	**25.3211**	**25.8940**	21
	CIPSO	29.1010	29.7414	21
	CIGA	26.3007	26.5641	**19**

In order to compare the influence of the weight of path length and threat intensity on the optimization process, four groups of experiments are conducted in complex environment by adjusting the weights of two sub-objectives, and each group of experiments is run independently for 20 times. The experimental results are shown in Table 5. It can be seen that the average iteration number of each algorithm is the least, the best fitness and the average fitness of ESPO-GA algorithm are the minimum, and the obtained path quality is the best when $w_1 = 0.8$, $w_2 = 0.2$.

4 Conclusion

In this paper, a hybridization of EPSO-GA algorithm is proposed for path planning of UAV in 2D static environment. Q-learning is utilized to initialize paths of UAV which improves the quality of initial paths and accelerates the convergence of the EPSO-GA. The acceleration coefficients of EPSO-GA are designed as adaptive ones by the fitness value to make full use of all particles and strengthen the global search ability of the algorithm, and a heuristics factor is introduced into mutation to speed up the convergence of algorithm. Through experiments in two different environments, it has been shown that our proposed algorithm has the advantages of path security and fast convergence in the path planning of single UAV. In the future work, we will concentrate on reducing time consumption of path planning for UAV and 3D environment will be introduced with dynamic obstacles.

References

1. Gao, X., Zhu, X., Zhai, L.: Minimization of aerial cost and mission completion time in multi-UAV-enabled iot networks. IEEE Trans. Commun. (2023)
2. Liu, Y., Chen, B., Zhang, X., Li, R.: Research on the dynamic path planning of manipulators based on a grid-local probability road map method. IEEE Access **9**, 101186–101196 (2021)
3. Li, Z., You, K., Sun, J., Song, S.: Fast trajectory planning for dubins vehicles under cumulative probability of radar detection. Signal Process. **210**, 109085 (2023)
4. Wang, J., Meng, M.Q.H., Khatib, O.: Eb-rrt: optimal motion planning for mobile robots. IEEE Trans. Autom. Sci. Eng. **17**(4), 2063–2073 (2020)
5. Chen, Y., Bai, G., Zhan, Y., Hu, X., Liu, J.: Path planning and obstacle avoiding of the USV based on improved ACO-APF hybrid algorithm with adaptive early-warning. IEEE Access **9**, 40728–40742 (2021)
6. Sun, J., Tang, J., Lao, S.: Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm. IEEE Access **5**, 18382–18390 (2017)
7. Guo, X., Peng, G., Meng, Y.: A modified Q-learning algorithm for robot path planning in a digital twin assembly system. Int. J. Adv. Manuf. Technol. **119**, 3951–3961 (2022)
8. Yuan, J., Liu, Z., Lian, Y., Chen, L., An, Q., Wang, L., Ma, B.: Global optimization of UAV area coverage path planning based on good point set and genetic algorithm. Aerospace **9**(2), 86 (2022)
9. Lee, J., Kim, D.W.: An effective initialization method for genetic algorithm-based robot path planning using a directed acyclic graph. Inf. Sci. **332**, 1–18 (2016)
10. Mesquita, R., Gaspar, P.D.: A novel path planning optimization algorithm based on particle swarm optimization for uavs for bird monitoring and repelling. Processes **10**(1), 62 (2021)
11. Zeng, M.R., Xi, L., Xiao, A.M.: The free step length ant colony algorithm in mobile robot path planning. Adv. Robot. **30**(23), 1509–1514 (2016)

12. Stodola, P.: Hybrid ant colony optimization algorithm applied to the multi-depot vehicle routing problem. Nat. Comput. **19**(2), 463–475 (2020)
13. Das, P., Jena, P.K.: Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators. Appl. Soft Comput. **92**, 106312 (2020)
14. Yin, G., Zhou, S., Mo, J., Cao, M., Kang, Y.: Multiple task assignment for cooperating unmanned aerial vehicles using multi-objective particle swarm optimization. Comput. Modernization **8**, 7–11 (2016)
15. Tian, D., Shi, Z.: Mpso: Modified particle swarm optimization and its applications. Swarm Evol. Comput. **41**, 49–68 (2018)
16. Shao, S., Peng, Y., He, C., Du, Y.: Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. ISA Trans. **97**, 415–430 (2020)
17. Qu, H., Xing, K., Alexander, T.: An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. Neurocomputing **120**, 509–517 (2013)