IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 3, 2020, accepted March 21, 2020, date of publication March 30, 2020, date of current version April 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984236

A Novel Route Planning Method of Fixed-Wing Unmanned Aerial Vehicle Based on Improved QPSO

CHEN HUANG[1,2], JIYOU FEI[2], AND WU DENG[3]
[1]College of Civil Aviation, Shenyang Aerospace University, Shenyang 110136, China
[2]College of Vehicle Engineering, Dalian Jiaotong University, Dalian 116028, China
[3]College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China

Corresponding author: Jiyou Fei (fjy@djtu.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61771087, and in part by the Doctor Research Startup Foundation of Shenyang Aerospace University under Grant 18YB36.

ABSTRACT This paper proposes a quick and accurate method based on an improved quantum-behaved particle swarm optimization (QPSO) algorithm for route planning of fixed-wing unmanned aerial vehicle (UAV). To overcome the deficiencies of local optima and slow global convergence speed, a novel strategy of particle dimension search is proposed in the QPSO algorithm. It is implemented by transforming original evaluation function into evaluation function of waypoint to more easily escape from local optima and accelerate global convergence speed. In addition, an efficient pretreatment technology for the initial trajectory is set to shorten the calculation time of route planning. Compared with other representative route planners, the comparison results indicate that the proposed route planner is more effective and feasible, which can take on faster convergence speed and better global search ability. The proposed route planner can provide a valuable reference for the route planning of fixed-wing UAVs in different environments.

INDEX TERMS Unmanned aerial vehicle, route planning, QPSO, particle dimension search.

I. INTRODUCTION

UAV systems with the great advantages of low-cost and flexibility have been applied widely in military and nonmilitary tasks to implement searching, patrolling, target tracking, and surveillance in complex indoor/outdoor environments [1]–[4]. In order to enhance the automation level of UAV systems, it is necessary to carry out various technologies, such as the environment detection, path planning, and the design of flight control systems [5]. As a sub-module of UAV control modules, path planner plays an important role in UAV systems. The key task for route planning is to find a flyable route from the starting point to destination avoiding all the collisions and threats on the basis a series of performance measure, e.g. path length, radars/missiles threats and flight height [6], [7]. On the one hand, route planning can be regarded as a multiple-objective optimization problem. On the other hand, the path planning problem is more complex due to many constraints, e.g. complex environment constraints and UAV physical constraints etc. Therefore, it is a complicated optimization problem with multi-objective and multi-constraint in essence.

Offline planner mainly deals with the path planning problem when the global information is known in advance [8], which has been more deeply and extensively studied in recent years. At present, there are three classes of approaches to solve offline path planning problem. The first type is based on graph algorithms, such as Voronoi diagram [9] and Visibility Graph approach [10], [11]. However, the kinematic and dynamic constraints of UAV are unable to be taken into account based on graph algorithms. The second type is based on heuristic search algorithms. A* algorithm is a typical heuristic representative, which generates the trajectory with the guide of the least-cost from a given initial node to a target node [12]. As a result, when the mission space enlarges, the calculation amount of finding the optimal route will increase explosively [13]. The last type is based on evolutionary computation algorithms [6], [14], [15]. Multiple routes are randomly initialized and the corresponding objective function values are calculated according to given evaluation criteria. Each solution is updated at the beginning of each loop. The general UAV planning system

The associate editor coordinating the review of this manuscript and approving it for publication was Sunil Karamchandani.

based on evolutionary computation algorithms is shown in figure 1.

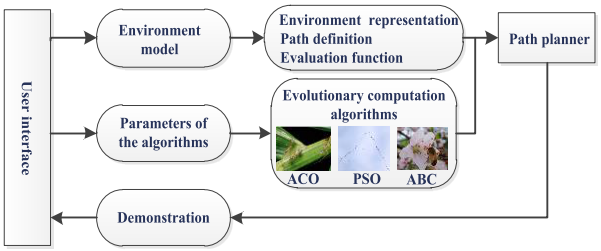


FIGURE 1. General planning system based on evolutionary computation algorithms.

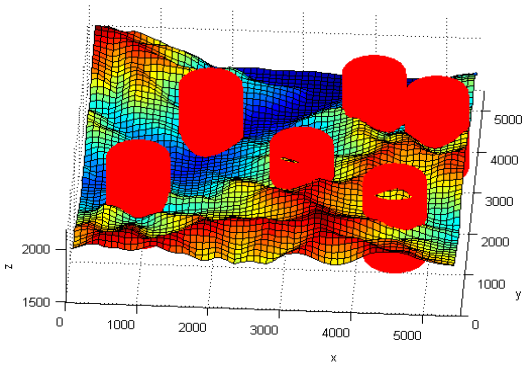


FIGURE 2. The environment representation.

Due to the satisfactory performance of evolutionary computation algorithms, they are more flexible and effective than most of other existing approaches for UAV route planning. However, it is found that the evolutionary computation algorithms still have some drawbacks (e.g. local optima and slow global convergence speed). In order to overcome these shortcomings in planning problem, Pehlivanoglu [16] proposed multi-frequency vibrational genetic algorithm (MVGA) with a new mutation application strategy and diversity variety to shorten the computational time. Özalp and Sahingoz [17] developed genetic algorithm and parallel approach to reduce path planning calculations under 3D dimensional structure. Mac *et al.* [18] proposed a multi-objective PSO with an accelerated update methodology based on Pareto dominance principle to enhance computational efficiency. Zhang *et al.* [19] proposed a hybrid multi-objective bare-bones PSO with differential evolution to improve the feasibility of an infeasible path.

Particle swarm optimization (PSO) algorithm is a popular evolutionary computation algorithm, which has relatively less adjustment parameters and better optimization effects. It has been successfully introduced into many fields to solve NP-hard optimization problems [20]–[22]. In PSO algorithm, the motion state of each particle is described by position and speed, and its position is updated in the process of evolution [23]. Due to the constraint of the velocity of the particle, the search of the feasible solution is difficult to cover the entire mission space. QPSO algorithm is proposed to improve the randomness and global search performance of the particles [24]. The principle of QPSO algorithm is derived from quantum mechanics and PSO model, which ensures the global convergence of the algorithm. Similar to other evolutionary algorithms, the slow convergence speed and the local optimum entrapment still exist inevitably [25]. To solve these problems, Fu *et al.* [26] combined the DE algorithm with the QPSO algorithm in an attempt to further enhance the performance of QPSO algorithm. Tokgo and Li [27] proposed a sorted QPSO algorithm and the group with best fitness is replaced by the group of the particles with worst fitness. In order to enhance the convergence speed and optimization precision, Xue *et al.* [28] proposed a hybrid improved quantum-behaved PSO (LTQPSO) for trajectory planning of autonomous mobile robot (AMR) in the environment with random obstacles. Rehman *et al.* [29] proposed a novel fitness selection methodology and a dynamic parameter update strategy to avoid trapping into local optima. While QPSO algorithm is adopted to solve multi-dimensional optimization problem in a complex three-dimension environment, all the waypoints of a route are regarded as an individual to participate into the selection process of QPSO algorithm. In this way, some presented high quality waypoints in previous iteration may be discarded owing to the effect of other low quality waypoints. As a result, it is difficult that the high-quality waypoints enter into the further evolution process. Therefore, a novel algorithm based on particle dimension search, denoted as SDQPSO, is presented to improve the search accuracy and convergence speed of QPSO for route planning of fixed-wing UAVs.

The remainder of the paper is organized as follows. Section II specifies the description of flight route and planned space environment. Section III introduces the route evaluation function in detail. Section IV presents the improved QPSO. In Section V, the effectiveness of the proposed planner is tested by comparing simulation results with other representative route planners in real terrain elevation maps. Finally, the conclusions are provided in Section VI.

II. ENVIRONMENT AND FLIGHT ROUTE DESCRIPTION

For the planning problem of the flight route, some key elements should be considered in advance, e.g. environment description of task space, flight trajectory etc. At first, the task space is decomposed into a series of three-dimensional mesh, and the corresponding 2D matrix representation of 3D environment (figure 2) is shown in figure 3. In digital elevation map, the danger zones from the radars or missiles are equivalent to some standard cylinders in order to simplify the models of the threats [6]. So the threat zones can be described by a separate matrix as follows:

$$Z_{\text{threat}} = \begin{pmatrix} x_1 & y_1 & r_1 \\ x_2 & y_2 & r_2 \\ \cdots & \cdots & \cdots \\ x_n & y_n & r_n \end{pmatrix} \tag{1}$$

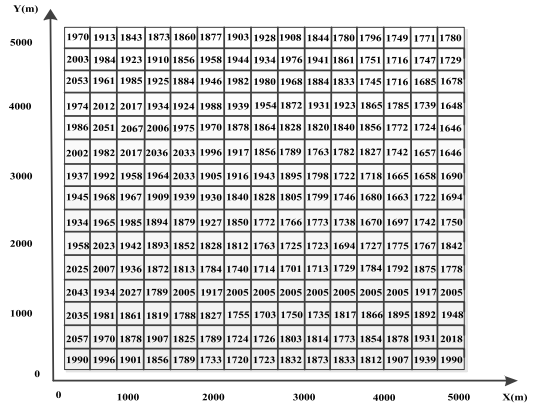


FIGURE 3. The corresponding matrix of the terrain in Figure 2.

where x_i and y_i are the coordinates of the bottom center point of the i-th threat source, r_i is the radius of the i-th cylinder.

The flight path for UAV is composed of many waypoints in three-dimension space. In figure 4, suppose the flight task is from the start point S to the target point T, and $W_1, W_2, \cdots, W_{N_w}$ are the waypoints of the flight route. Then a flight path is denoted as $\boldsymbol{P} = \{S, W_1, W_2, \cdots, W_{N_w}, T\}$.

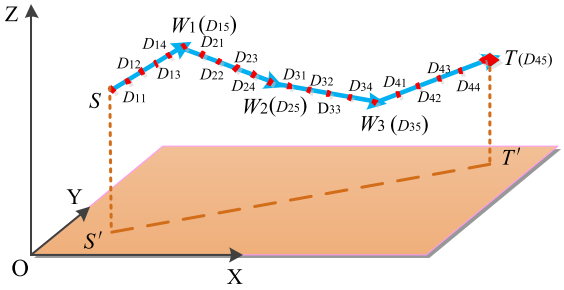


FIGURE 4. The flight path in 3D space.

The segment between the start point S and the first waypoint W_1 is divided into N_d subsegments equally, which are labeled with $D_{11}, D_{12}, \cdots, D_{1N_d}$. Figure 4 shows a flight path including multiple waypoints and division points in three-dimension space.

Define $D_{ik} = (x_{ik}, y_{ik}, z_{ik})$ as the k-th dividing point in the segment $\overrightarrow{W_{i-1}W_i}$ between the $(i-1)$-th and the i-th waypoint, the coordinates of the division point can be calculated as follows:

$$\begin{cases} x_{ik} = x_{i-1} + k \cdot (x_i - x_{i-1})/N_d \\ y_{ik} = y_{i-1} + k \cdot (y_i - y_{i-1})/N_d \\ z_{ik} = z_{i-1} + k \cdot (z_i - z_{i-1})/N_d \end{cases} \tag{2}$$

where $i = 1, 2, \cdots, N_w + 1$, $k = 1, 2, \cdots, N_d$.

III. EVALUATION FUNCTION OF FLIGHT ROUTE

The performance of each candidate route is often associated with the evaluation function $F_{evaluate}$. The general representation of route evaluation function is defined as follows:

$$F_{evaluate}(P) = \sum_{k=1}^{3} J_k(P) + \sum_{v=1}^{3} C_v(P) \tag{3}$$

where the first part $\sum_{k=1}^{3} J_k(P)$ is the objective function of the flight route, the other $\sum_{v=1}^{3} C_v(P)$ is the constraint of the route. The evaluation function of the flight route is presented in the following subsection in detail.

A. OBJECTIVE FUNCTION OF FLIGHT ROUTE

The objective function of flight route is defined as follows:

$$\sum_{k=1}^{3} J_k(P) = J_{length}(P) + J_{altitude}(P) + J_{threat}(P) \tag{4}$$

where $J_{length}(\cdot)$ is the path length cost, $J_{altitude}(\cdot)$ is the flight altitude cost and $J_{threat}(\cdot)$ is the threat cost, which is mainly used to penalize the routes through threat zones.

(1) Minimal Route Length

Route length is measured by path length ratio (PLR), hence the term J_{length} is defined as

$$J_{length}(P) = \frac{L_{traj}}{L_{ST}} = \frac{\sum_{i=1}^{N_w+1} ||\overrightarrow{W_{i-1}W_i}||}{L_{ST}} \tag{5}$$

where L_{ST} denotes the length of the straight line connecting the start point with the target, L_{traj} denotes the actual trajectory length of the UAV route, $||\cdot||$ is the Euclidean distance of a vector, when $i = 1$,W_{i-1} means the start point, when $i = N_w+1$, W_{i+1} means the target. In general, the path length L_{traj} is expected to be $(1\sim1.5)L_{ST}$ for a feasible route [20], as

$$J_{length} \in [1, 1.5] \tag{6}$$

(2) Minimal Flight Altitude

When the UAV flies at low altitude, UAV can avoid from the danger being detected or attacked by the unknown radars or surface-to-air missiles due to the protection of the terrain. The term $J_{altitude}$ is calculated as

$$J_{altitude}(P) = \frac{A_{route}(P) - Z_{min}}{Z_{max} - Z_{min}} \tag{7}$$

where A_{route} is the average flight altitude of the actual route, Z_{min} and Z_{max} represent the lower altitude limit and upper altitude limit in planning space, respectively. Then

$$J_{altitude} \in [0, 1] \tag{8}$$

(3) Minimal Threat Risk

The threat function J_{threat} is introduced to reduce the risk of UAV flying into the scope of the hostile defending radar or surface-to-air missile (SAM), hence the term J_{threat} is defined as

$$J_{threat}(P) = \frac{\sum L_{inside}}{\sum_{k=1}^{N_T} D_k} \tag{9}$$

where L_{inside} is the total length of the subsections of the actual flight route going through any threat zones, D_k is the diameter of the k-th threat source, N_{T} is the total number of threat sources located in the mission space. J_{threat} is required to be in the range of [0, 1].

B. CONSTRAIN FUNCTION OF FLIGHT ROUTE

To ensure the feasibility of route, a series of constraints must be satisfied firstly, e.g. the turning angle, the slope angle, and the flight height. The constraint function C is defined as

$$\sum_{v=1}^{3} C_v(P) = C_{\text{collision}}(P) + C_{\text{turning}}(P) + C_{\text{slope}}(P) \quad (10)$$

where $C_{\text{collision}}(\cdot)$ is used to penalize the routes that collide with the ground, $C_{\text{turning}}(\cdot)$ penalizes the routes requiring larger turning angle than the maximum available turning angle, and $C_{\text{slope}}(\cdot)$ penalizes the routes requiring slope angle beyond the given slope angle. The constraint function of a flight route is depicted in detail as follows:

(1) Terrain Height Constraint

The term $C_{\text{collision}}(P)$ associated with terrain collisions is calculated by

$$C_{\text{collision}}(P) = \begin{cases} 0, & L_{\text{under}} = 0 \\ P_{\text{en}} + \frac{L_{\text{under}}}{L_{\text{traj}}}, & L_{\text{under}} > 0 \end{cases} \quad (11)$$

where L_{under} is the sum of the length of the flight trajectory below the altitude of the terrain, L_{traj} is the total length of the actual route and P_{en} is penalty constant in order to separate non-feasible solutions from all the candidate routes. For the evaluation function, the constant P_{en} should be more than the fitness value of the worst feasible path. Therefore, it is set to 3.5.

(2) Turning Angle Constraint

The turning angle constraint associated with the dynamic characteristic of UAV is calculated as

$$C_{\text{turning}}(P) = \begin{cases} 0, & \text{if } \sum_{i=1}^{N_{\text{w}}} a1_i = 0 \\ P_{\text{en}} + \frac{\sum_{i=1}^{N_{\text{w}}} a1_i}{N_{\text{w}}}, & \text{if } \sum_{i=1}^{N_{\text{w}}} a1_i > 0 \end{cases}$$
$$\text{with } a1_i = \begin{cases} 0, & \text{if } \theta_i < \theta_{\max} \\ 1, & \text{if } \theta_i > \theta_{\max} \end{cases} \quad (12)$$

where $\theta_{\max}$ is the given maximum turning angle. The turning angle θ_i is depicted as

$$\theta_i = \arccos\left(\frac{(x_{i+1} - x_i, y_{i+1} - y_i) \cdot (x_i - x_{i-1}, y_i - y_{i-1})^T}{\| (x_{i+1} - x_i, y_{i+1} - y_i) \cdot (x_i - x_{i-1}, y_i - y_{i-1}) \|}\right) \quad (13)$$

(3) Slope Angle Constraint

The slope angle constraint is calculated as

$$C_{\text{slope}}(P) = \begin{cases} 0, & \text{if } \sum_{i=1}^{N_{\text{w}}+1} a2_i = 0 \\ P_{\text{en}} + \frac{\sum_{i=1}^{N_{\text{w}}+1} a2_i}{N_{\text{w}}+1}, & \text{if } \sum_{i=1}^{N_{\text{w}}+1} a2_i > 0 \end{cases}$$
$$\text{with } a2_i = \begin{cases} 0, & \text{if } \varphi_{\min} < \varphi_i < \varphi_{\max} \\ 1, & \text{else} \end{cases} \quad (14)$$

where $\varphi_{\min}$ and $\varphi_{\max}$ are the lower and upper bounds of the slope angle, respectively. The slope angle φ_i is calculated as

$$\varphi_i = \arctan\left(\frac{z_i - z_{i-1}}{\| (x_i - x_{i-1}, y_i - y_{i-1}) \|}\right) \quad (15)$$

The constraint function is used to ensure the feasibility of the candidate route, and the objective function aims to find the optimal flight route on the basis of satisfying the constraints.

IV. SDQPSO PLANNER

A. EVALUATION FUNCTION OF WAYPOINT

To take into account many desired characteristics, the evaluation function of flight route is used to find the optimal path, whereas the quality of each waypoint is difficult to be assessed according to the function. In view of the waypoints are independent of each other in a route, the evaluation function of the whole path is divided into the fitness function of the waypoint as the new criteria to evaluate the quality of the candidate waypoints. Each dimension of the particle is chosen according to the evaluation function of waypoint.

The evaluation function of the i-th waypoint is

$$SE(W_i) = \sum_{k=1}^{3} S_k(\overrightarrow{W_{i-1}W_i}) + \sum_{v=1}^{3} SC_v(\overrightarrow{W_{i-1}W_i}) \quad (16)$$

where $\sum_{k=1}^{3} S_k(\overrightarrow{W_{i-1}W_i})$ is the objective function of the waypoint, $\sum_{v=1}^{3} SC_v(\overrightarrow{W_{i-1}W_i})$ is the constraint function of the waypoint.

The objective function of waypoint is defined as

$$\sum_{k=1}^{3} S_k(\overrightarrow{W_{i-1}W_i}) = S_{\text{length}}(\overrightarrow{W_{i-1}W_i}) + S_{\text{altitude}}(\overrightarrow{W_{i-1}W_i}) + S_{\text{threat}}(\overrightarrow{W_{i-1}W_i}) \quad (17)$$

where $S_{\text{length}}(\overrightarrow{W_{i-1}W_i})$,$S_{\text{altitude}}(\overrightarrow{W_{i-1}W_i})$ and $S_{\text{threat}}(\overrightarrow{W_{i-1}W_i})$ are the objective functions of the length, the altitude and the threat risk for the i-th waypoint, respectively.

(1) Minimal Segment Length

For the i-th waypoint, $S_{\text{length}}(\overrightarrow{W_{i-1}W_i})$ is defined as

$$S_{\text{length}}(\overrightarrow{W_{i-1}W_i}) = \frac{\left\|\overrightarrow{W_{i-1}W_i}\right\| + \left\|\overrightarrow{W_iT}\right\|}{\left\|\overrightarrow{W_{i-1}T}\right\|}$$

$$= \frac{\|x_i - x_{i-1}, y_i - y_{i-1}, z_i - z_{i-1}\| + \|x_T - x_i, y_T - y_i, z_T - z_i\|}{\|x_T - x_{i-1}, y_T - y_{i-1}, z_T - z_{i-1}\|} \tag{18}$$

where T is the target point. Therefore

$$S_{\text{length}}(\overrightarrow{W_{i-1}W_i}) \in [1, 1.5] \tag{19}$$

(2) Minimal Segment Flight Altitude

For the i-th waypoint, $S_{\text{altitude}}(\overrightarrow{W_{i-1}W_i})$ is defined as

$$S_{\text{altitude}}(\overrightarrow{W_{i-1}W_i}) = \frac{A(\overrightarrow{W_{i-1}W_i}) - Z_{\min}}{Z_{\max} - Z_{\min}} \tag{20}$$

where $A(\overrightarrow{W_{i-1}W_i})$ is the average flight of the path segment. Therefore

$$S_{\text{altitude}}(\overrightarrow{W_{i-1}W_i}) \in [0, 1] \tag{21}$$

(3) Minimal Segment Threat Risk

For the i-th waypoint, $S_{\text{threat}}(\overrightarrow{W_{i-1}W_i})$ is defined as

$$S_{\text{threat}}(\overrightarrow{W_{i-1}W_i}) = \frac{L_{\text{inside}}}{\sum_{k=1}^{n_1} D_k} \tag{22}$$

where L_{inside} is the length of the segment that goes through the threat zones, n_1 is the number of the threat sources in the segment $\overrightarrow{W_{i-1}W_i}$. Therefore

$$S_{\text{threat}}(\overrightarrow{W_{i-1}W_i}) \in [0, 1] \tag{23}$$

The constraint of waypoint is defined as

$$\sum_{v=1}^{3} SC_v(\overrightarrow{W_{i-1}W_i}) = SC_{\text{collision}}(\overrightarrow{W_{i-1}W_i}) + SC_{\text{turning}}(\overrightarrow{W_{i-1}W_i}) + SC_{\text{slope}}(\overrightarrow{W_{i-1}W_i}) \tag{24}$$

(1) Segment Terrain Height Constraint

For the i-th waypoint, $SC_{\text{altitude}}(\overrightarrow{W_{i-1}W_i})$ is defined as

$$SC_{\text{altitude}}(\overrightarrow{W_{i-1}W_i}) = \frac{\sum_{k=1}^{N_d} s_1(\overrightarrow{W_{i-1,k}W_i})}{N_d}$$
$$\text{with} s_1(\overrightarrow{W_{i-1,k}W_i}) = \begin{cases} C, & \text{if } z_{i-1,k} \le MAP(x_{i-1,k}, y_{i-1,k}) \\ 0, & \text{else} \end{cases} \tag{25}$$

where $MAP(x_{i-1,k}, y_{i-1,k})$ is the altitude of the corresponding point ($x_{i\text{-}1,k}$, $y_{i\text{-}1,k}$) in the map, and C is the penalty constant.

(2) Segment Turning Angle Constraint

For the i-th waypoint, $SC_{\text{turning}}(\overrightarrow{W_{i-1}W_i})$ can be defined as

$$SC_{\text{turning}}(\overrightarrow{W_{i-1}W_i}) = \begin{cases} C, \text{if } \theta_i > \theta_{\max} \\ 0, \text{else} \end{cases} \tag{26}$$

where constant C penalizes the segment $\overrightarrow{W_{i-1}W_i}$ which requires larger turning angle than the maximum turning angle $\theta_{\max}$.

(3) Segment Slope Constraint

For the i-th waypoint, $SC_{\text{slope}}(\overrightarrow{W_{i-1}W_i})$ can be defined as

$$SC_{\text{slope}}(\overrightarrow{W_{i-1}W_i}) = \begin{cases} 0, & \text{if } \varphi_{\min} \le \varphi_i \le \varphi_{\max} \\ C, & \text{else} \end{cases} \tag{27}$$

where constant C penalizes the segment $\overrightarrow{W_{i-1}W_i}$, which requires slope angle beyond the given scope $[\varphi_{\min}, \varphi_{\max}]$.

B. TRADITIONAL QPSO

In the PSO system, the convergence of particles is achieved through the form of orbits. During the search process of the algorithm, the motion area of each time will be limited due to the maximum speed of the particles. QPSO algorithm is a stochastic parallel probability search algorithm based on the PSO framework. It is inspired by probability optimization algorithm of quantum computing principle, which is believed that particles have quantum behavior. The state of particle is described by a wave function as

$$\varphi(Y) = \frac{1}{\sqrt{L}} e^{\frac{-|Y|}{L}}, L = \frac{1}{\beta} = \frac{h^2}{mr} \tag{28}$$

where L is the probability that the particle appears at a relative point, and Y is the state of particle.

For QPSO algorithm, it is assumed that the particle move in a one-dimensional potential well centered on the attraction point q, the position of each particle is updated by Monte Carlo method. The position equation of the particle is obtained as

$$X_{t+1} = q \pm \frac{L}{2} \ln(\frac{1}{u}), u \sim U(0, 1) \tag{29}$$

where L is the characteristic length of delta potential well.

Set the global best particle position $\boldsymbol{P}_g = [p_{g1}, p_{g2}, \cdots, p_{gn}]$, the personal best position $\boldsymbol{P}_i = [p_{i1}, p_{i2}, \cdots, p_{in}]$, the attractor $\boldsymbol{Q}_i = [q_{i1}, q_{i2}, \cdots, q_{in}]$. q_{ij}^k is the j-th dimension of the i-th particle's local attractor at the k-th iteration, which is defined as

$$q_{ij}^k = \zeta p_{ij}^k + (1 - \zeta) p_{gj}^k \tag{30}$$

The wave function for each dimension of the particle is defined as

$$\varphi[x_{ij}(t+1)] = \frac{1}{\sqrt{L_{ij}(t)}} \exp\left[-\frac{|x_{ij}(t+1) - q_{ij}(t)|}{L_{ij}(t)} \right] \tag{31}$$

The position of each particle is updated as follows:

$$x_{ij}^{k+1} = q_{ij}^k \pm (L_{ij}^k/2) \ln(1/u)\ u \sim U(0, 1) \tag{32}$$

where L_{ij}^k is computed as

$$L_{ij}^k = 2b \left| mbest_j^k - x_{ij}^k \right|, \quad with\ mbest_j^k = \sum_{i=1}^{m} p_{ij}^k / m \tag{33}$$

where b is the contraction-expansion coefficient, $b < 1.782$, and $mbest_j^k$ is the mean best position of m particles. Substitute (30) and (33) into (32), the updated function is given by

$$x_{ij}^{k+1} = \zeta p_{ij}^k + (1 - \zeta) p_{gj}^k \pm b \left| \sum_{i=1}^{m} p_{ij}^k / m - x_{ij}^k \right| \ln(1/u) \tag{34}$$

C. THE IDEA OF SDQPSO

For QPSO, each dimension of the particles is often initialized randomly. Therefore, it is difficult to ensure that most dimensions in the initial solutions are optimal according to the evaluate function. After the initialization process, each particle with multiple dimensions is regarded as an integrated individual, which will be brought into the fitness function and conduct the selection operation. With the increase of iterations, there are several considerations for an ideal path planner including: firstly, the quality of each dimension in the particles can hardly be evaluated during the optimization operation by the evaluation function of flight route. The particle is chosen associated with the performance of the whole path. Moreover, a particle with many superior dimensions may be discarded due to the impact of some bad dimensions. Thus the search efficiency of the algorithm will be greatly reduced. In order to solve the above these problems, each dimension of the particle is regarded as an individual, which replace the particle as a whole in QPSO, a dimension by a dimension, to pick out an entire particle from the $m-1$ particles.

Suppose the position of each particle is $\boldsymbol{x}_i^t = [x_{i1}^t, x_{i2}^t, \cdots, x_{in}^t]$, each particle is an individual with n dimensions, which is shown in figure 5. At the first stage of the search, if the best fitness is achieved by $x_{m-1,1}^t$ according to the evaluation function of waypoint, $x_{m-1,1}^t$ will be chosen as the first dimension solution x_{m1}^t of the m-th particle. It means that $x_{m-1,1}^t$ is most valuable dimension among $m-1$ particles. Then on the basis of this dimension x_{m1}^t, we continue to search for the next dimension of the m-th particle. From figure 5, it is seen that x_{32}^t is chosen as the second dimension solution x_{m2}^t of the m-th particle. Through the strategy of dimension search, an optimal solution including all the dimensions is generated from m-1 particles associated with the new evaluation criteria. The particle forms incrementally a new solution x_m^t of the problem at time $t \quad (t = 1, 2, \cdots, t_{\max})$ when all the dimensions are obtained.

The strategy of particle dimension search is introduced into QPSO algorithm to solve the complex problem, which not only optimizes the composition of particles, but also improves the search efficiency of the algorithm in solution space.

The procedure of SDQPSO is described as follows:

Step 1: Set appropriate parameters of QPSO, including the population size of particles, the dimension of particle, the maximum iteration $T_{\max}$ and contraction-expansion coefficient b. $m-1$ particles are initialized in problem space, and the m-th particle is generated by dimension search approach.

Step 2: The position of each particle is evaluated by the fitness function.

Step 3: Comparison to personal best. If the fitness value of the new personal optimal solution is better than the old one, replace the old personal best solution.

Step 4: Comparison to global best. If the current global best position is better than the global best position searched so far, the old global optimal location will be replaced.

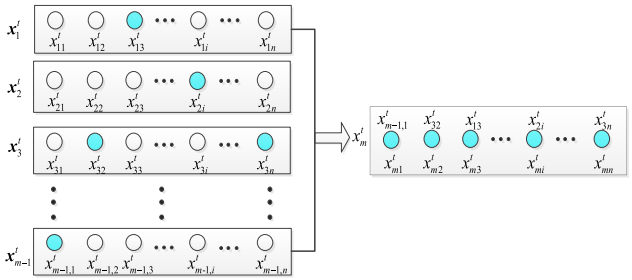


FIGURE 5. The dimension search process of the particles.

Step 5: Calculate the local attractor of each particle and the mean best position of the population, then update the position of m-1 particles according to Eq.(34). The m-th solution is generated by dimension search approach.

Step 6: If the end condition is met, i.e., when there is $t > T_{\max}$, then output the best solution. Otherwise, go to Step 2.

D. TIME COMPLEXITY ANALYSIS

According to the procedure of QPSO algorithm, there are four main steps to be considered for the time complexity of QPSO framework. The time complexity of QPSO algorithm is analyzed as follows:

In the initialization step of QPSO algorithm, it is assumed that the time of generating a random number with uniform distribution is set to $c_1 n$. The execution time of calculating the fitness of the evaluation function is denoted as a function $f(n)$ with number n of variables. The time complexity is described as

$$O(m(c_1 n + f(n))) = O(n + f(n)) \tag{35}$$

where m is the number of the particles.

In the step of updating personal best solution, the time of comparing a new personal best solution with an old one and replacing an old solution are c_2 and $c_3 n$, respectively. Thus the time complexity is represented as

$$O(m(c_2 + c_3 n)) = O(n) \tag{36}$$

In the step of updating global best solution, the time of comparing a solution with the best solution is c_4, the time of replacing an optimal solution is $c_5 n$, the time complexity of the best solution is described as

$$O(m(c_4 + c_5 n)) = O(n) \tag{37}$$

In the step of updating of particles' position, the execution time of generating a new solution using Eq.(34) is denoted as $c_6 n$. The time complexity is described as

$$O(m(c_6 n)) = O(n) \tag{38}$$

For each generation, the time complexity is written as

$$TE(n) = O(n + f(n)) + 3O(n) = O(n + f(n)) \tag{39}$$

If the maximum iteration is used as the termination conditions, the total time complexity of QPSO is calculated as

$$T(n) = O(I(n + f(n))) = O(n + f(n)) \tag{40}$$

where I is maximum number of iterations.

For the initialization process of the proposed algorithm (SDQPSO), the time of calculating evaluation fitness of waypoint is denoted as $(m-1)c_7$, the time of dimension search is denoted as $(m-1)c_8 n$, so the time complexity of step 1 is denoted as

$$O((m-1)c_1 n + mf(n) + (m-1)(c_7 + c_8 n)) = O(n + f(n)) \tag{41}$$

For step 2 and step 3, the time complexity is similar to QPSO,

$$O(m(c_2 + c_3 n + c_4 + c_5 n)) = O(n) \tag{42}$$

For step 4, the time complexity is described as

$$O((m-1) \cdot (c_6 n + c_8 n + c_7)) = O(n) \tag{43}$$

For each generation, the time complexity is written as

$$TE(n) = O((n + f(n)) + 2O(n) = O(n + f(n)) \tag{44}$$

The total time complexity of SDQPSO is described as

$$T(n) = O(I(n + f(n))) = O(n + f(n)) \tag{45}$$

Therefore, the time complexity of SDQPSO is consistent with QPSO.

E. ROUTE PLANNING METHOD OF SDQPSO

This section discusses the application of SDQPSO to the route planning in details, mainly including the optimization for the initialization process and the waypoint search based on dimension information.

(1) Initialization Process Optimization

In most planners, the waypoint of the route is often denoted as a 3-D coordinate in Cartesian coordinate system. To simplify the representation of the waypoint, a new coordinate system will be established by Cartesian coordinate system. Firstly, we project the start point $S\{x_s, y_s, z_s\}$ and target point $T\{x_t, y_t, z_t\}$ in the Cartesian coordinate system (x, y, z) to O-XY plane. The projection of S is taken as the origin O′ of the new coordinate system (x', y', z'). The projections of S and T are connected with a straight line, which is defined as the new X'-axis. The new X'-axis is divided into $n+1$ equal parts and the dividing points compose the set $\boldsymbol{D} = \{D_1, D_2, D_3, \cdots, D_i, \cdots, D_n\}$. Make the lines $L = \{L_1, L_2, L_3, \cdots, L_i, \cdots, L_n\}$ perpendicular to X'-axis at each waypoint, as shown in figure 6. Z'-axis of new coordinate system is the same direction as Z-axis, and Y'-axis is orthogonal to X'-axis and Z'-axis. The new coordinate system O'-X'Y'Z' is transformed by a rotation of the Cartesian coordinate system O-XYZ.

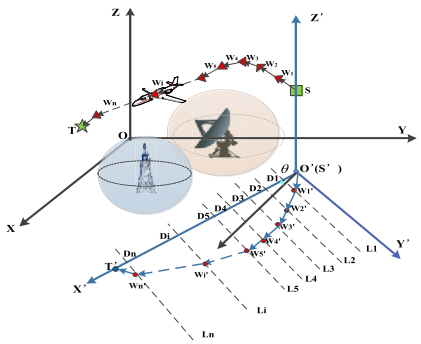


FIGURE 6. Coordinate system of UAV route.

The coordinate transformation between the new coordinate system O'-X'Y'Z' and the Cartesian coordinate system O-XYZ is defined as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} + \begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix} \tag{46}$$

where (x, y, z) and (x', y', z') are coordinates of Cartesian coordinate system O-XYZ and the new coordinate system O'-X'Y'Z' frame, respectively.

Because the map information is known before planning, the altitude data of the map can be fully adopted to optimize the initial path during the initialization process. z-coordinate is initialized as follows:

$$Z_{\max} \geq Z_i^t \geq (Z_{\text{terrain}} + D_{\text{safe}}) \tag{47}$$

where Z_{terrain} is the altitude in the map, D_{safe} is a safety reference, $Z_{\max}$ is the upper limit.

In general, the length of the path is considered at the first step. The shorter the length of the path, the higher the quality of the path is. Taking into account the term of path length, y-coordinate of the first particle for the initial particles is expressed by

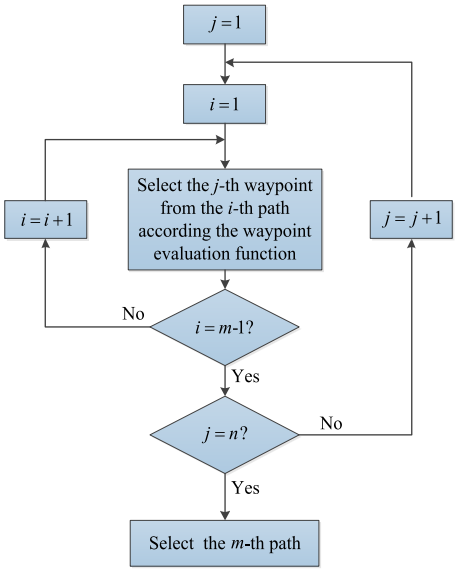
$$y = kx + b \quad \text{with} \begin{cases} k = \frac{y_t - y_s}{x_t - x_s} \\ b = y_s \end{cases} \tag{48}$$

(2) Waypoint Search Based on Particle Dimension

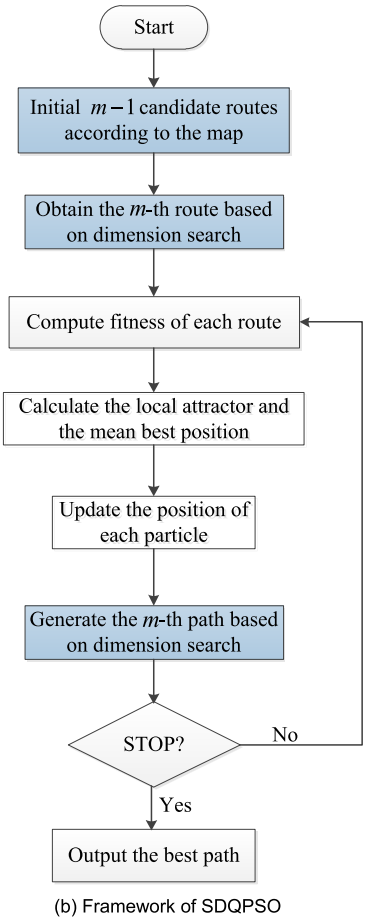
The general flow chart of SDQPSO algorithm is described in figure 7, and the improvement parts are outlined in blue. The particles are initialized according to the start point, the target and the height data of the terrain. The procedure of path planner based on SDQPSO algorithm is described as follows:

Step 1: Choose appropriate parameters of SDQPSO, including the population of particles, the dimension of particle, the maximum iteration $T_{\max}$ and contraction-expansion coefficient b.

Step 2: Input the environmental information, such as the upper bound and the lower bound of planning space, and the threat terrain.



(a) Framework of dimension search



(b) Framework of SDQPSO

FIGURE 7. Flow chart of SDQPSO algorithm.

Step 3: Initialize $m-1$ routes in the mission space. The x-coordinate is initialized by the principle of equal division. The z-coordinate of each particle is set according to Eq.(47). The y-coordinate of the first particle is given by Eq.(48), and the y-coordinates of the other particles are initialized randomly.

Step 4: The dimension index increases from $j = 1$ to $j = n$, the m-th initial route is generated by dimension search approach.

Step 5: Calculate the fitness of each route, and obtain the personal best position of each particle and the global best position.

Step 6: Calculate the local attractor of each particle and the mean best position of the population, then update the position of m-1particles according to Eq.(34). The m-th route is generated by dimension search approach.

Step 7: If the stopping condition is met, i.e. when there is $t > T_{\max}$, then output the best fitness and the corresponding route. Otherwise, go to Step 5.

V. SIMULATION EXPERIMENT AND ANALYSIS

To test the performance of SDQPSO with the constrained problem of UAV route planning, a series of simulation experiments were implemented in MATLAB R2014a on a PC with 2.4 GHz running Windows 10. The mission space was from the real terrain elevation maps [30]. To avoid the randomness of test results, each group of experiments was executed 30 runs independently.

A. COMPARION OF DIFFERENT N_W

To test the impact of the number of waypoints, the simulation comparisons with different N_w, e.g. $N_w = 6$, $N_w = 9$, $N_w = 12$, $N_w = 15$, $N_w = 18$, and $N_w = 21$, were executed in two cases by 30 runs. The parameters of SDQPSO are set as follows: the population size m is 100, the maximum iteration $It_{\max}$ is 200, the contraction-expansion coefficient b is decreased linearly from 0.7 to 0.3, and the number of the division point N_d is 5. The scope of the mission space is limited within the space $[0, 9000] \times [0, 9000] \times [0, 800]$. The start location of mission is set to (900, 8100, 350), and the destination location is set to (7650, 1800, 500). The same terrain is used in case I and case II. But the numbers of the obstacles in two cases are different (i.e. case I with 8 obstacles, case II with 28 obstacles), as shown in figure 8 and figure 9.

The relationships between iteration number ($0 \le It \le 200$) and average best fitness in 30 runs are displayed in figure 8(a) and figure 9(a) with given different numbers of the waypoints. It is noted that the average best fitness value is described as the reciprocal of the mean best cost value in figure 8(a) and figure 9(a). The statistical results about the minimum cost, the mean cost, the standard deviation of cost value, FR, $\tilde{G}_c$ and G_c are listed in table 1, and where the best results are highlighted in boldface. Specially, FR represents the percentage of feasible paths in all the routes. $\tilde{G}_c$ is the average iteration number when all the constraints are met, which describes the convergence speed of evolution. G_c is the best result of the iteration number among 30 runs. It is noted that only the successful runs which can

TABLE 1. The statistical results for the different N_w in two cases.

Item	N_w	Min. Cost	Mean Cost	Std. dev.	G_c	$\bar{G}_c$	FR(%)
Case I	6	**1.6508**	**1.6576**	**0.018667**	**1**	8	**100**
	9	**1.6508**	1.6581	0.022749	**1**	8	**100**
	12	1.6523	1.6884	0.036529	2	**7**	**100**
	15	1.6535	1.7918	0.558620	3	14	96.67
	18	1.6544	1.7983	0.548243	3	14	96.67
	21	1.6551	2.0156	0.953777	5	16	90
Case II	6	1.6771	**1.9861**	**0.806324**	**1**	**11**	**93.33**
	9	**1.6704**	2.0752	0.957853	2	**11**	90
	12	1.6745	2.0980	1.176079	3	21	86.67
	15	1.6707	2.4486	1.279472	3	22	76.67
	18	1.6757	3.0429	1.562069	4	24	66.67
	21	1.6757	3.1257	1.650658	5	32	60

find feasible routes are used for the statistics of $\tilde{G}_c$ and G_c in 30 runs experiments. In the two cases, it is obvious that with the increase of the waypoint number N_w, the standard deviation, the mean fitness value and the average iteration $\tilde{G}_c$ become larger, and FR goes down gradually. In case I, $N_w = 6$ is superior to $N_w = 9$, $N_w = 12$, $N_w = 15$, $N_w = 18$ and $N_w = 21$ in mean cost, and standard deviation. Besides, the smallest value of $\tilde{G}_c$ is obtained when N_w=6 and $N_w = 9$.

In case II, due to the increase of the threat sources, nearly all the indicators get worse than those of case I. Because the route of $N_w = 6$ has a risk of trapping into the dangerous zones, $N_w = 9$ has a better convergence speed and safety than other N_w value. $N_w = 9$ is the most appropriate choice in case II. Under the impact of the increasing threat source in mission space, the routes with different N_w can converge to a rather good position to avoid the risk of the radars and missiles in case II. Because there are more threat sources of the radars, missiles and anti-aircraft guns, it is obvious that the successful rate FR is significantly lower than that of case I. If N_w is too large, the problem of route planning will become very complicated and the possibility of finding an optimal path is decreased. Besides, the larger N_w is, the slower the convergence speed is. However, if N_w is too small, the route can hardly to be described accurately, and it is unable to successfully avoid all obstacles and threat sources. Therefore, N_w should be large enough to not only protect the danger from being detected and attacked, but also guarantee the searching accuracy.

B. COMPARISON OF DIFFERENT N_d

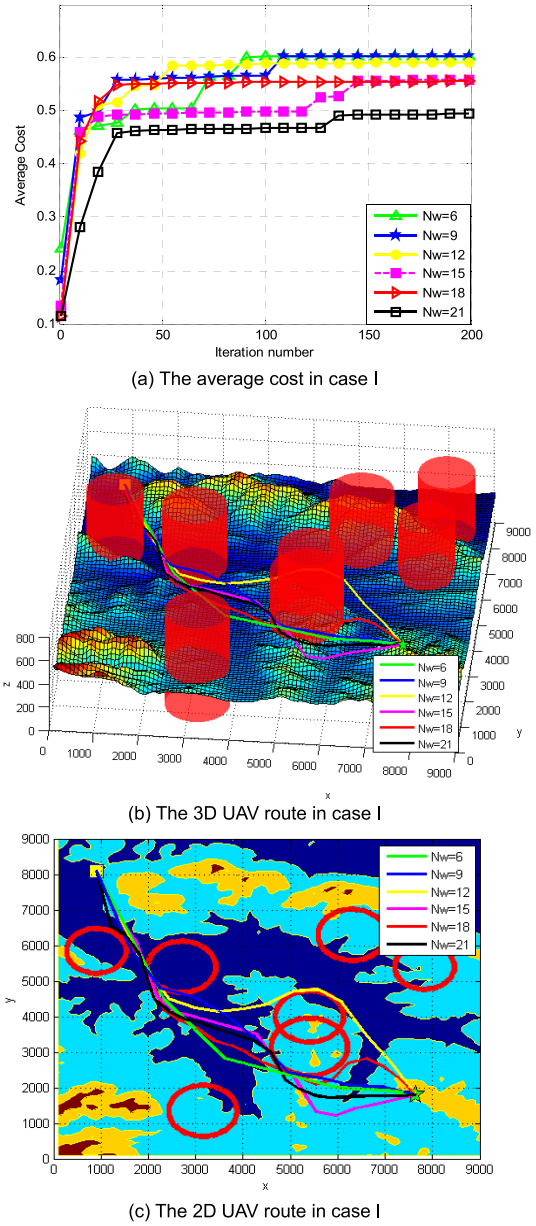
N_d is another non-QPSO-related parameter, which represents the number of division points in each segment of a flight route. The number of division points N_d should be enough to detect whether the route violates the threats of missiles, radars, anti-aircraft guns and mountains in the mission space. It is necessary that the number of division points between two adjacent segments is smaller than the scope of missiles, radars or mountains. According to [8], the relationship between N_w and N_d is described as:

$$\frac{PL}{(N_w + 1) \cdot N_d} < D \tag{49}$$

where PL represents the route length, and D is the minimal diameter of all the missiles and radars in space, N_w is the number of the waypoints (not including start point and endpoint). In general, the range of mountains and missiles is usually larger than 0.5, therefore, D is set to 0.5.

Because the path length ratio is required to be less than 1.5, there is $N_d > 4$ with the smallest $N_w = 6$ according to Eq.(49).

To further analyze the influence of N_d on the evaluation accuracy and computational efficiency, different N_d is

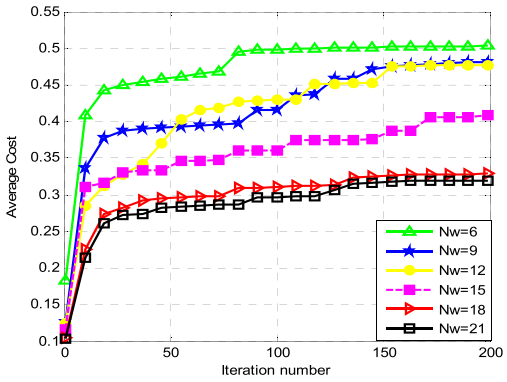


(a) The average cost in case I

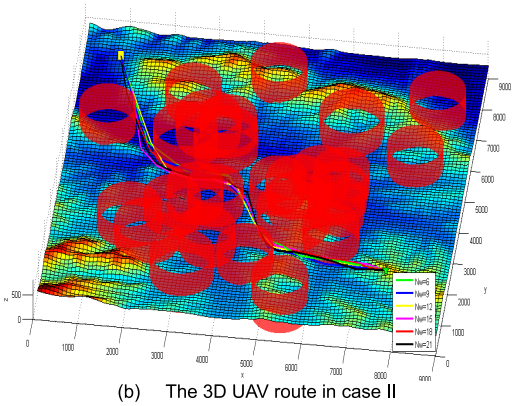
(b) The 3D UAV route in case I

(c) The 2D UAV route in case I

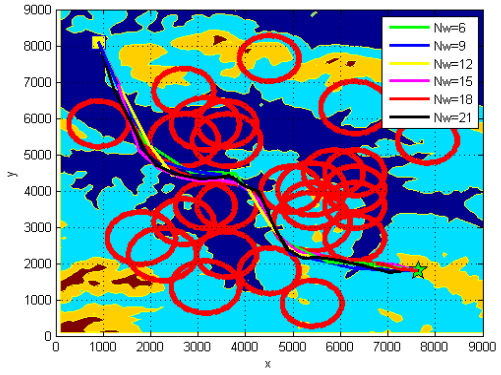
FIGURE 8. The cost and route in case I.



(a) The average cost in case II



(b) The 3D UAV route in case II



(c) The 2D UAV route in case II

FIGURE 9. The cost and route in case II.

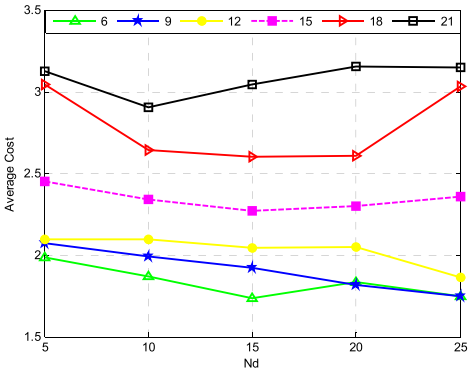


FIGURE 10. The average cost of different N_d value in case II.

TABLE 2. Performance comparison of different. N_d in case II.

Item	N_d	Mean Cost	Std. dev.	G_c	$\bar{G}_c$	FR (%)
$N_w = 6$	5	1.9861	0.806324	**1**	11	93.33
	10	1.8666	0.569755	**1**	5	96.67
	15	**1.7361**	**0.063463**	**1**	4	**100**
	20	1.8357	0.562897	**1**	**3**	96.67
	25	1.7467	0.071956	**1**	4	**100**
$N_w = 9$	5	2.0752	0.957853	2	11	90
	10	1.9920	0.875790	2	16	93.33
	15	1.9199	0.827847	**1**	12	93.33
	20	1.8147	0.605953	2	13	96.67
	25	**1.7479**	**0.178064**	**1**	**10**	**100**
$N_w = 12$	5	2.0980	1.176079	3	21	86.67
	10	2.0955	0.960834	2	24	90
	15	2.0446	0.925799	2	23	90
	20	2.0479	0.950057	**1**	**15**	90
	25	**1.8614**	**0.596642**	2	26	**96.67**
$N_w = 15$	5	2.4486	1.279472	**3**	22	76.67
	10	2.3403	1.249265	**3**	14	83.33
	15	**2.2716**	**1.169228**	**3**	**8**	83.33
	20	2.2985	1.178222	**3**	17	**86.67**
	25	2.3584	1.233191	**3**	23	**86.67**
$N_w = 18$	5	3.0429	1.562069	4	24	66.67
	10	2.6403	1.407858	**3**	**15**	73.33
	15	**2.6033**	**1.394549**	**3**	29	**76.67**
	20	2.6048	1.395692	4	27	**76.67**
	25	3.0341	1.552579	**3**	25	63.33
$N_w = 21$	5	3.1257	1.650658	5	32	60
	10	**2.9023**	1.552213	5	**13**	**66.67**
	15	3.0460	**1.536114**	**4**	14	60
	20	3.1545	1.585999	**4**	35	60
	25	3.1805	1.582018	6	36	56.67

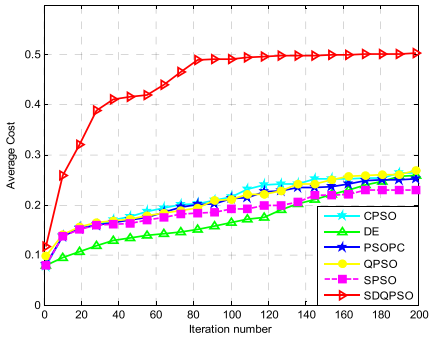
selected, e.g. $N_d = 5$,$N_d = 10$, $N_d = 15$, $N_d = 20$ and $N_d = 25$ to test the effect during 30 runs. The average convergence curves and the statistic data in case II with different N_d under different N_w are shown in the figure 10 and table 2.

The statistic results are listed in table 2. It is seen that the impact of different N_d is closely related to N_w. In each column, we can analyze the performance of the planner associated with the change in value of N_w. FR of $N_d = 25$ is the best when N_w is less than or equal to 15.

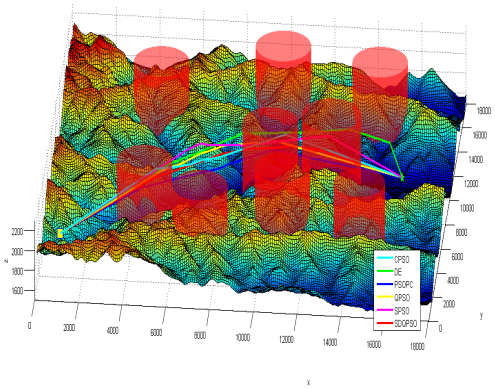
It is evident that, when N_d is enough large (e.g. $N_d = 25$) and N_w is less than 15, the most indicators of the average mean cost, FR and the standard deviation are better than those when N_d is set to be a small value (e.g. $N_d = 5$). Hence $N_d = 25$ is recommended with $N_w \leq 15$. But when N_w is larger than 15, the performance of SDQPSO with too large N_d is worse than the situation that N_d is equal to a medium value (e.g. $N_d = 15$). So the number of the waypoints N_w has a significant effect on the performance of N_d. Generally, the route can obtain superior convergence speed and accuracy by choosing an appropriate N_d on the basis that the number of waypoints N_w has been determined

C. COMPARISON OF DIFFERENT METHODS

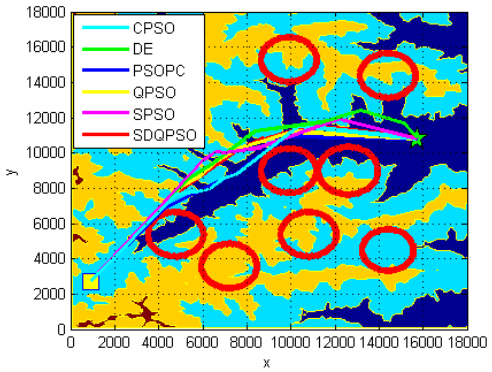
To verify the superiority of the proposed algorithm, QPSO algorithm [31] and other representative optimization algorithms, i.e., CPSO [32], DE [33], PSOPC [34] and SPSO [6],



(a) The average cost in case III

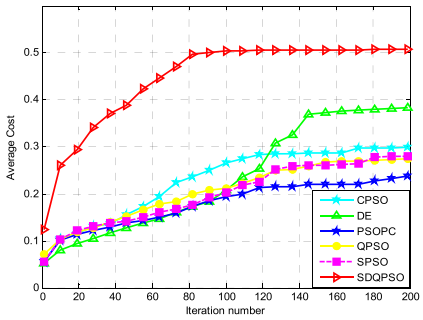


(b) The 3D UAV route in case III

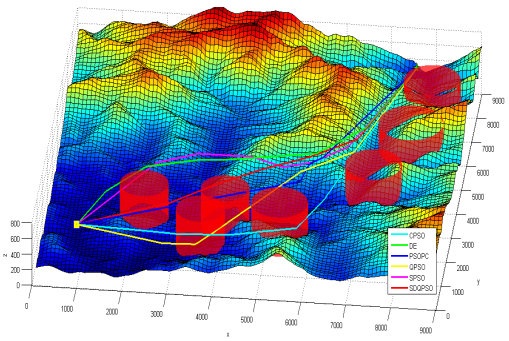


(c) The 2D UAV route in case III

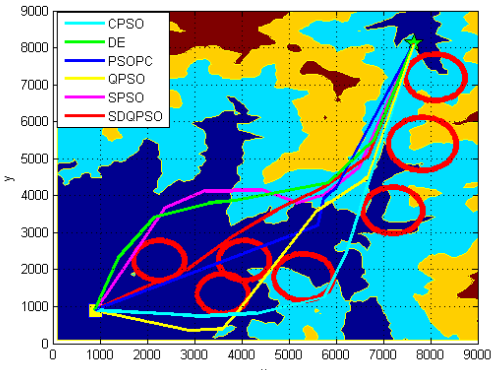
FIGURE 11. The comparison results by various methods in case III.



(a) The average cost in case IV



(b) The 3D UAV route in case IV



(c) The 2D UAV route in case IV

FIGURE 12. The comparison results by various methods in case IV.

were compared for the search effectiveness in three cases of UAV route planning. The parameter values are set in table 3, and the statistical results during 30 runs are listed in table 4. In case III and case IV, the number of waypoints is the same for all the comparison methods, and the waypoints number N_w is set to nine for the six methods in case V.

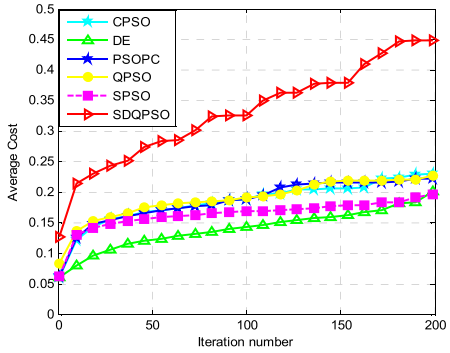
Figure 11, figure 12 and figure 13 show the UAV routes for cases III, IV and V generated by the above six algorithms during 30 independent runs. Parts (b) of the figures show the 3D views of UAV routes obtained by the comparison algorithms in the digital maps, and the cylinders represent the threat areas of radars, missiles and anti-aircraft guns. Parts (c) of the figures display the 2D views of parts (b).

Under the conditions of the same parameters, all the planners are able to find an optimal route and satisfy the constraints in case III, but the routes are different with various

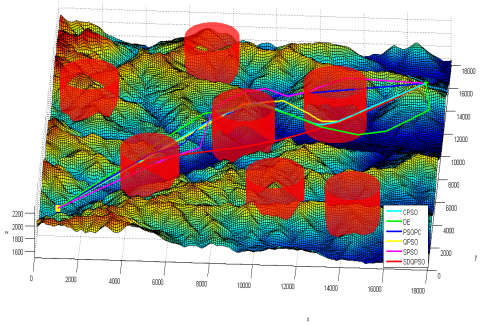
TABLE 3. The parameter values of six methods.

Parameter	SDQPSO	CPSO	DE	PSOPC	QPSO	SPSO
Size	100	100	100	100	100	100
$It_{\max}$	200	200	200	200	200	200
ω	NA	1	NA	[0.7 0.9]	NA	0.7298
χ	NA	0.73	NA	NA	NA	NA
C_1	NA	2.05	NA	0.5	NA	1.4960
C_2	NA	2.05	NA	0.5	NA	1.4960
C_3	NA	NA	NA	[0.4 0.6]	NA	NA
F	NA	NA	0.4	NA	NA	NA
Cr	NA	NA	0.9	NA	NA	NA
b	[0.3 0.7]	NA	NA	NA	[0.3 0.7]	NA

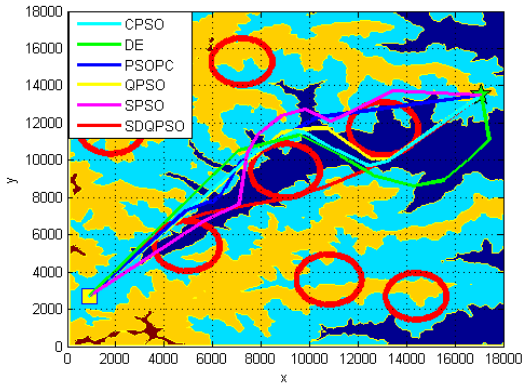
methods. The FR of SDQPSO can achieve 96.67%, while those of CPSO, DE, PSOPC, QPSO and SPSO are 53.33%, 60%, 50%, 56.67% and 53.33%, respectively, as shown in table 4. The standard deviation of SDQPSO is about 0.58.



(a)The average cost in case V



(b)The 3D UAV route in case V



(c) The 2D UAV route in case V

FIGURE 13. The comparison results by various methods in case V.

It is obvious that SDQPSO is smaller than other five planners, which reflects SDQPSO to be more stable than other methods. It is seen that $\tilde{G}_c$ of SDQPSO is the smallest in the six methods, which demonstrates that SDQPSO has faster convergence speed than other methods. Moreover, SDQPSO obtains more excellent results of the minimum cost, mean cost and G_c among all the planners. The robustness and effectiveness of SDQPSO are superior to other five methods. Due to the adoption of the initial optimization strategy, it is easy to find that the initial fitness value of the proposed algorithm is superior to the other five methods, as shown in figure 11(a). For the feasible results, figure 11(b) and figure 11(c) show the planned routes of CPSO, DE, PSOPC, QPSO, SPSO and SDQPSO in case III, all the six planners can find a safe and collision-free route. At the same time, the route obtained by SDQPSO has the best average cost value and the minimum number of iterations to satisfy the constraints.

TABLE 4. The comparison of different algorithms in three cases.

Item	Method	Mean Cost	Std. dev.	$\tilde{G}_c$	FR(%)	t / s
Case III	SDQPSO	**1.9882**	**0.578465**	**19**	**96.67**	50.2335
	CPSO	3.7842	1.767702	84	53.33	49.1650
	DE	3.7506	1.592502	148	60	49.3234
	PSOPC	3.9719	2.130870	92	50	52.2217
	QPSO	3.7196	1.789706	101	56.67	48.0931
	SPSO	4.1808	1.856266	118	53.33	**47.2355**
Case IV	SDQPSO	**1.9718**	**0.773363**	**21**	**93.33**	45.6094
	CPSO	3.3513	1.520785	75	50	50.9383
	DE	2.6110	1.352406	114	73.33	48.7721
	PSOPC	4.1955	1.968875	87	33.33	49.3434
	QPSO	3.6478	1.770559	91	46.67	**43.1823**
	SPSO	3.5621	1.863998	94	50	44.7163
Case V	SDQPSO	**2.2283**	**0.932269**	**67**	**90**	49.7268
	CPSO	4.3498	1.539006	112	33.33	51.7520
	DE	4.9638	1.474429	176	23.33	48.2756
	PSOPC	4.4842	1.537511	116	26.67	48.0467
	QPSO	4.4012	1.486321	113	30	47.1889
	SPSO	5.0877	1.173927	186	13.33	**46.4779**

Figure 12(a) displays the corresponding convergence curves of statistical result in case IV. The path quality generated by SDQPSO is better than those generated by the other five algorithms. Compared to other methods, the feasible rate FR of SDQPSO is 93.33%, which is still much higher than those of the other five algorithms. Besides that, the minimum cost, mean cost and standard deviation obtained by SDQPSO are still smaller than those by the other five algorithms. The FR of PSOPC is close to CPSO and SPSO in case III, but the FR of PSOPC is much lower than CPSO and SPSO in case IV. It is seen that PSOPC fails to find a safe path since the route passes through a threat in case IV. SDQPSO is originally designed to make full use of the information of the given environment.

To further demonstrate the robustness and effectiveness of SDQPSO, the number of waypoints is increased in the comparison experiment (case V). The convergence curves of the average cost values over 30 runs are shown in figure 13(a). It is seen that the initial fitness cost value of SDQPSO is better than other algorithms. For the simulation results, the FR still remains about 90%. It is obvious that SDQPSO achieves a higher possibility to obtain a feasible route. In table 4, $\tilde{G}_c$ represents that SDQPSO planner has a faster convergence speed when the number of waypoints increases. However, the convergence of SDQPSO in case V is a little slower than the other two cases.

VI. CONCLUSION

In this paper, an improved QPSO algorithm, namely SDQPSO, has been proposed by employing a novel search strategy of particle dimension. It is applied to solve effectively the route planning problem considering the dynamic properties of fixed-wing UAV and the complexity of real 3D environment. Meanwhile, the evaluation function for waypoint is presented to raise the search efficiency of the

new path planner. The initialize optimization process is used to improve the quality of the initial routes. Based on the experimental discussion, the two important parameters, i.e. N_w and N_d, were discussed about the impact on the statistical performance of the route planning. The idea of particle dimension search depends on the independence of the evaluation function of waypoint. The evaluation criteria of waypoints can be employed independently. Therefore, the proposed search strategy may be extended to other path planners based on intelligent optimization. The comparison experimental results in the three typical cases demonstrated that the planner based on SDQPSO can produce superior trajectories than other five representative planners under the same 3D circumstances. SDQPSO can be applied to solve route planning problem of fixed-wing UAVs under various constraints at a faster convergence speed, more steady robustness and higher accuracy.

It is noted that as the number of the particle dimensions increases, the probability of SDQPSO to obtain a feasible solution would be decreased. If the number of the waypoints is too large, it will be difficult for current SDQPSO algorithm to find an optimal solution. Therefore, for large-scale path planning problem, we may consider cooperative coevolution framework to solve by decomposing the large-scale optimization problem into some sub-problems in future work. In addition, the convergence speed of SDQPSO and the probability of finding a satisfactory route should be further investigated in large scale problem.

REFERENCES

[1] Y. Liu, Q. Wang, H. Hu, and Y. He, "A novel real-time moving target tracking and path planning system for a quadrotor UAV in unknown unstructured outdoor scenes," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 11, pp. 2362–2372, Nov. 2019.

[2] S. Minaeian, J. Liu, and Y.-J. Son, "Vision-based target detection and localization via a team of cooperative UAV and UGVs," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 7, pp. 1005–1016, Jul. 2016.

[3] F. Luo, C. Jiang, S. Yu, J. Wang, Y. Li, and Y. Ren, "Stability of cloud-based UAV systems supporting big data acquisition and processing," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 866–877, Jul. 2019.

[4] T. Moranduzzo and F. Melgani, "Detecting cars in UAV images with a catalog-based approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6356–6367, Oct. 2014.

[5] P. Yao and H. Wang, "Dynamic adaptive ant lion optimizer applied to route planning for unmanned aerial vehicle," *Soft Comput.*, vol. 21, no. 18, pp. 5475–5488, Sep. 2017.

[6] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 132–141, Feb. 2013.

[7] Y. Chen, J. Yu, Y. Mei, Y. Wang, and X. Su, "Modified central force optimization (MCFO) algorithm for 3D UAV path planning," *Neurocomputing*, vol. 171, pp. 878–888, Jan. 2016.

[8] P. Yang, K. Tang, J. A. Lozano, and X. Cao, "Path planning for single unmanned aerial vehicle by separately evolving waypoints," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1130–1146, Oct. 2015.

[9] T. Petković, D. Puljiz, I. Marković, and B. Hein, "Human intention estimation based on hidden Markov model motion validation for safe flexible robotized warehouses," *Robot. Comput.-Integr. Manuf.*, vol. 57, pp. 182–196, Jun. 2019.

[10] M. Zimmermann and C. König, "Integration of a visibility graph based path planning method in the ACT/FHS rotorcraft," *CEAS Aeronaut. J.*, vol. 7, no. 3, pp. 391–403, Sep. 2016.

[11] E. Masehian and M. R. Amin-Naseri, "A Voronoi diagram-visibility graph-potential field compound algorithm for robot path planning," *J. Robotic Syst.*, vol. 21, no. 6, pp. 275–300, Jun. 2004.

[12] B. Fu, L. Chen, Y. Zhou, D. Zheng, Z. Wei, J. Dai, and H. Pan, "An improved A* algorithm for the industrial robot path planning with high success rate and short length," *Robot. Auto. Syst.*, vol. 106, pp. 26–37, Aug. 2018.

[13] Y. Fu, M. Ding, and C. Zhou, "Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 2, pp. 511–526, Mar. 2012.

[14] B. Zhang and H. Duan, "Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 97–107, Jan. 2017.

[15] W. Zhu and H. Duan, "Chaotic predator–prey biogeography-based optimization approach for UCAV path planning," *Aerosp. Sci. Technol.*, vol. 32, no. 1, pp. 153–161, Jan. 2014.

[16] Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV," *Aerosp. Sci. Technol.*, vol. 16, no. 1, pp. 47–55, Jan. 2012.

[17] N. Ozalp and O. K. Sahingoz, "Optimal UAV path planning in a 3D threat environment by using parallel evolutionary algorithms," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, May 2013, pp. 308–317.

[18] T. T. Mac, C. Copot, D. T. Tran, and R. D. Keyser, "A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization," *Appl. Soft Comput.*, vol. 59, pp. 68–76, Oct. 2017.

[19] J.-H. Zhang, Y. Zhang, and Y. Zhou, "Path planning of mobile robot based on hybrid multi-objective bare bones particle swarm optimization with differential evolution," *IEEE Access*, vol. 6, pp. 44542–44555, 2018.

[20] W. Deng, H. Zhao, L. Zou, G. Li, X. Yang, and D. Wu, "A novel collaborative optimization algorithm in solving complex optimization problems," *Soft Comput.*, vol. 21, no. 15, pp. 4387–4398, Aug. 2017.

[21] W. Deng, H. Zhao, X. Yang, J. Xiong, M. Sun, and B. Li, "Study on an improved adaptive PSO algorithm for solving multi-objective gate assignment," *Appl. Soft Comput.*, vol. 59, pp. 288–302, Oct. 2017.

[22] S. Khan, M. Kamran, O. U. Rehman, L. Liu, and S. Yang, "A modified PSO algorithm with dynamic parameters for solving complex engineering design problem," *Int. J. Comput. Math.*, vol. 95, no. 11, pp. 2308–2329, Nov. 2018.

[23] S. Khan, S. Yang, and O. U. Rehman, "A dynamic particle swarm optimization method applied to global optimizations of engineering inverse problem," *COMPEL-Int. J. Comput. Math. Electr. Electron. Eng.*, vol. 37, no. 1, pp. 98–117, Jan. 2018.

[24] P. Jia, S. Duan, and J. Yan, "An enhanced quantum-behaved particle swarm optimization based on a novel computing way of local attractor," *Information*, vol. 6, no. 4, pp. 633–649, 2015.

[25] O. U. Rehman, S. Tu, S. Khan, H. Khan, and S. Yang, "A modified quantum particle swarm optimizer applied to optimization design of electromagnetic devices," *Int. J. Appl. Electromagn. Mech.*, vol. 58, no. 3, pp. 347–357, Nov. 2018.

[26] Y. Fu, M. Ding, C. Zhou, and H. Hu, "Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 6, pp. 1451–1465, Nov. 2013.

[27] M. Tokgo and R. Li, "Estimation method for path planning parameter based on a modified QPSO algorithm," in *Proc. Int. Conf. Artif. Intell., Methodol., Syst., Appl.*, 2014, pp. 261–269.

[28] T. Xue, R. Li, M. Tokgo, J. Ri, and G. Han, "Trajectory planning for autonomous mobile robot using a hybrid improved QPSO algorithm," *Soft Comput.*, vol. 21, no. 9, pp. 2421–2437, May 2017.

[29] O. U. Rehman, S. U. Rehman, S. Tu, S. Khan, M. Waqas, and S. Yang, "A quantum particle swarm optimization method with fitness selection methodology for electromagnetic inverse problems," *IEEE Access*, vol. 6, pp. 63155–63163, 2018.

[30] A. Jarvis, H. I. Reuter, A. Nelson, and E Guevara. (2008). *Hole-Filled Seamless SRTM data V4, International Centre for Tropical Agriculture (CIAT)*. [Online]. Available: http://srtm.csi.cgiar.org

[31] Y. Fu, M. Ding, C. Zhou, C. Cai, and Y. Sun, "Path planning for UAV based on quantum-behaved particle swarm optimization," *Proc. SPIE*, vol. 7497, Oct. 2009, Art. no. 74970B.

[32] M. Clerc and J. Kennedy, "The particle swarm–explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, 1st Quart., 2002.

[33] K. Y. Kok and P. Rajendran, "Differential-evolution control parameter optimization for unmanned aerial vehicle path planning," *PLoS ONE*, vol. 11, no. 3, 2016, Art. no. e0150558.
[34] S. He, Q. H. Wu, J. Y. Wen, J. R. Saunders, and R. C. Paton, "A particle swarm optimizer with passive congregation," *Biosystems*, vol. 78, nos. 1–3, pp. 135–147, Dec. 2004.

CHEN HUANG received the Ph.D. degree in mechanical engineering from Dalian Jiaotong University, China, in 2018. She is currently a Lecturer with the School of Civil Aviation, Shenyang Aerospace University. Her research interest includes intelligence systems and path planning.

JIYOU FEI received the Ph.D. degree from Xian Jiaotong University, China, in 2003. He is currently a Professor with the School of Vehicle Engineering, Dalian Jiaotong University. His research interests include automatic control and advanced measurement.

WU DENG received the Ph.D. degree in computer application technology from Dalian Maritime University, China, in 2012. Since 2019, he has been a Professor with the College of Electronic Information and Automation, Civil Aviation University of China. His research interest includes artificial intelligence, optimization method, and fault diagnosis.