HAL
open science

Potential Odor Intensity Grid Based UAV Path Planning Algorithm with Particle Swarm Optimization Approach

Yang Liu, Xuejun Zhang, Xiangmin Guan, Daniel Delahaye

To cite this version:

Yang Liu, Xuejun Zhang, Xiangmin Guan, Daniel Delahaye. Potential Odor Intensity Grid Based UAV Path Planning Algorithm with Particle Swarm Optimization Approach. Mathematical Problems in Engineering, 2016, 2016, 10.1155/2016/7802798 . hal-01366808

HAL Id: hal-01366808

https://enac.hal.science/hal-01366808

Submitted on 15 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 7802798, 16 pages
http://dx.doi.org/10.1155/2016/7802798

Research Article

Potential Odor Intensity Grid Based UAV Path Planning Algorithm with Particle Swarm Optimization Approach

Yang Liu,[1,2] Xuejun Zhang,[1] Xiangmin Guan,[3] and Daniel Delahaye[4]

[1] *School of Electronic & Information Engineering, Beihang University, Beijing, China*
[2] *School of Information Science & Electric Engineering, Shandong Jiaotong University, Jinan, China*
[3] *School of General Aviation, Civil Aviation Management Institute of China, Beijing, China*
[4] *MAIAA Laboratory, Ecole Nationale de l'Aviation Civile, Toulouse, France*

Correspondence should be addressed to Xuejun Zhang; zhxj@buaa.edu.cn

Received 18 March 2016; Accepted 21 August 2016

Academic Editor: Mauro Pontani

Copyright © 2016 Yang Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a potential odor intensity grid based optimization approach for unmanned aerial vehicle (UAV) path planning with particle swarm optimization (PSO) technique. Odor intensity is created to color the area in the searching space with highest probability where candidate particles may locate. A potential grid construction operator is designed for standard PSO based on different levels of odor intensity. The potential grid construction operator generates two potential location grids with highest odor intensity. Then the middle point will be seen as the final position in current particle dimension. The global optimum solution will be solved as the average. In addition, solution boundaries of searching space in each particle dimension are restricted based on properties of threats in the flying field to avoid prematurity. Objective function is redesigned by taking minimum direction angle to destination into account and a sampling method is introduced. A paired samples t-test is made and an index called straight line rate (SLR) is used to evaluate the length of planned path. Experiments are made with other three heuristic evolutionary algorithms. The results demonstrate that the proposed method is capable of generating higher quality paths efficiently for UAV than any other tested optimization techniques.

1. Introduction

Unmanned aerial vehicle (UAV) is a kind of aircraft without onboard pilots that can be remotely controlled or fly autonomously based on the preplanned flying routes, increasingly suitable for a real-world environment [1–4]. Currently, UAV has been widely used in civil and military fields, such as aerial photography, search and rescue tasks, geophysical survey, environmental and meteorological monitoring, surveillance, reconnaissance, high-risk target penetration, suppressing enemy air defense, deep target attacking, and dominating the battle space [5–7]. But how to fly safely in all the fields is vital for mission effectiveness, in which path planning is one of the most important technologies for autonomous flight of UAV.

UAV path planning is a key aspect of the autonomous control module, whose mission is to provide an optimal flying path from the starting point to the desired destination avoiding artificial threats and some natural terrain constraints with least cost and shortest length of flying distance. Usually the flying path of UAV is automatically provided by a path planner based on an objective function [8, 9] considering all the constraints. Planning intuitive flying routes for UAV in large real-world scenarios and in the presence of obstacles is more complicated due to the mostly open structure of the airspace. Particularly in the near future UAV is to be integrated into national airspace system (NAS).

In most situations UAV path planning is often formulated as a global optimization problem, in which the feasibility of the candidate path depends on the mission, environment, and UAV physical constraints. Although the applications of UAV are so different, the optimality of a feasible path for any of them can be defined by different optimization planning criteria (such as minimal flying time and/or path length) and fulfillment of some mission constraints (such as flying at a given altitude or visiting some points) [10]. Besides, the

physical characteristics of the UAV and the environment also restrict the feasibility of any path and should be considered by a realistic path planning problem.

In the past few years several path planning algorithms have been proposed, which can be mainly divided into two categories, graph-based and population-based evolutionary algorithms. The former includes Voronoi diagram search method [11], mathematical programming method [12], A^* searching algorithm [13, 14], D^* lite algorithm [15], and bilevel programming method [16]. In these algorithms Eppstein's k-best paths algorithm [17] is used to find an optimal path for UAV. The biggest deficiency of graph-based ones is that it is difficult to combine the motion constraints of UAV itself, which means it usually cannot be used in practical situations. UAV self-performance is an absolutely necessary factor needed to be taken into account when designing a path planning algorithm. Another important category is the population-based optimization algorithms. They could make UAV flying routes generated by reducing the complexity and dimensions, which is a NP-hard problem. These algorithms mainly include genetic algorithm (GA) [18], particle swarm optimization (PSO) [10], ant colony optimization (ACO) [19], artificial bee colony (ABC) [20], differential evolution (DE) [21], gravitational search algorithm (GSA) [22], intelligent water drops optimization (IWD) [23], memetic computing method [24], and firefly algorithm (FA) [25]. Each of these categories has its own advantages over others in certain aspects. However, among them, GA, PSO, and FA are the most three popular for their simplicity and effectiveness, which are becoming the hottest research topics and are most suitable for solving the global optimization problems with large scales. For example, GA is famous for the ease of implementation for both continuous and discrete problems. There are no extra requirements for the continuity in response functions and could be used efficiently with large numbers of parallel processors. The generated global or near global solutions are more robust. And FA is recently developed to solve nonlinear design problems. In FA all fireflies are unisexual and any individual firefly will be attracted to others based on their higher brightness. But their brightness decreases as their mutual distance increases. By iterations of brightness oriented movement, the global optimal solutions could be found finally.

Another important one is called PSO, which is short for particle swarm optimization. It is an evolutionary computation algorithm first proposed by Kennedy and Eberhart [26], which is designed based on the study of the social behavior of bird flocking and fish schooling. Each particle adjusts its flying positions in the searching space in terms of its own flying experience and the whole swarm flying experiences. Successful applications of PSO in neural network training, function optimization, and fuzzy system control have demonstrated that PSO is a promising and efficient optimization method. So far many significant improvements are proposed to improve the performance of standard PSO algorithm. An excellent overview of the basic concepts of PSO and its variants can be found in [27]. PSO has been seen as an attractive optimization tool for the advantages of simple implementation procedure, good performance, and fast convergence speed. However, it has been shown that this method is easily trapped into local optima when coping with complicated problems.

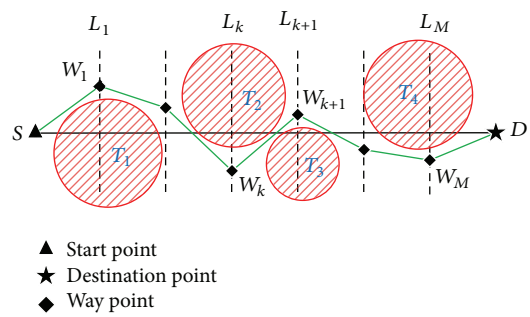


Figure 1: Typical UAV path planning model.

In this paper, a potential odor intensity grid based UAV path planning algorithm is proposed combining standard PSO technique. By identifying the different levels of odor intensity, a potential grid construction operator is designed for standard PSO, which is implemented easily and avoids local optima and slow convergence. The potential grid construction operator indicates two potential location grids with highest odor intensity, including centers of the two grids and the preset side length. Then middle point of the two grid centers will be seen as the final global position in current particle dimension space of current iteration. The global optimum solution will be solved as the average. Experimental results demonstrate that the proposed method is capable of generating higher quality paths efficiently for UAV than any other tested optimization algorithms.

The remainder of this paper is organized as follows. Path planning problem description is provided in Section 2. Concise standard PSO algorithm and explicit realization of proposed method are given in Section 3, respectively. Section 4 presents the settings in application conditions of the proposed algorithm and experimental results of global route planning are listed. Finally, the paper is concluded in Section 5.

2. Problem Formation

UAV path planning can be seen as a global optimization problem to generate a serial of way points from the start to the destination with least cost values. Terrain and threat modeling and the design of objective function become two key problems in route generation for UAV.

2.1. Terrain and Threat Modeling. As shown in Figure 1 the mission of path planning is to generate a feasible flying route from the start point S to destination point D without flying out of the designated map or being taken down by threats, as well as least flying cost. There are some threat areas locating in the map, such as radars, missiles, and artillery, which are all denoted as red circles in Figure 1. Once the UAV is in the coverage of any threat, it will be vulnerable to the threat with a certain probability proportional to the distance away from the threat centers. Oppositely there will not be any danger if

UAV is flying out of the threat covered region. Considering the threat areas and other affecting factors, how to get to the desired destination safely is the key problem.

2.1.1. Terrain Restriction. In real situations all UAVs will have to fly in a specified region for different tasks, all of which are limited in power supply, flying distance, and flying time. Any UAV flying out of the region will lead to high risk out of control and crashing for the use up of energy. In this way any way point generated by the path planning algorithms should be in the flying region. If out, it will be punished. The following equation is given to record the distances to the coverage boundary of all way points:

$$J_{\text{out}} = \begin{cases} \sum_i \text{Dis}(i) & \text{if } (x_i, y_i)^T \notin R \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Here in (1), $(x_i, y_i)^T$ is the current coordinate of way point i, R is the designated flying region, and Dis(i) is the distance from current way point i to the nearest boundary of designated flying region R. In most situations the flying area will be set as a rectangle, in which start and destination locate.

2.1.2. Threat Modeling. There are two kinds of threats: one is random and the other is deterministic. For random situations, path planning algorithm could generate paths by refreshing itself to avoid the pop-up threats if real time information could be obtained. In this paper deterministic threat areas are taken into account, all of which are known in advance as shown in Figure 1.

It is desirable that all the generated UAV way points are kept away from the threat as far as possible and the farther the better. For this purpose, a new model for the threatening space definition is developed as follows. First let the vector $\mathbf{P}^i_{\text{threat}} = [x^i_{\text{threat}}, y^i_{\text{threat}}, r^i_{\text{threat}}]^T$ be assigned to the threat i with its center coordinate $(x^i_{\text{threat}}, y^i_{\text{threat}})^T$ and destruction range r^i_{threat}. An effective destruction gain G_i is considered for each threat i, which is different for different threats. For any UAV way point j with current coordinate $(x_j, y_j)^T$ we can define the effects of all threats on the planned path as follows:

$$J_{\text{expo}} = \begin{cases} \sum_j \sum_i \dfrac{G_i \left[\left(r^i_{\text{threat}}\right)^2 - \left(r_{ji}\right)^2 \right]}{\left(r^i_{\text{threat}}\right)^2} & \text{if } r_{ji} < r^i_{\text{threat}} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Here r_{ji} is the distance from current position of UAV way point j to the center of threat i which can be written as

$$r_{ji} = \sqrt{\left(x_j - x^i_{\text{threat}}\right)^2 - \left(y_j - y^i_{\text{threat}}\right)^2}. \tag{3}$$

Equation (2) illustrates that once UAV is flying in the covered areas of any threat the probability to be detected by radar or brought down by missiles will be high.

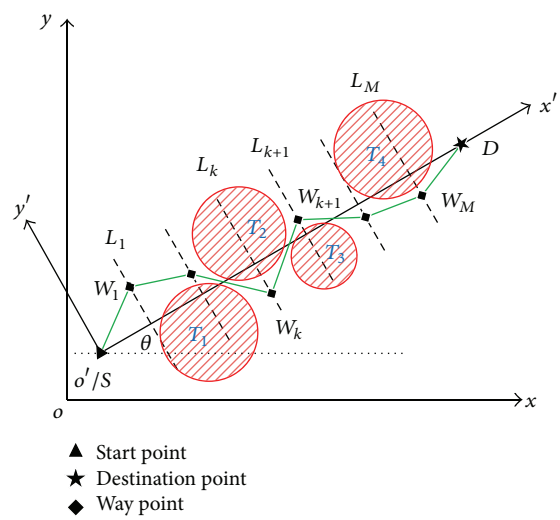


Figure 2: Transformation of coordinate system.

2.2. Objective Function of UAV Route Optimization

2.2.1. Problem Analysis. The task of path planning is to find a feasible flying route from the start point to destination point with least flying cost and avoiding entering threat areas [28], as the green line does in Figure 1. Here the straight line SD is divided into $(M + 1)$ segments by M vertical lines L_k $(k = 1, 2, \ldots, M)$. In this way the flying route can be denoted by M way points, namely, W_k $(k = 1, 2, \ldots, M)$, on each vertical line L_k $(k = 1, 2, \ldots, M)$. The core difficulty is how to determine these way points with least global flying cost and also without entering covered areas by threats.

In order to simplify the computation process and accelerate the searching speed of global solutions in the path planning algorithm, the transformation of coordinate system is made [29] as shown in Figure 2, in which the new coordinate system $x'o'y'$ is transformed from origin coordinate system xoy by setting the straight line $\overrightarrow{SD}$ as the new x axis and the start point S of UAV as the new origin of coordinate. The angle between $o'x'$ and ox is θ, which means the origin x-axis ox will be rotated anticlockwise with angle θ.

Suppose a way point in the original coordinate system with coordinate $(x_O, y_O)^T$ and correspondingly the transformed coordinate in the new coordinate system is $(x_T, y_T)^T$, and the relationship between the coordinates in two coordinate systems of the same way point can be written using the following equation:

$$\begin{bmatrix} x_T \\ y_T \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_O - x_S \\ y_O - y_S \end{bmatrix}. \tag{4}$$

In (4) $(x_S, y_S)^T$ is the coordinate of start point in original coordinate system. Once the transformation is completed the desired UAV flying path can be denoted as a sequence of way point W_k from the start point S to the destination point D, namely, $\{S, W_1, \ldots, W_M, D\}$. Combining Figures 1 and 2, the coordinates of way points S and D in $x'o'y'$ can be easily

known as $(0,0)^T$ and $(|SD|,0)^T$, respectively. In this way other abscissas of any way point W_k can be obtained using

$$x_k = \frac{|SD|}{M+1}k. \tag{5}$$

So the path planning problem becomes the optimization problem of longitudinal coordinates for any UAV way point W_k with least global minimum cost value. How to design objective function to compute longitudinal coordinate with global optimum will be given in the following.

2.2.2. Objective Function Design. The objective function is to evaluate a candidate path which is generated by path planning algorithms. Many factors should be taken into account, such as cost of the path, the performance of UAVs themselves, and the mission constraints. In our proposed method the following factors are considered.

(A) Cost of the Generated Path. As described before the way points out of threat areas should be chosen. Otherwise they will be penalized which lead to high cost of the whole path. Here how to evaluate the cost of the generated UAV path is given. In our proposed algorithm a sampling method for each path segment is proposed to better describe the quality of possible UAV path.

As described in the section before, after the coordinate system is transformed each path of UAV consisted of many way points W_i $(i = 1,2,\ldots,M)$ and $(M+1)$ segments are formed. Suppose any path segment $\overrightarrow{W_iW_{i+1}}$ as shown in Figure 3 is formed by two way points W_i and W_{i+1}, which locate on vertical lines L_k and L_{k+1}, respectively. The sampling method for any possible UAV path segment $\overrightarrow{W_iW_{i+1}}$ is as follows.

First any path segment $\overrightarrow{W_iW_{i+1}}$ is detected whether it falls into the areas covered by threats. If so sample points are set on the segment $\overrightarrow{W_iW_{i+1}}$ with the total number of L. Then the rules of these sample points can be built using

$$\begin{aligned} 2\left|W_iP_{1/L}\right| &= \left|P_{1/L}P_{2/L}\right| = \cdots = \left|P_{(L-1)/L}P_{L/L}\right| \\ &= 2\left|P_{L/L}W_{i+1}\right|. \end{aligned} \tag{6}$$

After the L sample points are determined the cost of the segment $\overrightarrow{W_iW_{i+1}}$ will be calculated as follows:

$$\begin{aligned} &J_T(i) \\ &= \frac{\left|\overrightarrow{W_iW_{i+1}}\right|}{L}\sum_{j=1}^{N_T} G_j\left(\frac{1}{d_{1/L,ij}} + \frac{1}{d_{2/L,ij}} + \cdots + \frac{1}{d_{L/L,ij}}\right). \end{aligned} \tag{7}$$

In (7) $|\overrightarrow{W_iW_{i+1}}|$ is the distance of segment $\overrightarrow{W_iW_{i+1}}$, L is the total number of sample points in this segment $\overrightarrow{W_iW_{i+1}}$, and G_j is the destruction gain of threat j. $d_{m/L,ij}$ denotes the distance of the mth sample point that is on segment i to the center of threat j and $m = 1,2,\ldots,L$. Obviously the number of threats that could affect this path segment $\overrightarrow{W_iW_{i+1}}$ is set as N_T.

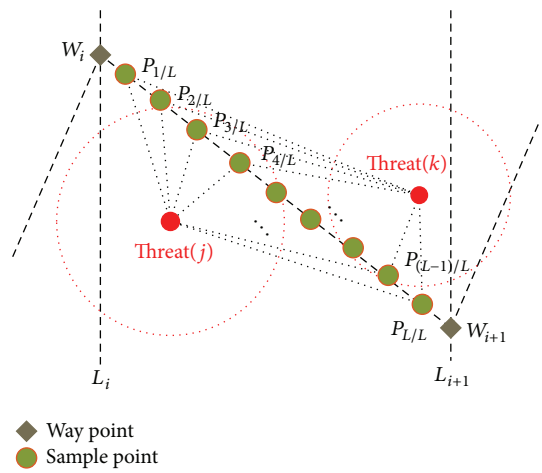


FIGURE 3: Sample method for UAV path segment $\overrightarrow{W_iW_{i+1}}$.

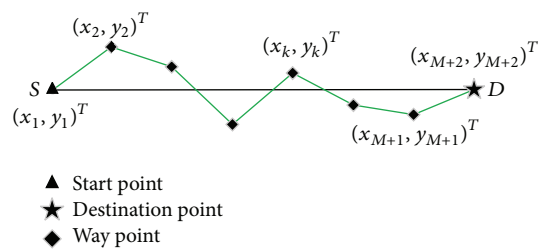


FIGURE 4: Definitions of straight line rate (SLR).

The total effects of threat areas on one planned flying path can be calculated by adding all the costs of path segments on the flying path by using (6) and (7).

$$J_{\text{path}} = \sum_i J_T(i). \tag{8}$$

(B) Total Flying Distance for UAV. Flying distance is another important factor that needs to be taken into account. In realistic operations the generated flying path should be as short as possible from the start to the destination point with all limitations satisfied. Short path length means high efficiency which saves energy and time for UAV and also a lower chance of being detected by unknown threats. In the proposed method an index called straight line rate (SLR) is defined to indicate performance of the total length of planned UAV flying route for simplicity. Suppose the coordinates of start point S and destination point D as $(x_1, y_1)^T$ and $(x_{M+2}, y_{M+2})^T$ showed in Figure 4.

As shown in Figure 4, M is the number of way points on the planned UAV path with coordinate of way point W_k as $(x_k, y_k)^T$. Straight line rate (SLR) can be obtained as follows:

$$J_{\text{SLR}} = \frac{\sum_{i=1}^{M+1}\sqrt{(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2}}{|SD|}. \tag{9}$$

In (9) $|SD| = \sqrt{(x_{M+2}-x_1)^2 + (y_{M+2}-y_1)^2}$ is the Euclidean distance from the start to the destination.

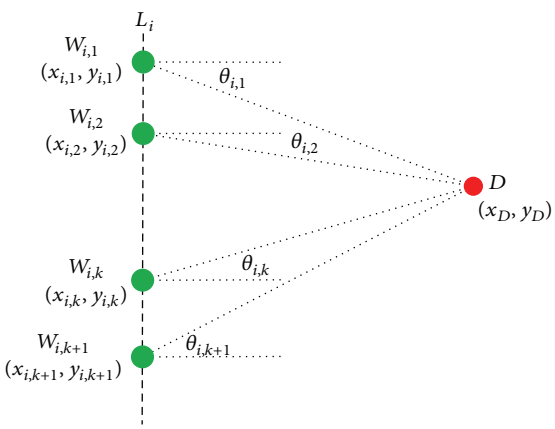


FIGURE 5: Definition of direction angle.

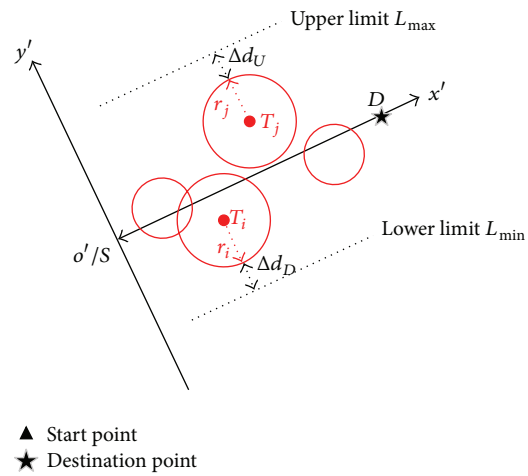


FIGURE 6: Optimized searching scope.

(C) Minimum Direction Angle. As shown in Figure 5, there will be many candidate way points $(W_{i,1}, W_{i,2}, \ldots)$ in the searching space which is denoted as L_i. In order to improve the searching speed and avoid falling into local optimum, direction angle of different way points $W_{i,k}$ on vertical line L_i in the searching space is defined as $\theta_{i,k}$. They are the angles between current way point and destination point which is denoted as Figure 5 shows. Obviously the way point on vertical line L_i with smaller direction angle is better than with larger ones. This parameter can evaluate the performances of generated UAV path as part of objective function.

Equation (10) combines effects from all the direction angles of possible way points on different vertical lines meaning different searching space.

$$J_{\mathrm{DA}} = \sum_i \min_k \{\theta_{i,k}\} = \sum_i \min_k \left\{\arctan \frac{|y_D - y_{i,k}|}{|x_D - x_{i,k}|}\right\}. \quad (10)$$

(x_D, y_D) is the coordinate of destination and $(x_{i,k}, y_{i,k})$ is the candidate way point position on vertical line L_i.

(D) Optimized Searching Scope. To further make the searching process of optimum way point simplified, the searching scope of each particle in the solution space is optimized [30], which is shown in Figure 6. The red circle is the threat in the UAV flying field with center T_i and radius r_i. The searching space is limited by upper limit $L_{\max}$ and lower limit $L_{\min}$. The margins for upper limit and lower limit are Δd_U and Δd_D, respectively. The searching is realized in this constrained area. Any way points out of this scope will be penalized. The two limits could be calculated as follows:

$$\begin{aligned} L_{\max} &= \max_i \{y_{\mathrm{threat}}(i) + r_{\mathrm{threat}}(i)\} + \Delta d_U, \\ L_{\min} &= \min_i \{y_{\mathrm{threat}}(i) - r_{\mathrm{threat}}(i)\} - \Delta d_D. \end{aligned} \quad (11)$$

In (11) $y_{\mathrm{threat}}(i)$ and $r_{\mathrm{threat}}(i)$ are the longitudinal coordinate and radius of threat i, respectively. In this way the cost of any UAV way points out of the limited searching space could be given in the following:

$$J_{\mathrm{out}} = \begin{cases} \sum_{i=1}^{M} [y_U(i) - L_{\max}] & \text{if } y_U(i) > L_{\max} \\ 0 & \text{otherwise} \\ \sum_{i=1}^{M} [L_{\min} - y_U(i)] & \text{if } y_U(i) < L_{\min}. \end{cases} \quad (12)$$

In the equation above M is the number of total way points on the generated UAV path. And $y_U(i)$ is the longitudinal coordinate of the ith way point.

Based on the analysis above the complete objective function can be given as follows, to realize the evaluation of one candidate UAV flying route and UAV self-performance:

$$J_{\mathrm{obj}} = \alpha J_{\mathrm{path}} + \beta J_{\mathrm{SLR}} + \chi J_{\mathrm{DA}} + \delta J_{\mathrm{out}}. \quad (13)$$

Here in the equation α, β, χ, and δ are four weighting parameters to assess the different effects of different elements on the objective function. The value of the four parameters is between $[0, 1]$.

3. Realization of Proposed Algorithm

In this section the proposed method for UAV path planning in two-dimensional space is given explicitly based on the standard particle swarm optimization (PSO). The standard PSO is described first.

3.1. Standard PSO. The standard particle swarm optimization (PSO) is a population-based nondeterministic optimization method which was first proposed by Kennedy and Eberhart [26]. It simulates the movement of a swarm of particles in a multidimensional searching space towards a global optimal solution by repeated iterations. The position of each particle in each iteration represents a candidate solution which is initialized randomly. In each iteration the current particle velocity is renewed based on the previous velocity of the

```
Standard PSO Algorithm
Set the generation counter k = 0
/*Initialization*/ Generate S individuals x_i of D dimensions with random initial location and speed in searching space.
/*Main loop*/
while termination criteria is not satisfied do
   generation counter k = k + 1
   for i = 1 : S do
   /*Computation*/ Calculate cost value of particle x_i using its current position in current generation.
   /*Update*/ Update the location and speed of particle x_i of current generation based on formula (17).
   /*Update*/ Update local best position P_i,best and global best position G_best based on current cost value.
   end for
end while
```

Algorithm 1: Pseudocode of standard PSO.

particle, the best position ever occupied by the particle (personal influence), and the best position ever occupied by any particle of the swarm (social influence). Then position could be renewed using the new updated velocity. The mathematical descriptions are listed in the following.

Suppose the size of the swarm is S and the dimension of each particle i is D. There are two parameters for each particle i, namely, position x_i and velocity v_i, whose dimension is the same as D obviously. And D also stands for the dimension of the problem to be solved. The two of each particle i can be written as a particle vector in the following equation:

$$(\mathbf{x_i},\mathbf{v_i}) = ((x_{i1}, x_{i2}, \ldots, x_{iD}), (v_{i1}, v_{i2}, \ldots, v_{iD})). \quad (14)$$

In this way a particle swarm with swarm size S can be written as a matrix made up of S particle vectors.

$$\{(\mathbf{x_1},\mathbf{v_1}), (\mathbf{x_2},\mathbf{v_2}), \ldots, (\mathbf{x_S},\mathbf{v_S})\}. \quad (15)$$

Then cost value of each particle in each iteration with current position could be calculated using specified objective function for further steps. There are two kinds of cost values in standard PSO, local best value $\mathbf{P_{i,best}}$ of each particle i and one global best value $\mathbf{G_{best}}$ of all particles, which can be written as (16). The number of local best values is the same as swarm size S.

$$\begin{aligned} \mathbf{P_{i,best}} &= (p_{i1,\text{best}}, p_{i2,\text{best}}, \ldots, p_{iD,\text{best}}), \\ \mathbf{G_{best}} &= (g_{1,\text{best}}, g_{2,\text{best}}, \ldots, g_{D,\text{best}}). \end{aligned} \quad (16)$$

Once the two values are obtained the position and velocity of each particle in each dimension are updated by keeping track of the two best positions, using the following equations:

$$\begin{aligned} v_{ij}^{k+1} &= wv_{ij}^{k} + c_1\xi\left(p_{ij,\text{best}}^{k} - x_{ij}^{k}\right) + c_2\eta\left(g_{j,\text{best}}^{k} - x_{ij}^{k}\right) \\ x_{ij}^{k+1} &= x_{ij}^{k} + rv_{ij}^{k+1} \\ & i = 1, 2, \ldots, S \;\; j = 1, 2, \ldots, D \;\; k = 1, 2, \ldots, N-1. \end{aligned} \quad (17)$$

In (17) w is inertia weight, which reflects the impacts of the particle velocity in previous iteration on its current iteration. ξ and η are random numbers between 0 and 1. c_1 and c_2 are positive constants, named self-cognition and social knowledge, which stand for the inheriting abilities from particle itself and the whole swarm, respectively. r is constant factor used to constrain the position updating. And N is the total iterations that the algorithm has to run. In most situations it is also set as termination criteria of PSO. The standard PSO will not stop until the termination criteria are satisfied. Pseudocode of standard PSO is given in Algorithm 1.

3.2. Proposed Algorithm. In this section the potential grid construction operator based on odor intensity is presented first and then the realization of the proposed method is described.

3.2.1. Potential Grid Construction Operator. The standard particle swarm optimization technique falls into local optimum easily and sometimes it is hard to converge with low speed. Even in extreme situations there are no optimal solutions which makes it fail to solve the UAV path planning problem. In our proposed algorithm potential grid is constructed based on odor intensity of particles to get over the defects of standard PSO, which is called potential grid construction operator here.

As described before the core step of UAV path planning is to find way points on different particle dimensions with least global cost values as well as avoid all the threats. For each dimension of each particle, it can be seen as solution space in which the global solutions may exist. In our model it is supposed that there are S particles denoted as P_i ($i = 1, 2, \ldots, S$) with the number of dimensions D. Each dimension of particle P_i can be written as P_{ij} with $j = 1, 2, \ldots, D$. Each particle will leave odor pheromone trails on the position in each dimension where it stayed and all the odor pheromone trails will be accumulated to form different odor intensities. Potential grid construction is given in Figure 7 in the solution space, which is based on odor intensities.

In Figure 7 each blank square with the same side length d_{ij} denotes a candidate position in an iteration in the jth dimension space of particle i. As shown in Figure 7 there will

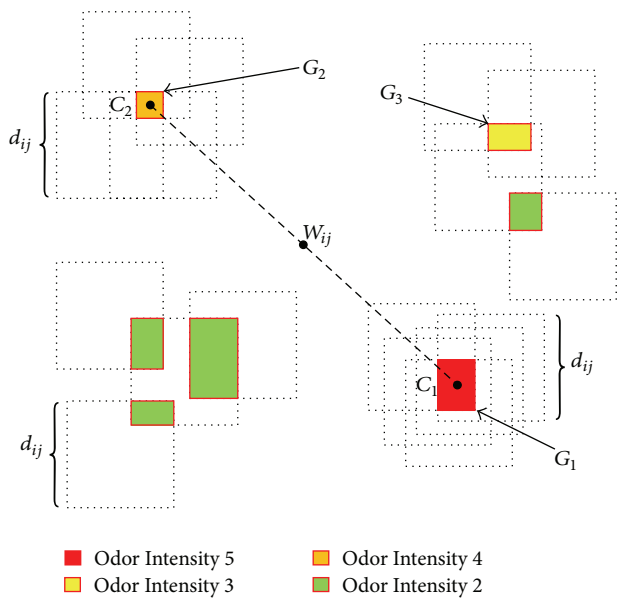


FIGURE 7: Potential grid construction in the jth dimension of particle i.

be odor pheromone trails filled in each square evenly. The odor trails from different particles will be overlay and accumulated. Subsequently different levels of odor intensities are formed. Potential grid construction based on odor intensity will be implemented in each P_{ij} as shown in Figure 7 with the following steps.

Step 1 (particle position square generation). For the kth generation in the jth dimension of particle i, there will be a possible particle location written as $L_{ij}(k)$. A particle position square $B_{ij}(k)$ is generated by setting $L_{ij}(k)$ as center and d_{ij} as the side length, which can be written as the following equation. Also odor pheromone trails will be filled in this position square $B_{ij}(k)$.

$$B_{ij}(k): \begin{cases} \text{Center: } L_{ij}(k) \\ \text{Side length: } d_{ij}. \end{cases} \tag{18}$$

Step 2 (odor accumulation). The position that a particle stays will leave odor pheromone trails in the particle position square covered area and will be accumulated. After all the N iterations are completed there will be some areas with higher odor intensity, as the different colorful grids, which is shown in Figure 7. As given in Figure 7 there are four odor intensities, namely, Odor Intensity 5, Odor Intensity 4, Odor Intensity 3, and Odor Intensity 2. These potential grids G_l, which stand for potential positions, can be obtained using the following equation:

$$\{G_1, G_2, \ldots\} = B_{ij}(1) \cap B_{ij}(2) \cap \cdots \cap B_{ij}(N). \tag{19}$$

Different odor intensities are colored in the end.

Step 3 (candidate way point generation). Once odor accumulation is done two of potential grids with the biggest odor intensities are formed. The centers of the two grids are denoted as c_1 and c_2, respectively. So the particle position W_{ij} for the jth dimension of particle i can be calculated using

$$W_{ij} = \frac{(c_1 + c_2)}{2}. \tag{20}$$

Step 4 (final particle position resolution). After the three steps above the final particle position L_j for each dimension j can be resolved for each dimension using

$$L_j = \frac{\sum_{i=1}^{S} W_{ij}}{S}. \tag{21}$$

So once the four steps are realized all the way points that a UAV can fly along with avoiding all threat areas and terrain obstacles can be written as L_i $(i = 1, 2, \ldots, D)$.

3.2.2. Realization of Proposed Algorithm. Based on the aforementioned potential grid construction operator designed for standard PSO, all the elements required to build a complete path planning module for UAV are discussed in detail in this part.

Step 1 (path planning field formation). Two-dimensional flying field for UAV is formulated first, including the size of the area, starting point, and destination point designated in advance. Then terrain restriction and threat modeling are finished using (1) and (2), respectively. Threat modeling includes the position coordinates of threat center and the destruction scope they can affect individually. Another crucial step is to make coordinate system transformation using (4). All the calculations mentioned in the following will be finished under new coordinate system for simplicity.

Step 2 (particle swarm initialization). In the proposed algorithm there are S particles in a swarm, each of which has the same particle dimension D. The initial positions and velocities of particles are randomly assigned using (14) and (15). Initial cost values of each particle are calculated using objective function as shown in (13). Initial local best of personal particle $\mathbf{P_{i,best}}$ and global best $\mathbf{G_{bset}}$ are represented as (16).

Step 3 (start algorithm iterations). Once the initialization is finished algorithm iterations start. The termination criteria are set as the total iterations, which is also called generations, written as N. Positions and velocities in current generation are updated by (17) based on the initial cost values. Then based on the updated positions and velocities new cost values for each particle are calculated. New local best of personal particle $\mathbf{P_{i,best}}$ and global best $\mathbf{G_{bset}}$ of the swarm could be found out by comparing the minimum different cost values.

Step 4 (call potential grid construction operator for the first time). Equation (18) is used to call potential grid construction operator for the first time. The potential grid construction for different dimensions of different particles in current iteration is realized, which is stored in the dimension space. Possible

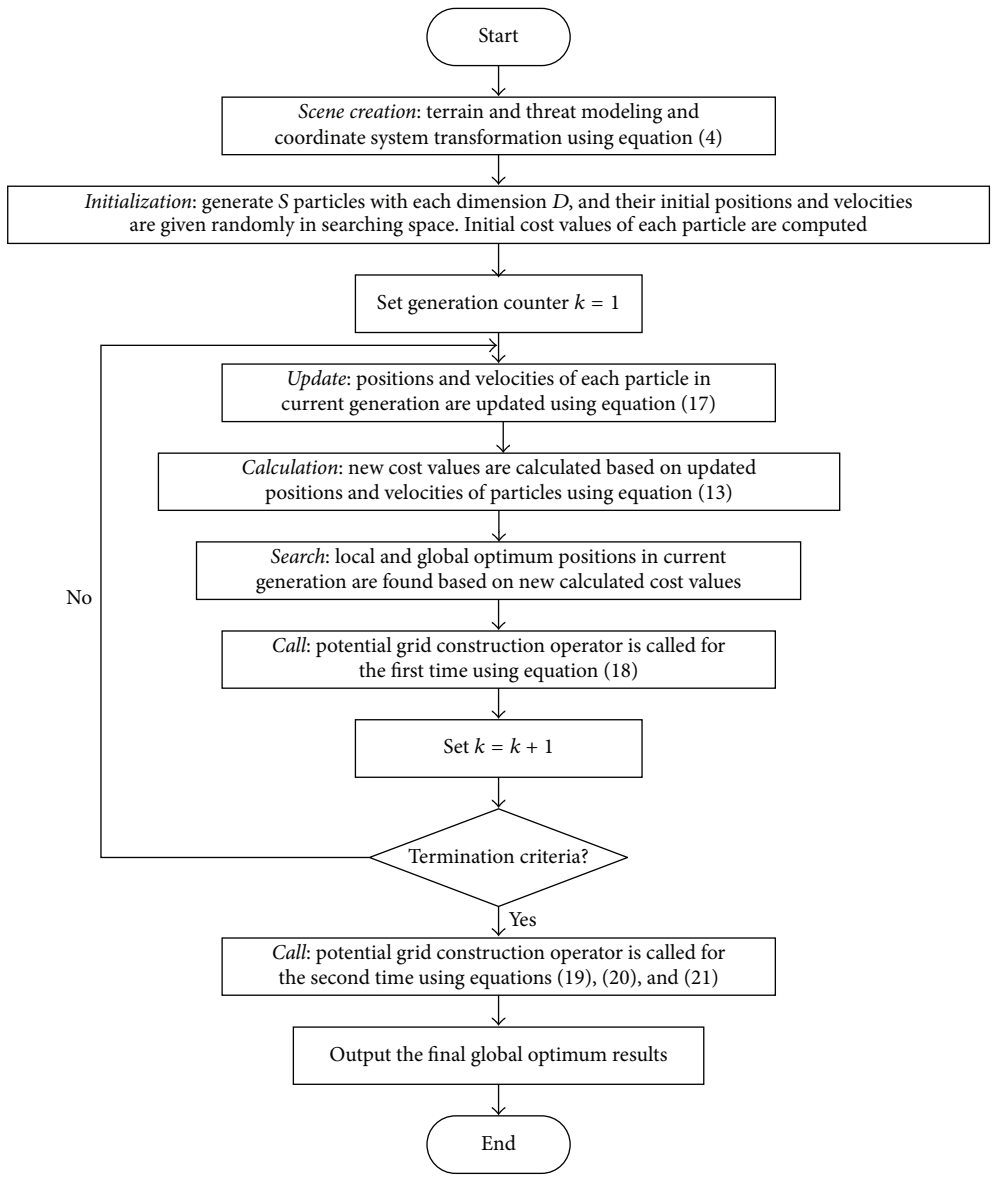


Figure 8: Flowchart of the proposed algorithm.

particle location is formed as a center of a rectangular, with a designated side length d_{ij}.

Step 5 (call potential grid construction operator for the second time). After all the iterations are stopped by the termination criteria, potential grid construction operator is called for the second time. Then (19), (20), and (21) are used to form the final global optimum way points for UAV, which are the best solutions.

After the steps above are finished global optimum way points for path planning will be output. The flowchart of the whole algorithm is given in Figure 8.

4. Experiment Validation and Comparison

To validate the effectiveness of the proposed algorithm in solving the two-dimensional path planning problem, experiments are conducted and also comparisons of our proposed method are made with stand PSO, GA, and FA with the same parameters settings.

4.1. Parameter Settings. The path planning field for UAV in the experiments is a square with side length 100 km; namely, the acreage of the whole covered area is 100 km × 100 km. There are some areas covered by many threats in the flying field, such as radars, missiles, and artilleries. Here two experiment scenarios are defined based on the degree of complexity. One is general scenario (*Scenario 1*) with five threats and the other is the complicated scenario (*Scenario 2*) with nine threats in the terrain, respectively. As mentioned before the destruction gains G_j of threat j are all set the same as grade 1; namely, $G_j = 1$. The explicit experimental parameters can be found in Table 1.

Besides external environment parameter settings there are some crucial parameters in the algorithm itself, which are

Table 1: Information of environment installations.

Scenario number	Number of iterations	Start point	Destination point	Threat center location	Threat radius (km)
General scenario: *Scenario 1*	400	[1, 1]	[95, 95]	[15, 25]	10
				[45, 25]	15
				[55, 58]	10
				[70, 82]	8
				[81, 58]	12
Complicated scenario: *Scenario 2*	400	[1, 1]	[95, 95]	[13, 15]	10
				[18, 48]	15
				[42, 15]	8
				[49, 45]	11
				[50, 80]	15
				[62, 31]	12
				[75, 75]	13
				[80, 15]	10
				[87, 50]	12

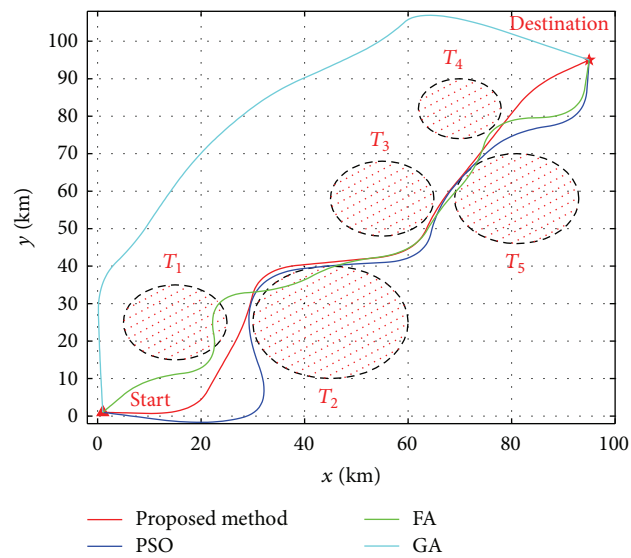


Figure 9: Planned path for UAV with $S = 100$ and $D = 10$ in *Scenario 1*.

size of the particle swarm, dimensions of a particle, number of iterations, the number (L) of sample points on path segment $\overrightarrow{W_iW_{i+1}}$, the size of constrained particle dimension searching limit Δd ($\Delta d = \Delta d_U = \Delta d_D$), and the side length ($d_{ij}$) of potential grid in the jth dimension of particle i. Another four weighting parameters α, β, χ, and δ which appeared in (13) are set with the same value of 1, which is written as $\alpha = \beta = \chi = \delta = 1$. Part of these crucial parameters in the algorithms are all set as in Table 2.

Three main indexes are listed as the performance indicator. The first is the planned UAV path in two-dimensional field. Whether the planned UAV path can avoid every threat successfully reflects the accuracy of the algorithm itself. The second is normalized cost values. The lower the cost is, the better the algorithm is. The third is the path length. Here in this paper, an index called straight line rate (SLR) is defined from (9), which could be rewritten as follows:

$$\mathrm{SLR} = \frac{\sum_{i=1}^{M+1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}{|SD|}. \tag{22}$$

And $(x_i, y_i)^T$ is the way point for UAV, in which the start and destination points are included. The four mentioned path planning algorithms will be implemented in the same situations with the same self-parameter settings. The results will be compared in the next part.

4.2. Result Comparisons. In this part performance comparisons under *Scenario 1* and *Scenario 2* are given first with different parameter settings. Then a paired samples t-test is made to analyze in a probability perspective. Self-performance comparisons are implemented under *Scenario 1* in the end.

4.2.1. Performance Comparisons under Scenario 1. Comparisons of four algorithms mentioned before are implemented in *Scenario 1* with the same parameter settings in this part. Two groups of experiment results are given with different swarm sizes. One group is set with swarm size $S = 100$ and particle dimension $D = 10$, and the other group is under $S = 400$ and $D = 10$, both of which are to explore the effects of different swarm sizes on the algorithms.

As shown in Figure 9 there are five independent threat areas in *Scenario 1*. The mission is to make UAV fly from start point to the destination point safely. It is obvious that PSO, GA, and our proposed algorithm could offer perfect flying route for UAV without entering any of these threats, which means they can solve the path planning problem with well solutions. However, FA fails to avoid all the threats. Parts of its planned trajectory fall into the areas covered by threats. As said before once a UAV is flying in the dangerous fields there will be a high risk of being detected or taken down. FA is the worst of all the four algorithms in solving this kind of route optimization problem in *Scenario 1*.

Another two indexes, normalized cost values and straight line rate (SLR), are given in Figures 10(a) and 10(b). In Figure 10(a) FA is the worst with the highest cost values from the first iteration to the end, which is mainly caused by falling into threats. But it is stable during the whole process since it finds a solution. PSO is the most stable and it could easily find the optimal flying path for UAV after the 120th iteration. But when all iterations are finished PSO is not the best, which means the optimal solution it finds is possibly local optimum. GA fluctuates fiercely from the beginning to the 300th iteration and then solutions could be found. Its solution is better than that of PSO after all the iterations finish. Our proposed algorithm presents a downtrend from the first to the last iteration, which reflects the process of optimizing. It fluctuates less than GA and convergence rate is also better. It can find out the global optimum instead of local optimum, which overcomes the defects of standard PSO. By calculations

Table 2: Parameters of algorithm itself.

Particle swarm size (S)	Particle dimension (D)	Number of sample points	Searching limit Δd	Side length of grid (d_{ij})
Variable	Variable	5	0	0.01

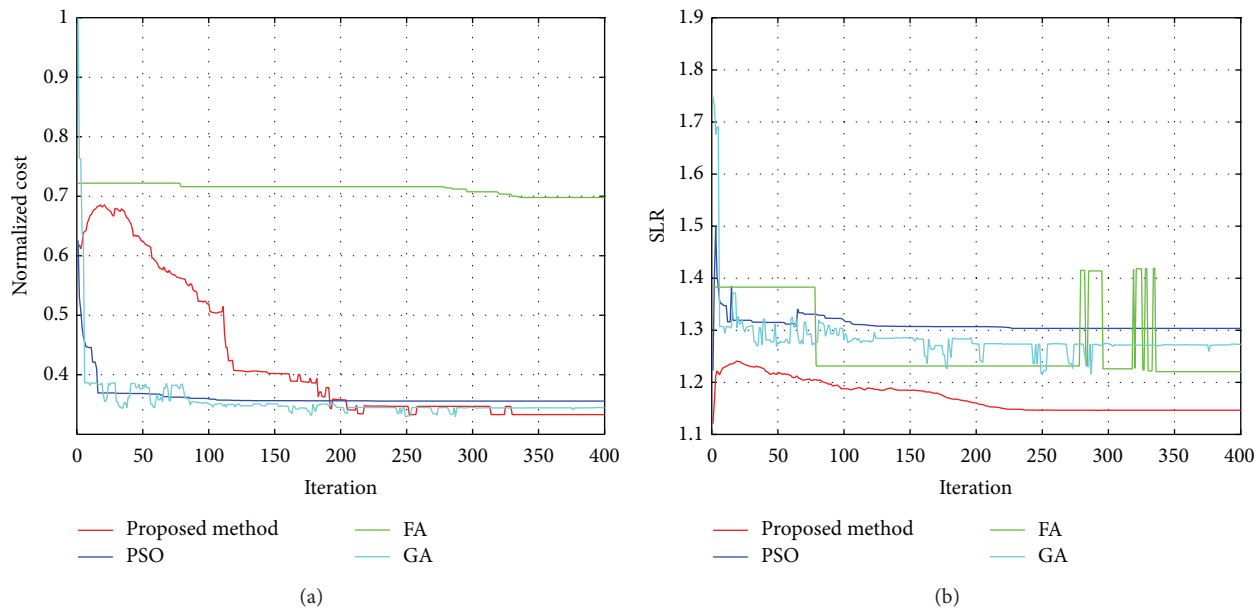


Figure 10: Performance comparisons of four algorithms with $S = 100$ and $D = 10$ in *Scenario 1*. (a) Normalized cost comparisons of different algorithms. (b) Path length comparisons of different algorithms.

Table 3: Comparisons with $S = 100$ and $D = 10$ in *Scenario 1*.

Category	Algorithm	Best	Worst	Mean	Std.
Cost value	Proposed	**0.4772**	**0.6245**	**0.4978**	0.4410
	PSO	0.5100	0.6256	0.5079	0.4659
	GA	0.4939	1	0.4990	1
	FA	1	0.7221	1	***0.1617***
Path length	Proposed	**149.0**	**165.0**	**155.9**	**0.2182**
	PSO	162.5	199.3	174.2	0.2244
	GA	161.6	232.6	170.6	0.7200
	FA	162.3	188.7	168.9	1

our proposed method is 6.37%, 52.28%, and 3.38% better compared with PSO, FA, and GA in cost values, respectively.

Figure 10(b) gives SLR comparisons of the planned path from four methods. The proposed one is the best option with shortest flying distance for UAV. Also it is the most stable from the beginning to the end. PSO is the worst with the longest flying route. GA and FA is between them. It is hard for the two to find the global optimum solutions which makes FA and GA produce a lot of fluctuations. The optimizing process is unstable at all. Our proposed method is 12.06%, 6.08%, and 9.95% shorter than PSO, FA, and GA, respectively, with the planned flying path.

Table 3 shows other indexes under different situations. There is no doubt that our proposed one is the best of all under all the indexes except the standard deviation of cost value. Oppositely FA is the best, which is because it could not find global best solutions and falls into local optimum at the beginning till the end which can be illustrated in Figure 10(a).

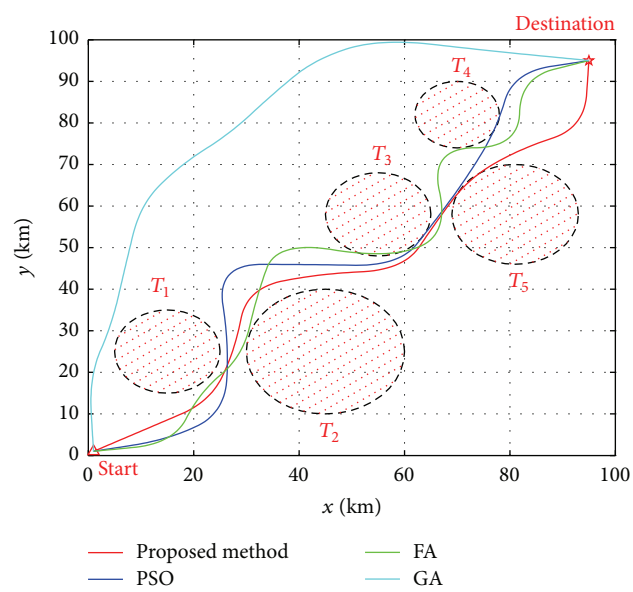


Figure 11: Planned path for UAV with $S = 400$ and $D = 10$ in *Scenario 1*.

In order to further explore the effects of different swarm sizes on the four algorithms, this parameter is increased to 400. So the planned path for UAV with $S = 400$ and $D = 10$ in *Scenario 1* is given in Figure 11. The biggest difference from

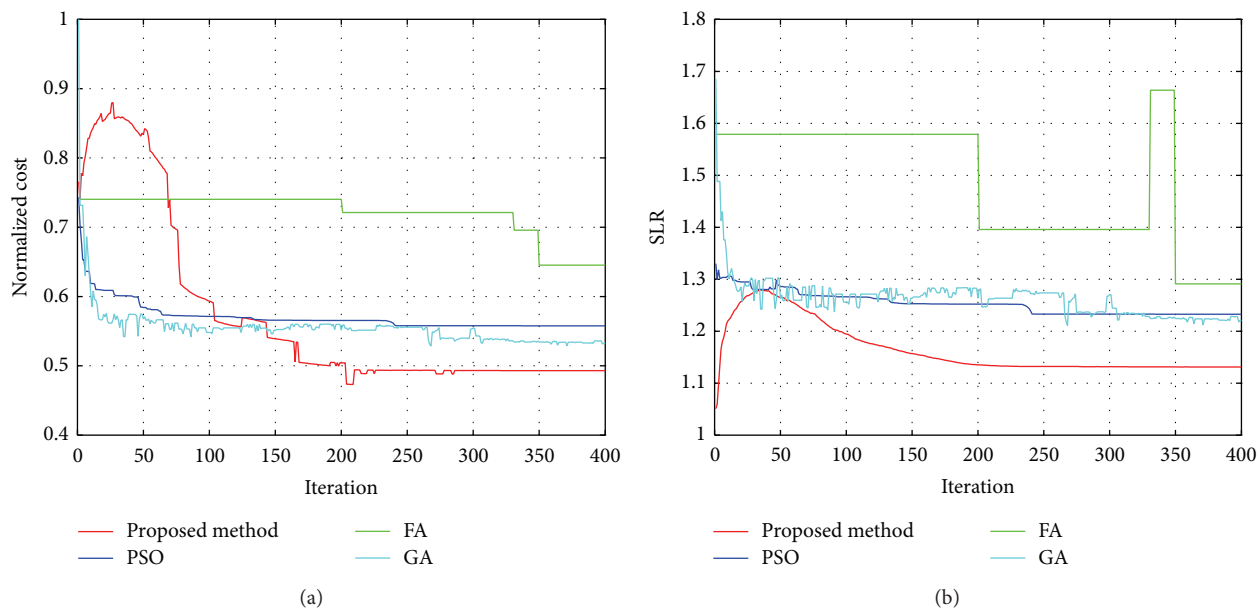


Figure 12: Performance comparisons of four algorithms with $S = 400$ and $D = 10$ in *Scenario 1*. (a) Normalized cost comparisons of different algorithms. (b) Path length comparisons of different algorithms.

all the curves in Figure 9 is that FA realizes path planning perfectly with $S = 400$ avoiding all the threats. Because by increasing the number of particles in the swarm the searching space and searching times are enlarged, which makes more candidate solutions included, in which the global solutions are included. The other three algorithms keep the same with good performances. Particularly GA still tries to fly bypassing all the threats.

The comparisons of cost and path length are given in Figures 12(a) and 12(b), respectively. There are no doubts all the algorithms are getting smooth which are caused by large number of particles in the swarm. FA is still the worst with the biggest cost values. PSO and GA are in the middle place. The two give downtrend from the beginning to the last iteration. The proposed one is the best with least cost. The proposed method become the best before the 150th generation. It is 11.61%, 23.60%, and 7.45% better compared with PSO, FA, and GA, respectively, in cost values.

In Figure 12(b) FA jumps up and down obviously as many steps. PSO and FA are similar with big fluctuation around an approximation. GA is the most unstable and the curve always changes, which is hard to find an optimal value. The curve of proposed method is the smoothest of all almost without any big jumping, which is 8.24%, 12.39%, and 7.33% better compared with PSO, FA, and GA, respectively.

Table 4 lists other properties of the four methods. It can conclude that the proposed path planning algorithm could fulfill the mission perfectly and its performances are the best compared with other three classic algorithms.

4.2.2. Performance Comparisons under Scenario 2. To study the feasibilities of the algorithms under other situations we make experiments under a more complicated scene, namely, *Scenario 2*, with much more threat areas. As stated before there will be nine threat areas in the flying fields and also they are overlapped with each other, which is shown in Figure 13.

Table 4: Comparisons with $S = 400$ and $D = 10$ in *Scenario 1*.

Category	Algorithm	Best	Worst	Mean	Std.
Cost value	Proposed	**0.7640**	**0.7397**	**0.7519**	**0.6351**
	PSO	0.8643	0.7432	0.7920	0.6366
	GA	0.8255	1	0.7706	1
	FA	1	0.7403	1	0.9504
Path length	Proposed	**150.4**	**170.2**	**154.8**	**0.1404**
	PSO	163.9	176.9	166.6	0.1816
	GA	162.3	224.2	167.9	0.3494
	FA	171.6	221.2	197.6	1

Figure 13 gives the UAV path planned by the four different algorithms with $S = 100$ and $D = 10$ in *Scenario 2*. Under this complicated scenario, FA and PSO all failed to get to the destination without flying out of the threats. Particularly most of the flying route given by FA is in the threats and there are also many turnings on the path, which is difficult for a UAV to follow. Because the motility of UAV itself is very limited. There is a small part of PSO path also in the dangerous area because of so many threats in the field. GA shows the same characteristics of flying bypassing all the threats. But our proposed method is still perfect. Not only it avoids threats but also it is easy for UAV to follow for least turnings on it with big turning angles.

In *Scenario 2* shown in Figure 14(a) PSO easily falls into local optimum at the beginning and stays to the end. FA is the worst of all and it needs much time to find a solution. GA is in the middle position. At the beginning it fluctuates and then stays stable till the end. The proposed one is the best except

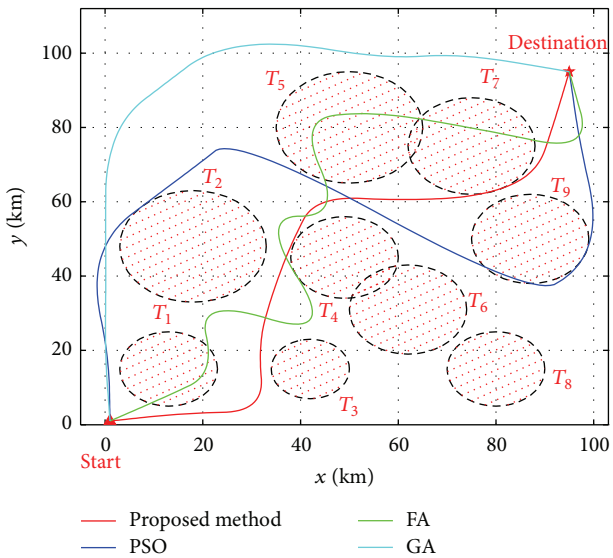


Figure 13: Planned path for UAV with $S = 100$ and $D = 10$ in *Scenario 2*.

a singular point at the 64th generation. The global optimal solution is found around 200 iterations with least cost. It is 4.46%, 68.45%, and 18.81% better than PSO, FA, and GA, respectively, in path cost.

Figure 14(b) gives the path length comparisons. From the FA curve we can find it not stable at all with a lot of big jitters, which proves it is hard to find a reliable solution in this scenario. Many jumping points are located on the curve. PSO moves steadily and almost in a straight line, but it is much longer than that of GA and proposed method. Also we can find PSO easily falls into local optimal. There are a lot of small fluctuations on GA compared with FA. But it is shorter than that of FA. Combining the GA curves in Figure 13, GA always flies a long distance to avoid all the threat in the flying areas. Finally, there is no doubt our proposed method is the best since the 79th generation. It moves stably and produces the shortest path length for UAV in two-dimensional field.

4.2.3. Paired Samples t-Test. In order to find out how different the proposed method is from the other three by probability analysis, a paired samples t-test is made with PSO, FA, and GA, respectively, under the experiment conditions of $S = 100$ and $D = 10$ in *Scenario 1*. The data of cost values and planned path lengths are used for the realization of t-test. In this way there are three pairs as follows: proposed method versus PSO, proposed method versus FA, and proposed method versus GA. The results are shown in Tables 5 and 6.

In Table 5 the original hypothesis H_0 and alternative hypothesis H_1 are listed. The original hypothesis H_0 means that there are no essential differences between the two compared algorithms. Oppositely the alternative hypothesis H_1 holds that big differences exist in the related two methods in a group and the two are totally different methods. Table 6 shows the results of t-test, which is indicated in a bold font.

By the P values in the t-test as shown in Table 6, conclusions are given in Table 5; namely, all the original

Table 5: Paired samples t-test conclusions.

t-test	$S = 100, D = 10$					
H_0	Proposed = PSO		Proposed = FA		Proposed = GA	
H_1	Proposed ≠ PSO		Proposed ≠ FA		Proposed ≠ GA	
P value	Cost	Length	Cost	Length	Cost	Length
	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Result	Reject		Reject		Reject	

hypothesis H_0 is rejected and alternative hypothesis H_1 is all accepted. In this way it means that our proposed algorithm is a feasible path planning technique for UAV which is also essentially different from other three classic ones. Also its performances are the best of all compared with the other classic three.

4.2.4. Self-Performance Comparisons under Scenario 1. In order to explore the effects of different swarm sizes and particle dimensions on the cost values and planned path lengths, experiments of our proposed method are made under *Scenario 1*. All the results are all shown in the following.

The swarm sizes are set as $S = 50$, $S = 200$, $S = 300$, and $S = 400$, respectively. The particle dimensions are set as $D = 10$, $D = 15$, $D = 20$, and $D = 25$.

As Figure 15(a) shows the number of swarm sizes is changed from 50 to 400. The flying paths for UAV are all planned well under different swarm sizes, which avoids all the threat areas. Also when $S = 200$ and $S = 300$ the two trajectories are in coincidence most of the time. The differences between four curves are not obvious. In this way the swarm size can be set between 50 and 200, which not only saves calculation time and power consumption but also can obtain good performances. Figure 15(b) gives the effects of particle dimensions which is changed from 10 to 25. The proposed method can realize path planning under the four different conditions avoiding all the threat areas. The differences are not so big. Figure 15 proves the accuracy of our proposed method under all kinds of environment setting, which could be used in a wide range of real applications.

Once the effects of the two parameters on the planned UAV path are determined, they can be set as the environment changes under different applications. The cost values and path lengths are compared in Figures 16(a) and 16(b).

Figure 16 gives the swarm size's effects on cost values and path lengths, respectively. The curves of cost values present the same trend till the global optimal solutions are found. As listed in Figure 16(a) the worst two of them are $S = 50$ and $S = 400$. When $S = 200$ it has the least cost value. $S = 300$ is in the middle position. So conclusions can be made in the following. Too many or too few particles in a swarm will both lead to higher cost values. However, the medium numbers of swarm sizes could obtain better cost values. Figure 16(b) shows how the path length changes as the swarm sizes. The trend of the four is more or less the same, especially when $S = 200$ and $S = 400$. Also $S = 50$ is the worst and $S = 300$ is the best. So it can be set as different values based on the UAV flying distance.

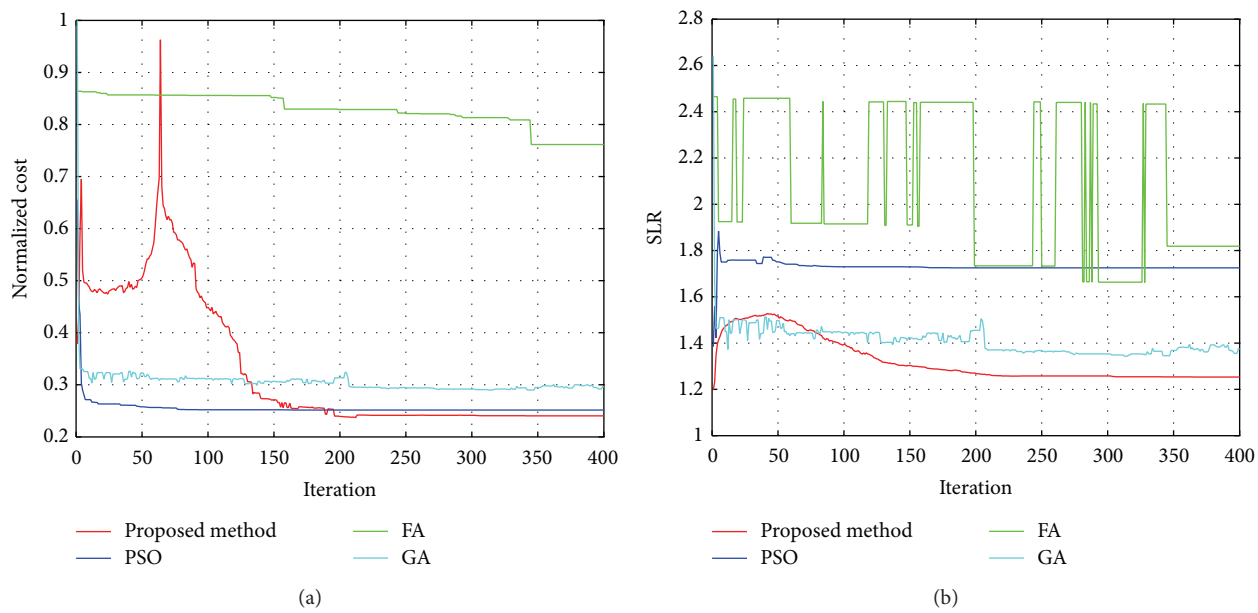


Figure 14: Performance comparisons of four algorithms with $S = 100$ and $D = 10$ in *Scenario 2*. (a) Normalized cost comparisons of different algorithms. (b) Path length comparisons of different algorithms.

Table 6: Results of the paired samples t-test.

(a) t-test of cost values

	Paired differences					t	df	Sig. (2-tailed)
	Mean	Std. deviation	Std. error mean	95% confidence interval of the difference				
				Lower	Upper			
Pair 1 proposed method: PSO	−.05496	.04522	.01011	−.07612	−.03379	−5.435	19	**.000**
Pair 2 proposed method: FA	−.42184	.04838	.01082	−.44448	−.39919	−38.996	19	**.000**
Pair 3 proposed method: GA	−.06850	.02255	.00504	−.07905	−.05794	−13.582	19	**.000**

(b) t-test of planned path lengths

	Paired differences					t	df	Sig. (2-tailed)
	Mean	Std. deviation	Std. error mean	95% confidence interval of the difference				
				Lower	Upper			
Pair 1 proposed method: PSO	−13.54855	9.27905	2.07486	−17.89128	−9.20582	−6.530	19	**.000**
Pair 2 proposed method: FA	−34.36477	19.56367	4.37457	−43.52085	−25.20869	−7.856	19	**.000**
Pair 3 proposed method: GA	−18.12088	4.41683	.98763	−20.18802	−16.05374	−18.348	19	**.000**

From the analysis above we can find the swarm size can make effects on the cost values and planned UAV lengths in a similar way. Different values will produce different results. But the changing trends on the curves are similar, based on if it can meet many different requirements. In order to make perfect planned UAV flying route under different application scenarios, different parameter settings are needed.

Effects from different particle dimensions on the two indexes are given in Figure 17. As shown in Figure 17(a) the differences in cost values caused by particle dimensions are not obvious. Larger particle dimensions bring less cost than smaller ones. Figure 17(b) gives that the path lengths of $D = 10$ and $D = 15$ are very similar as well as the changing trend from the first iteration to the last one. It will be the shortest path length when $D = 25$.

From Figure 17 we can conclude that more particle dimensions will bring less cost and shorter path, but correspondingly the calculation will become more complicated and more calculation time will be needed. So how to set this parameter depends on the applied environments.

5. Conclusion

In this paper a potential odor intensity grid based UAV path planning algorithm is proposed by combining standard PSO technique. A potential grid construction operator is

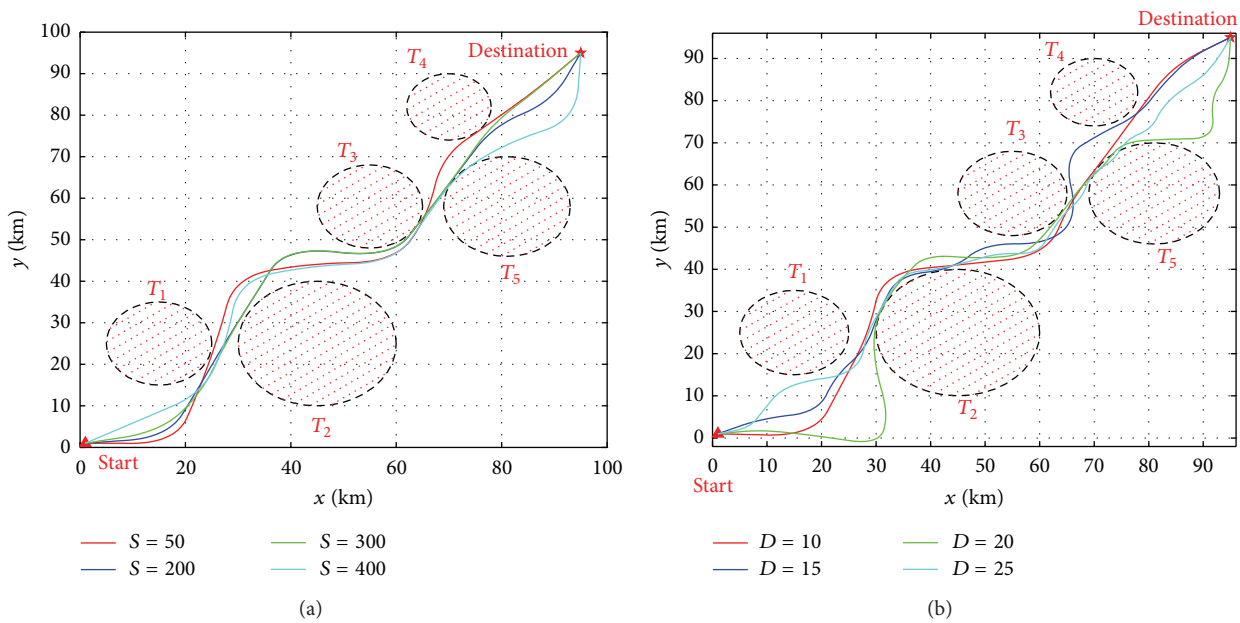


Figure 15: Planned path in *Scenario 1* with different experiment settings. (a) Effects of different swarm sizes. (b) Effects of different particle dimensions.

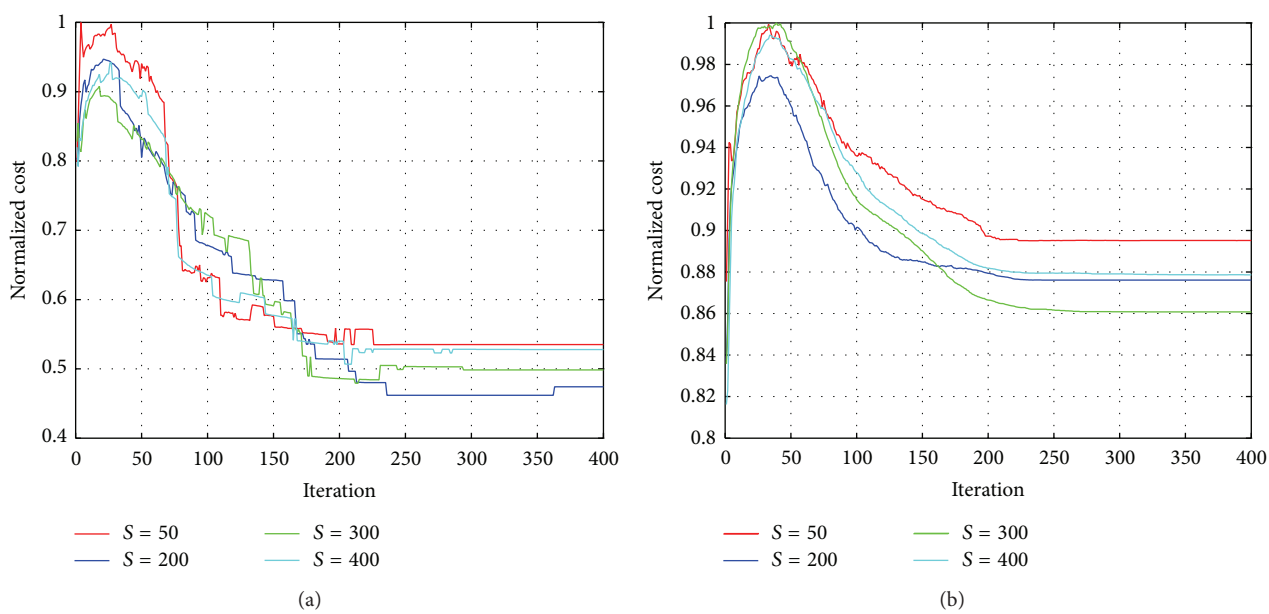


Figure 16: Effects of different swarm sizes on cost values and planned path length. (a) Cost value comparisons of different swarm sizes. (b) Path length comparisons of different swarm sizes.

designed in our model to identify the different levels of odor intensity, which is implemented easily and avoids local optima and slow convergence. Two of areas in the searching space with highest probability where candidate particles may locate will be colored depending on different odor intensities, including centers of the two grids and the set side length. Then middle point of the two grid centers will be used as the final position in current particle dimension of current iteration. The global optimum solution will be solved as the average. Also upper and lower boundaries of solution space in each particle dimension are restricted based on properties of threats themselves in the field to avoid prematurity In addition, objective function is redesigned by taking minimum direction angle to destination into account and a sampling method is introduced to better evaluate the cost of path segments into the threat areas and straight line rate (SLR) is used to evaluate the planned path length. A paired samples t-test and experimental results both demonstrate

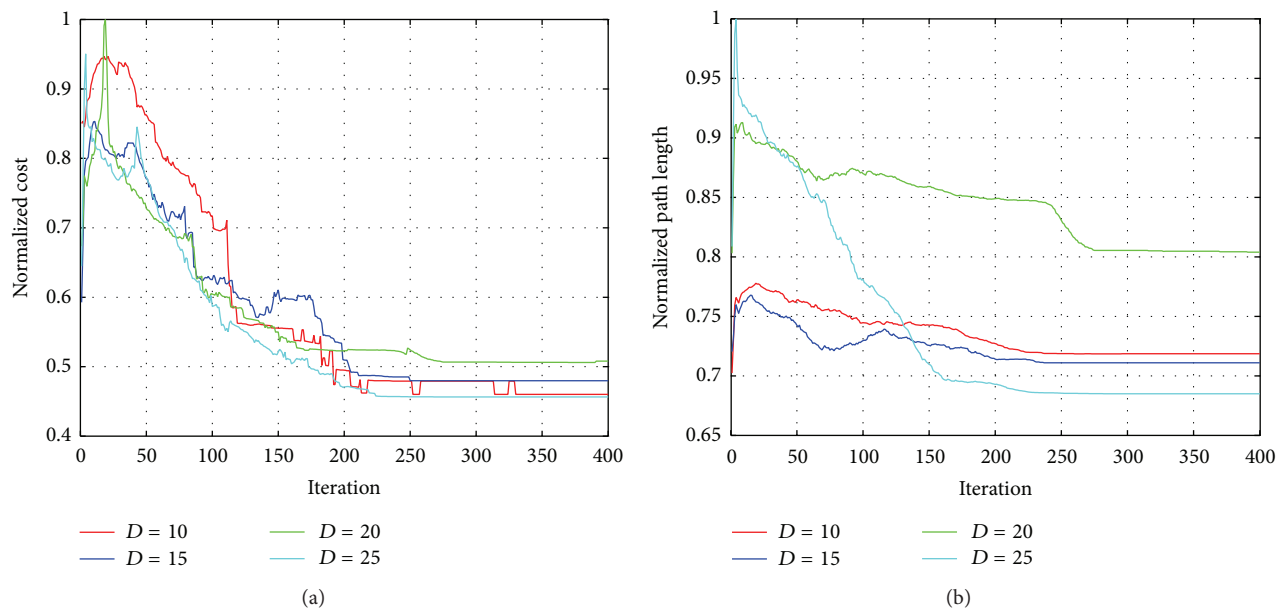


Figure 17: Effects of different particle dimensions on cost values and planned path length. (a) Cost value comparisons of different particle dimensions. (b) Path length comparisons of different particle dimensions.

the proposed method is capable of generating higher quality paths efficiently for UAV than any other tested optimization algorithms, standard PSO, FA, and GA.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. U1433203 and Grant nos. U1533119 and L142200032) and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant no. 61221061).

References

[1] W. Zhan, W. Wang, N. Chen, and C. Wang, "Efficient UAV path planning with multiconstraints in a 3D large battlefield environment," *Mathematical Problems in Engineering*, vol. 2014, Article ID 597092, 12 pages, 2014.

[2] Y. Zhang, L. Wu, and S. Wang, "UCAV path planning by fitness-scaling adaptive chaotic particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 705238, 9 pages, 2013.

[3] Q. Zhang, J. Tao, F. Yu, Y. Li, H. Sun, and W. Xu, "Cooperative solution of multi-UAV rendezvous problem with network restrictions," *Mathematical Problems in Engineering*, vol. 2015, Article ID 878536, 14 pages, 2015.

[4] J. M. Peschel and R. R. Murphy, "On the human-machine interaction of unmanned aerial system mission specialists," *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 1, pp. 53–62, 2013.

[5] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 132–141, 2013.

[6] J. Kim, D. Lee, K. Cho, J. Kim, and D. Han, "Two-stage trajectory planning for stable image acquisition of a fixed wing UAV," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 3, pp. 2405–2415, 2014.

[7] A. Babaie and J. Karimi, "Optimal trajectory planning for a UAV in presence of terrain and threats," *Aerospace and Mechanics Journal of Imam Hossein University*, vol. 7, no. 2, pp. 57–69, 2011.

[8] E. Besada-Portas, L. De La Torre, A. Moreno, and J. L. Risco-Martín, "On the performance comparison of multi-objective evolutionary UAV path planners," *Information Sciences*, vol. 238, pp. 111–125, 2013.

[9] A. Altmann, M. Niendorf, M. Bednar, and R. Reichel, "Improved 3D interpolation-based path planning for a fixed-wing unmanned aircraft," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 76, no. 1, pp. 185–197, 2013.

[10] F. Yangguang, D. Mingyue, and Z. Chengping, "Route Planning for Unmanned Aerial Vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 43, no. 6, pp. 1451–1465, 2013.

[11] Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV," *Aerospace Science and Technology*, vol. 16, no. 1, pp. 47–55, 2012.

[12] Y. Wu and X. Qu, "Path planning for taxi of carrier aircraft launching," *Science China Technological Sciences*, vol. 56, no. 6, pp. 1561–1570, 2013.

[13] P. Melchior, B. Orsoni, O. Lavialle, A. Poty, and A. Oustaloup, "Consideration of obstacle danger level in path planning using A* and fast-marching optimisation: comparative study," *Signal Processing*, vol. 83, no. 11, pp. 2387–2396, 2003.

[14] R. J. Szczerba, P. Galkowski, I. S. Glickstein, and N. Ternullo, "Robust algorithm for real-time route planning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 3, pp. 869–878, 2000.

[15] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 354–363, 2005.

[16] W. Liu, Z. Zheng, and K.-Y. Cai, "Bi-level programming based real-time path planning for unmanned aerial vehicles," *Knowledge-Based Systems*, vol. 44, pp. 34–47, 2013.

[17] D. Eppstein, "Finding the k shortest paths," *SIAM Journal on Computing*, vol. 28, no. 2, pp. 652–673, 1999.

[18] Y. G. Fu, M. Y. Ding, and C. P. Zhou, "Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 42, no. 2, pp. 511–526, 2012.

[19] H. B. Duan and F. P. Li, *Bio-Inspired Computation in Unmanned Aerial Vehicles*, Springer, Berlin, Germany, 2014.

[20] A. Ponsich and C. A. C. Coello, "Differential evolution performances for the solution of mixed-integer constrained process engineering problems," *Applied Soft Computing*, vol. 11, no. 1, pp. 399–409, 2011.

[21] A. N. Brintaki and I. K. Nikolos, "Coordinated UAV path planning using differential evolution," *Operations Research*, vol. 5, no. 3, pp. 487–502, 2005.

[22] F. Neri and E. Mininno, "Memetic compact differential evolution for cartesian robot control," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 54–65, 2010.

[23] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[24] G. Iacca, F. Caraffini, and F. Neri, "Memory-saving memetic computing for path-following mobile robots," *Applied Soft Computing*, vol. 13, no. 4, pp. 2003–2016, 2013.

[25] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.

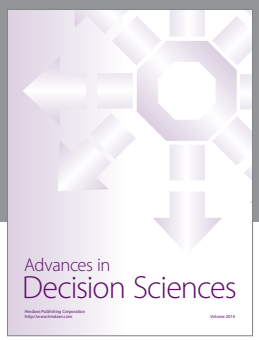
[26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, IEEE, Perth, Australia, December 1995.

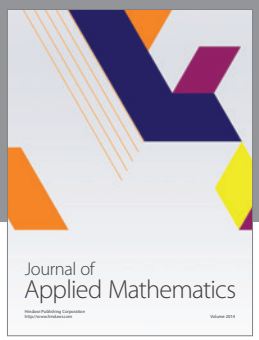
[27] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: basic concepts, variants and applications in power systems," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171–195, 2008.

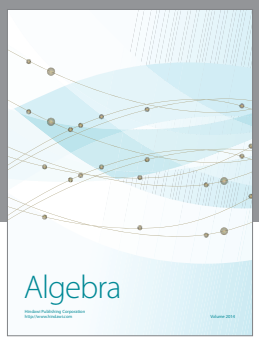
[28] P. Yao, H. Wang, and Z. Su, "Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment," *Aerospace Science and Technology*, vol. 47, pp. 269–279, 2015.

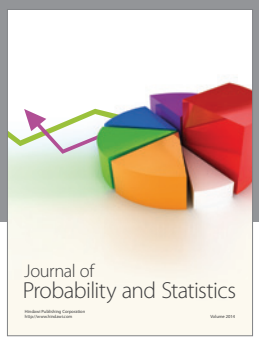
[29] J. Karimi and S. H. Pourtakdoust, "Optimal maneuver-based motion planning over terrain and threats using a dynamic hybrid PSO algorithm," *Aerospace Science and Technology*, vol. 26, no. 1, pp. 60–71, 2013.

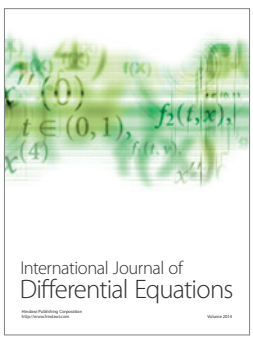
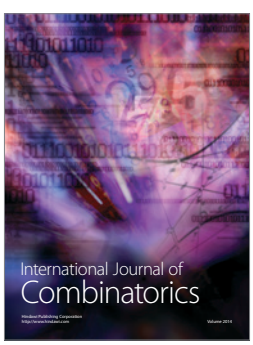
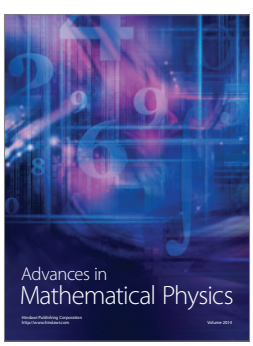
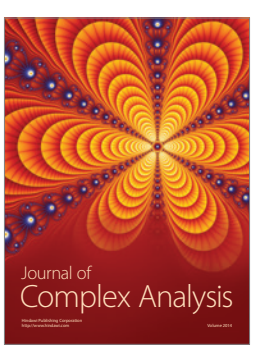
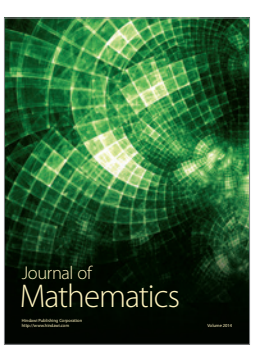
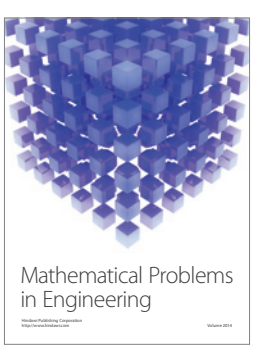
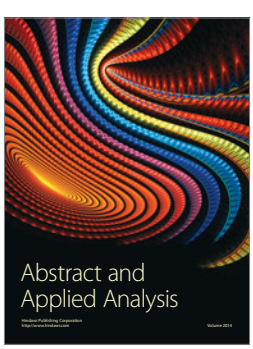
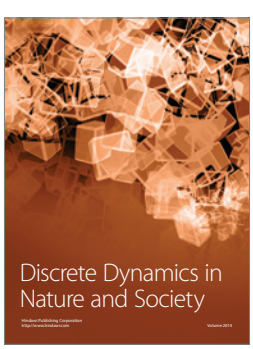
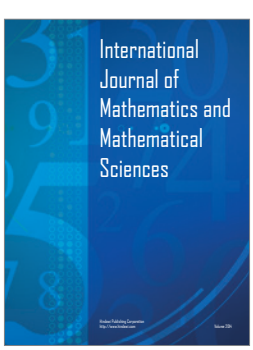
[30] X. Zhang and H. Duan, "An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning," *Applied Soft Computing Journal*, vol. 26, pp. 270–284, 2015.









Hindawi

Submit your manuscripts at
http://www.hindawi.com