



Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm



Chen YongBo^{a,b}, Mei YueSong^{a,b,*}, Yu JianQiao^{a,b}, Su XiaoLong^{a,b}, Xu Nuo^{a,b}

^aSchool of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China

^bKey Laboratory of Dynamics and Control of Flight Vehicle, Ministry of Education, Beijing 100081, China

ARTICLE INFO

Article history:

Received 7 March 2015

Revised 22 May 2017

Accepted 23 May 2017

Available online 31 May 2017

Communicated by Prof. Zidong Wang

Keywords:

Unmanned aerial vehicle (UAV) path planning

Modified wolf pack search (WPS) algorithm

Genetic algorithm (GA)

Three dimensional (3D) space

Cubic B-spline curve

ABSTRACT

The unmanned aerial vehicle (UAV) has been a research focus in recent years. The path planner is a key element of the unmanned aerial vehicle autonomous control module. In this paper, the modified wolf pack search (WPS) algorithm is applied to compute the quasi-optimal trajectories for the rotor wing UAVs in the complex three-dimensional (3D) spaces including the real and fake 3D spaces. Moreover, it adopts the multi-objective cost function. In the path planning process, some concepts in the genetic algorithm (GA) are applied to realize the WPS algorithm. Then, the crossover and mutation operators in the GA method are introduced to improve the original WPS algorithm. Considering the dynamic properties of the vehicle, the path smoothing process based on the cubic B-spline curve is used to make the planning path suitable for the fixed wing UAVs. Simulation results show that this approach is efficient for the rotor wing UAVs and the fixed wing UAVs when taking into account of all kinds of constraints and the path generated is flyable. Moreover, the comparisons of the four algorithms show that the trajectories produced by the modified WPS algorithm are far superior to the original WPS algorithm, the GA and the random search way under the same conditions.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

The Unmanned Aerial Vehicle (UAV) path planning has become one of the most important elements to define a UAV mission. It allows the UAV to autonomously compute the best path from a start point to a target based on the task requirements and restraints. With the extensive use of various types of UAVs, the topic has attained a high popularity in recent years.

Various methods have been raised to deal with the UAV path planning problem, such as artificial potential field (APF) method [1], A* algorithm [2], genetic algorithm (GA) [3], artificial intelligence algorithms [4,5] and so on. They can be segmented into traditional optimizing methods, computational intelligence methods and non-optimal algorithms. Most of these methods can solve the path planning problems effectively in general planning space.

The traditional optimizing methods include the APF method, the mixed-integer linear programming (MILP) method, the optimal control method and so on. Essentially, the APF method is based on

the gradient method. This traditional method has been improved in the last several years. The authors introduced the airborne communication relay approach and the harmonic functions into the APF method for the autonomous UAVs path planning [6]. But the APF method has its own obvious deficiency. The main concern in the potential field methods is how to choose such a navigation function in order to avoid the vehicle being trapped in a local minimum. The MILP method is widely used in many areas of science, and it has been applied in the UAV path planning problem [7,8]. Since the integer programming, in general, is an NP-hard (non-deterministic polynomial-time hard) problem, the solution to the MILP optimization problem is not necessarily unique [9]. Similarly, the optimal control method is also a universal method [10]. Not only it is used in the UAV path planning problem, but also it can be applied in the UAV trajectory planning problem [11]. But the optimal control problem is very difficult to solve in most situations.

In general, the traditional optimizing methods are easily trapped into the local minimum. And there also exists the problem that they lack sufficient robustness. So the computational intelligence methods become more and more popular in the UAV path planning problem. The authors in [3] used the GA to solve the multi-constraints 3D UAV path planning problem. The authors of [5] compare the parallel GA with the parallel PSO algorithm in the real-time 3D path planning problem. The authors in [12]

* Corresponding author at: School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China.

E-mail addresses: bit_chenyongbo@163.com (C. YongBo), mys001@bit.edu.cn (M. YueSong), jianqiao@bit.edu.cn (Y. JianQiao), suxiaolong@bit.edu.cn (S. XiaoLong), promise_moon@126.com (X. Nuo).

develop an anytime algorithm using the particle swarm optimization to solve the multiple UAV path planning problem. The authors of [13] presented two types of path planners based on the evolutionary algorithm in the on-line and off-line situation. Although the most computational intelligence methods have a powerful ability of global optimization, the efficiencies of the different methods are uneven for the same problem. So the efficiency is a very important index to measure an algorithm is good or bad.

The non-optimal algorithm is the generic terms of many different methods, includes the A* method, the Voronoi diagrams approach [14], the rapidly-exploring random tree (RRT) approach [15], the partially observable Markov decision processes (POMDP) and some other heuristic algorithms. The A* method and Voronoi diagrams approach are the graph-based algorithms. Importantly, the RRT approach is very popular in the UAV path planning area [13,15,16]. There are a series of papers and books focused on this approach. For example, the authors in [16] introduced an anytime algorithm based on a novel RRT* algorithm (like the RRT) which can converge to an optimal solution. In addition, a novel path-planning algorithm based on the POMDPs is presented recently. The authors of [17,18] design a path-planning algorithm to guide the UAVs for tracking multiple ground targets based on the POMDPs.

Because the single method has its own inherent shortage, there are more and more researchers using the hybrid algorithm which integrates two or more methods to get a better planning result. For example, the authors put forward a novel algorithm combining the APF method with the GA for the robot path planning and the obstacle-avoidance [19].

The idea of the wolf pack search algorithm is first presented by the authors in [20]. They give a new algorithm, the wolf pack search-marriage in honey bees optimization (WPS-MBO) which introduces the WPS algorithm into the local search process of Marriage in Honey Bees Optimization algorithm [21]. At the same time, the definition of the WPS algorithm is proposed. The WPS algorithm is a swarm-intelligence method that imitates the arrest activity of a wolf group. Some popular complex Evaluation Functions and Traveling Salesman Problem (TSP) are successfully solved by the WPS algorithm. Importantly, its convergence performance is better than some other swarm-intelligence methods.

However, in the field of the complex 3D path planning for the UAV, no application of the WPS algorithm exists yet. In our paper, an original WPS algorithm and an improved WPS algorithm are used to solve the 3D UAV path planning problem. The concept of the GA chromosome is introduced to describe the meaning of the wolf in the path planning problem. Each wolf means an inherent chromosome and a possible path during the whole solution. The chromosome is the main operation object of the WPS algorithm. Especially, the environment description and the encoding of the chromosome are the major part of the solution process. After that stage has been overcome, the original WPS algorithm consists of the following steps: 1) Initialization – an initial wolf pack; 2) Fitness – evaluate the cost functions of the whole pack; 3) Elitism – select a more appropriate small group in the pack; 4) Safari – optimization in a small scale; 5) Update – every wolf gets close to the best wolf; 6) Replacement – replace old weak wolves by new ones; 7) Loop – go back to step 2 [22]. Of course, the whole circulation will stop if an end condition is satisfied, otherwise, it will continue. Here, the crossover and mutation operations in the GA method are added between Step 5 and Step 6 to propose a new metaheuristic algorithm according to the principle of the WPS algorithm, and then the improved WPS algorithm is used to search the optimal or suboptimal path in the planning space.

Therefore, it is observed that the processes of the original and modified WPS algorithm are both random and heuristic. Essentially, the original WPS algorithm is a random multi-initial values

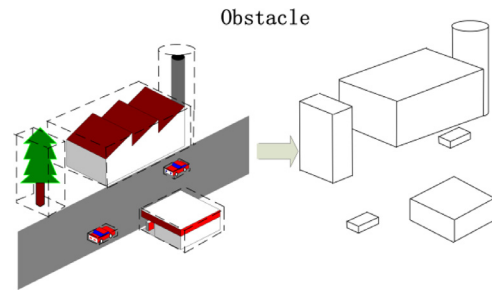


Fig. 1. The regular graphic way to describe the obstacles.

gradient algorithm, and it also has the characteristics included the parallel computing ability and the micro-region search capacity. So the original WPS algorithm characterizes with the good global search ability and the strong convergence ability. The original WPS algorithm is based on the relation between the best wolf and other common wolves. Therefore, the wolf pack is excessively dependent on the elitism. The connections of each two wolves are insufficient. By the help of the crossover and mutation operations, the relations among the whole wolf pack are introduced into the modified WPS algorithm. Therefore, the modified WPS algorithm has all advantages of the original WPS algorithm. At the same time, it also includes some new idea.

The general framework of this paper is as follows: firstly, the representations of the obstacles are statically defined. Then, the optimal path is computed off-line and it is properly smoothed. Finally, the UAV can move in the space according to the computed path.

The structure of this paper is as follows: In Section 2, the environment description models of the UAV path planning problem consider two different spaces. The encoding of the chromosome involving the genes and the structure of the chromosome and the multi-objective cost function of the chromosome are presented. Then, the whole optimization model of the path planning problem is established. In Section 3, the original WPS algorithm is introduced and applied to solve the problem. In Section 4, the improved WPS algorithm with the crossover and mutation operations for UAV is presented. The path smoothing which is the follow-up work of the UAV path planning problem is solved by the cubic B-spline curve in Section 5. At last, the simulation work and the further discussion are accomplished in Section 6.

2. Problem formulation

2.1. Obstacles

The first step of the UAV path planning is to describe the obstacles O of the planning space. The way used to describe the obstacles has an effect on the representation of the trajectory path and the search algorithm.

The way used to describe obstacles can be divided into two kinds. One way is to define the obstacles of planning space into several simple regular graphics, which can be applied in the simulation of small scale (see Fig. 1). These simple regular graphics include: cube, cylinder, sphere and so on. However, it is difficult to use this way to describe the broad space. So the other way is to discretize the broad planning space into a representation and the main obstacles of the broad planning space are terrains. The terrains can be generated using the interpolation method by the feature point coordinates (X_m, Y_m, Z_m) which can show the trend of the terrain (see Fig. 2), where m is the number of the points. We select the 4-point spline interpolation method for the generation of the terrain surface. After the interpolation process, the terrain is

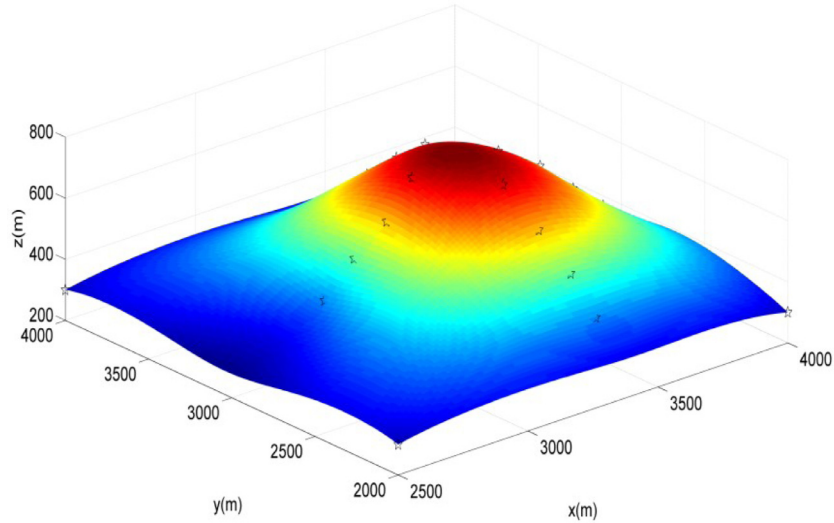


Fig. 2. The interpolation way to describe the obstacles.

decomposed into a matrix M whose elements represent the elevation Z_m of the terrain.

$$M = [Z_m] = [f_{\text{interpolation}}(X_m, Y_m)], \quad (1)$$

where $f_{\text{interpolation}}$ means the process of the interpolation. The matrix subscript means the horizontal coordinates Z_m of the terrain.

In the planning space, the threats T can be perceived as the soft barrier. In fact, UAV may fly in the danger zone at the real battle-field environment. But for simplicity, the threat zone is regarded as a no-fly zone where the trajectories of UAV cannot be allowed into. So the total obstacles O combine the terrain S with the threats T in the flying environment of the UAV:

$$O = S \cup T, \quad (2)$$

In addition, an error Δ may be introduced during the real obstacles description process. The range D_Δ of the error Δ is obtained by the performance of the sensors of the UAV or other reconnaissance equipment. In order to solve this problem, the suitable space margin needs to be left. In other words, the obstacles need to be magnified extra to ensure that the new obstacles O' are larger than the ideal obstacles O :

$$\begin{aligned} O' &= cO \\ O' &\supset O + D_\Delta \end{aligned} \quad (3)$$

where c means amplification factor. And it satisfies the condition $c > 1$.

2.2. Gene

In order to use the original and modified wolf pack search algorithm, the planning space needs to be discretized into a representation by the disperse waypoints $\text{waypoint}(n) = (x(n), y(n), z(n))$, ($n = 1, 2, \dots, N_{\text{all}}$) where n means the order number of the waypoint, and N_{all} means the total number of the waypoints. Based on the two different types of obstacles, the disperse waypoints can also be segmented into two kinds. One is used in the graphical way situation shown in Fig. 1, and the other is used in the interpolation way situation shown in Fig. 2.

The planning space using the graphic way (situation 1) is called a real 3D space whose the layout way of the waypoints is independent of the 3D obstacles. In detail, we need to distribute the points in the 3D space uniformly, and the distance between each two neighborhood points is dx , dy or dz . Once the sizes of the planning space dx , dy and dz are determined, the waypoints are

confirmed. The obstacles in the 3D planning space are independent, and the waypoints produce a dense space lattice. However, the layout method of the waypoints is abridged from the obstacle in situation 2. The relationship between the space coordinates of waypoints and obstacles is as follows: If the horizontal coordinates of waypoint is (x, y) and the corresponding space coordinates of the obstacle point is $(x, y, f_{\text{interpolation}}(x, y))$, the elevation of waypoint will be $f_{\text{interpolation}}(x, y) + h$ ($h > 0$), where h means the safety distance. Apparently, in this situation, the waypoints only locate in the 3D space curved surface. So the planning space using the interpolation way (situation 2) is called a fake 3D space.

After the layout of the waypoints, gene g , which is the concept in the genetic algorithm, is introduced to solve the UAV path planning problem. Meanwhile, the waypoints can be regarded as the genes:

$$g(k) \in \{\text{waypoint}(n), n = 1, 2, 3, \dots, N_{\text{all}}\}, k = 0, 1, 2, \dots, N, \quad (4)$$

where $g(k)$ means the gene in the k -th group. The gene $g(k)$ includes in the k -th group waypoints. So the waypoints need be divided into N groups.

Then, it is generally accepted that the genes can be distributed into the healthy genes G_h and the faulty genes G_f . In the wolf pack search algorithm, the faulty genes mean that the waypoints located in the obstacle O' and the rest of the genes are the healthy ones. So we need to classify the waypoints of the whole planning space into the healthy waypoints set H_w and the faulty waypoints set F_w . Obviously, At the same time, the waypoints also need to be grouped according to their positions. The detailed description of the classification process and the grouping process of the waypoints is showed in Table 1.

2.3. Chromosomes

It is well known that the genetic algorithm is based on the chromosome structure. Obviously, the whole process of GA is to operate the genes of the chromosomes in the different individual. So in this paper, the original and modified wolf pack search algorithm draws on the experience of the genetic algorithm. The original and modified wolf pack search algorithm also operates the waypoints of the planning paths by the inspiration of the genetic algorithm. Then, some concepts of the chromosome need to be reified in the original and modified wolf pack search algorithm.

As an individual, the i -th wolf has its unique chromosome C_i which means a whole planning path. Furthermore, the unique

Table 1
Algorithm 1.

Algorithm 1 The classification process of the waypoints

```

1: Define  $dx, dy, dz, h$ ; %Define the distances between each two neighborhood points and the safety distance.
2: if the type of the 3D space == real
3:      $l=length(space)/dx, J=width(space)/dy, K=height(space)/dz$ ; %Divide the planning space,  $length(space)$ ,  $width(space)$  and  $height(space)$  mean the
length, width and height of the planning space.
4:     waypoint( $n$ )= $(x(n), y(n), z(n))=(dx*fix[n/(J*K)], dy*fix[rem[n,(J*K)]/K], dz*rem[rem[n,(J*K)],K])$ ; %According to the serial numbers  $l, J$  and  $K$ , define the
waypoints.
5:     for  $n=0:1:l*J*K$ 
6:          $k=fix[n/(J*K)]$ ; % Determine the group number  $k$  of the  $n$ -th waypoint.
7:         if waypoint( $n$ )  $\in O$ 
8:             this waypoint is a faulty one and in  $k$ -th group;
9:         else
10:            this waypoint is a healthy one and in  $k$ -th group;
11:         end
12:     end
13: elseif the type of the 3D space == fake
14:      $l=length(space)/dx, J=width(space)/dy$ . % Divide the planning surface.
15:     waypoint( $n$ )= $(x(n), y(n), z(n))=(dx*fix[n/J], dy*rem[n,J], f_{interpolation}(x(n), y(n))+h)$ ; %According to the serial numbers  $l$  and  $J$ , define the waypoints.
16:     Determine the grouping scheme  $group$  according to the start point and target. There are no tough standards to define  $group$ . But the different genes in the
same  $group$  had better locate around the same vertical lines of the straight line between the start point and target.
17:     for  $n=0:1:l*J$ 
18:         for  $k=1:1:sum(group)$ 
19:             if waypoint( $n$ )  $\in k$ -th group
20:                 this waypoint is a healthy one and in  $k$ -th group
21:             end
22:         end
23:     end
24: end
25: The set includes all healthy waypoints is  $H_w$ , the set includes all faulty waypoints is  $F_w$ . And they meet:  $G_h = H_w$  and  $G_f = F_w$ .

```

Remark 1: Obviously, all genes in the fake 3D space are healthy genes.

Remark 2: The functions, for instance: fix, rem and sum are used in MATLAB.

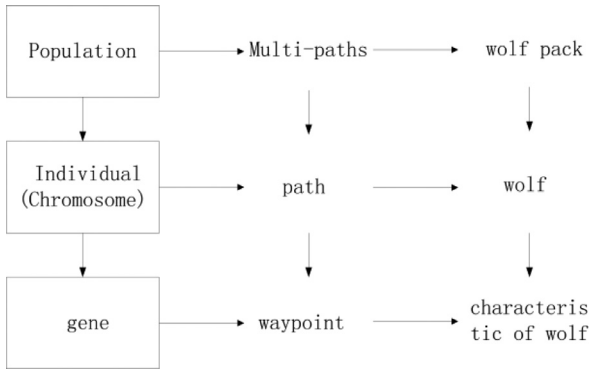


Fig. 3. The hierarchical relationship graph.

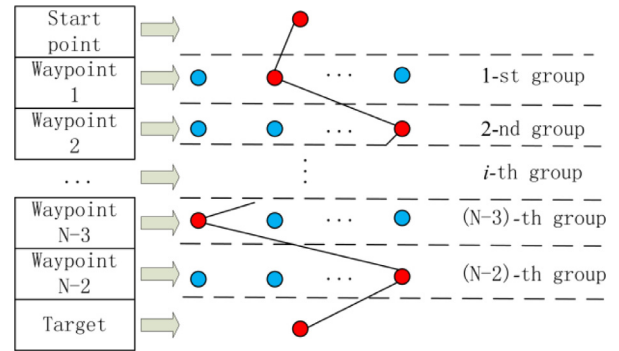


Fig. 4. The chromosome structure chart.

chromosome can show its macroscopic properties which are called the biological diversity. The unique chromosome is caused by the different genes and the disparate connection mode of each two adjacent genes. The hierarchical relationships of the population, individual (chromosome) and gene are shown in Fig. 3.

The chromosome C_i is encoded as a matrix with N genes, each of them is a column of the matrix. The first gene and the last one of the chromosome are the start point $\mathbf{g}(0)$ and the target point $\mathbf{g}(N-1)$. C_i represents a planning path from the start point to the target during the different waypoints:

$$C_i = [\mathbf{g}_i^T(0), \mathbf{g}_i^T(1), \mathbf{g}_i^T(2), \dots, \mathbf{g}_i^T(N-1)]^T. \quad (5)$$

In the UAV path planning problem, the concrete form of the gene is the waypoint that has the specific sequence. Moreover, each waypoint in the k -th point group is predefined. Then, the genes in the different groups are connected to each other and the lineage structure of the multi-genes is the chromosome. The structure of the chromosome is showed in Fig. 4.

In general, the connection way of the genes is the line segment:

$$\varphi(k) = \mathbf{g}(k+1) - \mathbf{g}(k). \quad (6)$$

This way is simple to implement, but there is a problem on the line segment. Obviously, although the healthy genes are not in the obstacles, the line segment between each two genes may go through the obstacles. The way to judge whether a path goes through the obstacles is to discretize the path into some path points P , and then compare their z coordinate with the heights of their corresponding points, which have the same x and y coordinates, on obstacles. It is noted that, in order to limit the number of the corresponding point into 1, we demand that the total obstacles O is convex, which is easy to be satisfied. For the fake 3D space, the adjustment of the safety distance h can avoid this phenomenon. But when the planning space is the real 3D space, this phenomenon will appear always. So the connection way needs to be modified in advance in the real 3D space. In order to simplify the solution process, the simple regular graphics of the real 3D

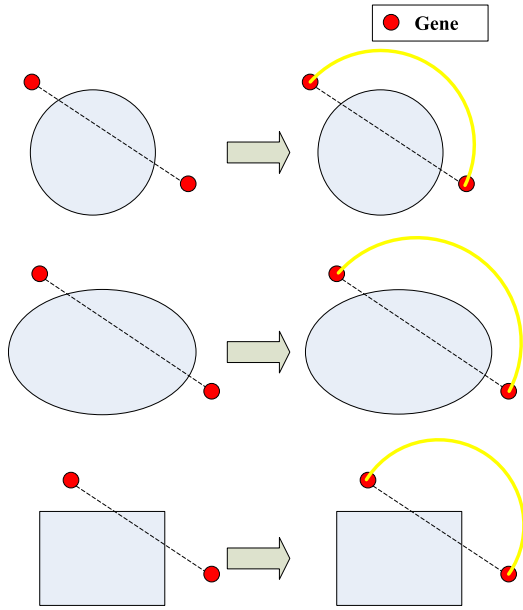


Fig. 5. The modified ways.

space is limited in the shapes: cube, cylinder and sphere. The modified way is shown in Fig. 5.

In Fig. 5, we can see that the modified way is to generate a semi-circle path to replace the original path. However, even the above modified way is used, a special situation also needs to be noted: When the shape of obstacles is vineous which means that the modified path still goes through the obstacles, the cost function of this path needs to be dealt with by a penalty function.

2.4. Cost function

It is known that searching for the best path is often associated with searching for the shortest path. But in the real battlefield environment the UAV needs to adapt to the constant changing conditions. So in the case of the UAV path planning, the optimal path is more complex and includes many different characteristics. Taken these characteristics into consideration, the more complicated cost function of the UAV path planning is used in this paper. The multi-objective cost function in our paper is as follows [23]:

$$J_{\text{cost}} = w_1 \cdot J_{\text{path}} + w_2 \cdot J_{\text{height}} + w_3 \cdot J_{\text{smooth}}, \quad (7)$$

where J_{cost} denotes the total cost function; J_{path} means the cost function of the length of the path; J_{height} is the cost function of the standard deviation of the height; J_{smooth} is the cost function of the smoothness of the planning path; w_i , $i=1,2,3$ denotes the weights of each function and they satisfy the conditions:

$$\begin{cases} w_i \geq 0 \\ \sum_{i=1}^3 w_i = 1 \end{cases} \quad (8)$$

In the UAV path planning process, the length of the planning path is also very important for most path planning missions. Obviously, the shorter path can save more fuel, save more time and give more security. Then, in general, J_{path} is defined as follows:

$$J_{\text{path}} = \sum_{k=1}^{N-1} \|\mathbf{g}(k) - \mathbf{g}(k-1)\|_2, \quad (9)$$

where $\mathbf{g}(k)$ means the k -th gene in a whole planning path. When the line segment between each two genes go through the obsta-

cles, the J_{path} needs to be computed by the following penalty function:

$$J_{\text{path}} = \begin{cases} J_{\text{path}} + \left(\frac{\pi}{2} - 1\right) \sum_{k_1=1}^{N_1-1} \|\mathbf{g}(k_1) - \mathbf{g}(k_1-1)\|_2 & \text{modified path does not go through obstacles,} \\ \infty & \text{modified path still go through obstacles} \end{cases} \quad (10)$$

where $\mathbf{g}(k_1)$ and $\mathbf{g}(k_1-1)$ mean the k_1 -th and k_1-1 -th special genes whose line segment go through the obstacles, N_1 means the total number of the appearance of the special situation. Of course, the infinite function in Eq. (10) is hardly expressed in real problem. In the real simulation, we had better add a large value V_{max} which is proportional to the number of the path points P_0 in the obstacles: $V_{\text{max}} = KP_0$, where K is the setting penalty value.

But in the special situation shown in Fig. 5, the cost function J_{path} needs to be modified. So J_{path} is changed as follows:

$$J_{\text{path}} = \sum_{k=1}^{N-1} \|\mathbf{g}(k) - \mathbf{g}(k-1)\|_2 + l, \quad (11)$$

where l means the additional part of the modified path.

At the same time, the stable flying height of the UAV is also important for the UAV path planning process. For most air vehicles, the flying height should not change too drastically. The stable flying height can help to reduce the burden of the control system and save more fuel. Then, J_{height} is defined as follows:

$$J_{\text{height}} = \sqrt{\frac{1}{N} \sum_{k=0}^{N-1} (z(k) - \bar{z})^2}, \quad (12)$$

where:

$$\bar{z} = \frac{1}{N} \sum_{k=0}^{N-1} z(k). \quad (13)$$

In general, the fuel cost of the UAV flying process is smallest when the UAV moves in the uniform rectilinear motion. Considering the flight efficiency of the UAV, the work state of the UAV is inefficient when the UAV does the turning maneuvers. Then, J_{smooth} is defined as follows:

$$J_{\text{smooth}} = \begin{cases} \sum_{k=1}^{N-2} \|\boldsymbol{\varphi}(k) - \boldsymbol{\varphi}(k-1)\|_2 & (k \geq 1) \\ 0 & (k = 0) \end{cases} \quad (14)$$

Of course, these three cost functions have some overlaps at the mathematical level to some extent. For example, the shortest path leads to the smoothest path. But all of these three functions has the different optimal solution sets and physical meanings. So they are all be retained as a part of the cost function J_{cost} . In short, the cost function J_{cost} represents a specific scenario which has the shortest path, the most stable flying height and the smoothest flying process, while respecting the UAV performance characteristic. The multi-objective cost function is comprehensive and complex for most real UAV path planning processes.

2.5. Optimization model formulation

In the area of the optimization, many problems are constrained by some constraint conditions. Generally speaking, this kind of problems can be defined as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}), \quad \mathbf{x} = [x_1, x_2, \dots, x_m] \\ \text{s.t.} \quad & h_{i_1}(\mathbf{x}) \leq 0, i_1 = 1, 2, \dots, p \\ & h'_{i_2}(\mathbf{x}) = 0, i_2 = 1, 2, \dots, q \end{aligned} \quad (15)$$

where $x_i \in [x_{i\min}, x_{i\max}]$, p is the number of the inequality constraints and q is the number of the equality constraints.

According to Sections 2.1–2.4, the optimization model of the path planning problem can be defined as follows:

$$\begin{aligned} \min_C \quad & J_{\text{cost}}(C), C = [\mathbf{g}^T(0), \mathbf{g}^T(1), \dots, \mathbf{g}^T(k), \dots, \mathbf{g}^T(N-1)]^T, \\ \text{s.t.} \quad & \mathbf{g}(k) \in H_{\text{worg}}(k) \in G_h, k = 0, 1, 2, \dots, N-1 \end{aligned} \quad (16)$$

where $C \in R_{N \times 3}$, $\mathbf{g}(k) \in R_{1 \times 3}$, $J_{\text{cost}}: R_{N \times 3} \rightarrow R$.

3. Wolf pack search algorithm

At present, there are many scholars who put forward some computational models which simulate the biological intelligence, which include the ant colony algorithm [24], the particle swarm optimization [25], the artificial fish school algorithm [26] and so on. The wolf pack search algorithm is a global probability search algorithm based on the hunting process of the wolf pack. In fact, the wolf pack search algorithm has good generality in many areas, but it is just used in few research fields now caused by novelty and un-mature. So in this section, the idea of the traditional wolf pack search algorithm is introduced into the UAV path planning problem. However, the traditional wolf pack search algorithm needs to be reinterpreted in this research area.

The wolf pack search algorithm is inspired by the uniform action of a wolf pack. They cooperate well with each other and attack the competitors and preys. Then, the whole process of the hunting activity of a wolf group can be summarized as the following process: firstly, the safari wolves need to be elected to walk around. Of course, the safari wolves are more experienced and stronger than the others for which the safari process is the most important action in the hunting activity; secondly, the safari wolves will search the smell around them. One of the safari wolves may find the thickest odor which means the quarries are around here; thirdly, it will inform all the other wolves by its howling. The whole wolf pack members will approach the safari wolf who finds the thickest odor of the quarries and will move towards the directions of the howling. At the same time, the quarries can be constrained in a smaller area; finally, some wolves may stop their hunting activity, for which their physical conditions are bad and they will die. Instead, some new wolves may join the wolf pack. So the members of the wolf pack are updated constantly during the hunting process.

Based on the above description of a wolf pack, the chromosomes C_i is introduced into the wolf pack search algorithm. A planning path created by a set of genes $\mathbf{g}(k)$ in the discrete planning space is regarded as a chromosome of a wolf (Fig. 6).

The frame of the wolf pack search algorithm based on the chromosomes is shown as follows:

Step 1: Determine the parameters of the wolf pack. Initialize the chromosomes C_i of the wolf pack randomly and $s \leftarrow -1$, where s means the number of cycles.

Step 2: Calculate the multi-objective cost function $J_{\text{cost}}(C_i)$ of each chromosome C_i and choose several chromosomes of the safari wolves who have lower cost function from the wolf pack.

Step 3: Randomly select a gene $\mathbf{g}^*(k)$ from the chromosome of each safari wolf. Then, the adjacent genes instead of the original gene produce some new chromosomes. Find the best one in these new chromosomes and the original one. The best one means the safari result of this safari.

Step 4: After the safari process, the chromosomes C_i in the planning space are updated. Calculate the cost function $J_{\text{cost}}(C_i)$ of all chromosomes and choose the best chromosome C_{best} from the new wolf pack:

$$C_{\text{best}} = \{C_b | J_{\text{cost}}(C_b) = \min_{i=1,2,\dots,M} J_{\text{cost}}(C_i)\}, \quad (17)$$

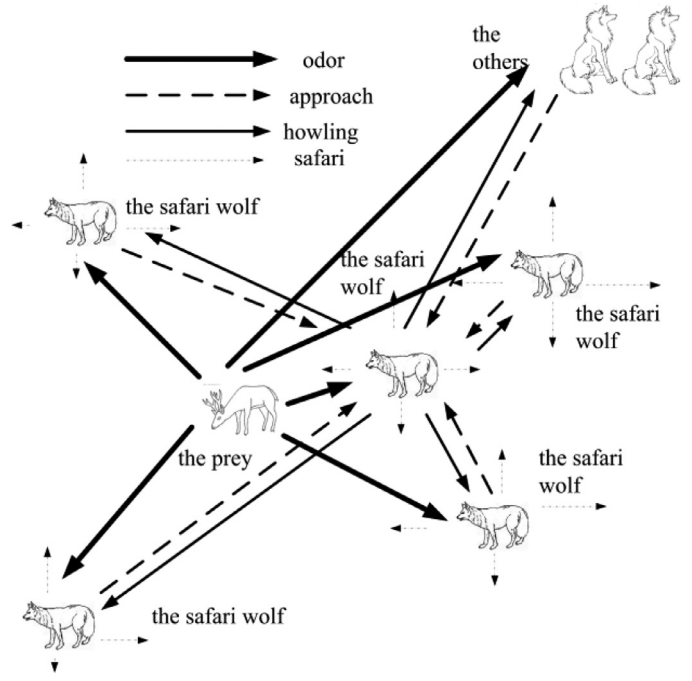


Fig. 6. The schematic diagram of the wolf pack search algorithm.

where M is the population size of the wolf pack.

Step 5: The whole wolf pack chromosomes will approach the best chromosome:

$$\begin{aligned} \mathbf{g}'_i(k) &= \mathbf{g}_i(k) + \frac{\mathbf{g}_{\text{best}}(k) - \mathbf{g}_i(k)}{|\mathbf{g}_{\text{best}}(k) - \mathbf{g}_i(k)|} \cdot \text{step}, \\ (k &= 0, 1, \dots, N-1; i = 1, 2, \dots, M; \text{step} \in Z), \end{aligned} \quad (18)$$

where $\mathbf{g}'_i(k)$ means the new k -th gene of the i -th chromosome, $\mathbf{g}_i(k)$ is the quondam k -th gene of the i -th chromosome, and $\mathbf{g}_{\text{best}}(k)$ is the k -th gene of the best chromosome C_{best} in the quondam wolf pack, step means the approaching velocity of the chromosome. step is defined by the human operator. If step is 1, which means the new gene $\mathbf{g}'_i(k)$ is the neighborhood point of the quondam gene $\mathbf{g}_i(k)$. The bigger step is, the faster convergence and the higher probability missing the optimal value will be. So the suitable step can avoid the premature convergence problem.

Step 6: Sort the new wolf pack. Delete some bad chromosomes, and randomly produce the new chromosomes whose amount is the same as the deleted chromosomes.

Step 7: If s is large enough or the other conditions are satisfied, stop the circulation. Otherwise, $s \leftarrow s+1$ and go back to Step 2.

4. Improved wolf pack search algorithm with crossover and mutation operations

Genetic algorithm that imitates the survival of the fittest is a stochastic and heuristic search method based on the mechanics of natural selection and genetics [27]. GA has become a powerful tool for solving the high computational complexity optimization problems. It can be used to search the good feasible solutions in a huge solution space, and converge to a good solution that may not be the optimal one. The most important basic operators in the GA are the selection, crossover and mutation operators. Thereinto, the selection operator represents the elitism thought of the GA. The selection operator part is abandoned in this improved method. The

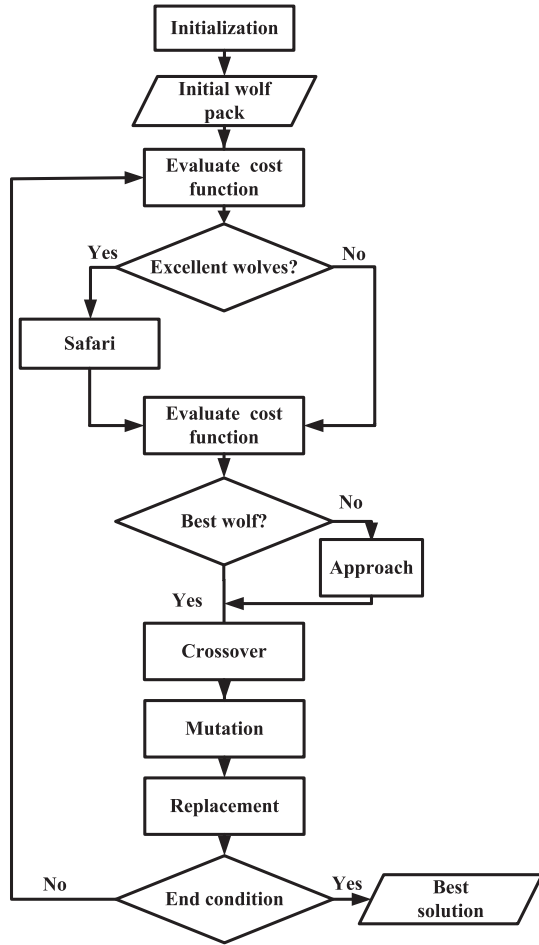


Fig. 7. The schematic diagram of the wolf pack search algorithm.

other two important basic operators fully embody the exchange of the excellent genes in the ordinary or eligible chromosomes. This power comes from the common individuals. At the same time, these two operators improve the communication among the whole population.

The original WPS algorithm is based on the safari wolves and the best wolf that are the elitism of the whole wolf pack. The common individuals which may be eligible are disregarded. Therefore, the thought of the crossover and mutation operators can just offset the shortage of the original WPS algorithm. In the improved WPS algorithm, the crossover and mutation operators in the GA method are introduced. A simple flow chart is illustrated in Fig. 7.

5. Path smoothing

After the application of the original and improved wolf pack search algorithm, our solution generates a path composed of the line segments. This path may be acceptable to the rotor wing UAV, but inadequate to the fixed wing UAV with the kinematic and dynamic constraints. Therefore, it is necessary to make the path smooth in order to be suitable for the fixed-wing UAVs.

Many methods can be considered for the path smoothing, for example: the B-spline curve, the Bezier curve, the clothoid curve, the simple circular arcs and so on [28–31]. They all need to meet the acceleration constraint under the known speed of the UAV. In so many ways, the B-spline curve has its characteristic advantages which include locality, geometrical invariability, continuity, symmetry, recursion, convexity-preserving and convex hull characters. It is well known that n B-spline curve has the high-degree con-

tinuous geometry (G^{n-h-1}) character, where h means the number of the coincident control points. The movement path of the fixed-wing UAVs needs to meet the G^2 continuous geometry at least. So the method we considered is the cubic B-spline curve (Fig. 8).

The cubic B-spline curve is built on some space control points. The quantity of the control points is always more than 3. The matrix representation of the cubic B-spline basis function $G_3(t)$ is [32]:

$$G_3(t) = \frac{1}{6} \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}, \quad t \in (0, 1). \quad (19)$$

The matrix expression of the cubic B-spline curve is:

$$\begin{cases} x(t) = G_3 \mathbf{X} & \mathbf{X} = [x_j, \dots, x_{j+3}]^T \\ y(t) = G_3 \mathbf{Y} & \mathbf{Y} = [y_j, \dots, y_{j+3}]^T \\ z(t) = G_3 \mathbf{Z} & \mathbf{Z} = [z_j, \dots, z_{j+3}]^T \end{cases}, \quad j = 1, 2, \dots, N_c - 3, \quad (20)$$

where $[x_j, y_j, z_j]$, $[x_{j+1}, y_{j+1}, z_{j+1}]$, $[x_{j+2}, y_{j+2}, z_{j+2}]$, $[x_{j+3}, y_{j+3}, z_{j+3}]$ are the coordinates of the four corresponding control points; j denote the sequence numbers of the control points; N_c means the total number of the control points; $x(t)$, $y(t)$, $z(t)$ mean the coordinates of the cubic B-spline curve when t changes from 0 to 1.

In order to apply the cubic B-spline curve to some special situations whose number of the control points are less than 4, two control points x_0, x_{N_c+1} are added as follows:

$$\begin{cases} x_0 = x_1 \\ x_{N_c+1} = x_{N_c} \end{cases}. \quad (21)$$

It is known that the acceleration of the UAV is related to the curvature of the path and the speed of the UAV. For simplicity, suppose that the velocity of the UAV is constant. So the relationship among the acceleration, the curvature radius and the velocity is:

$$a = \frac{v^2}{\rho}, \quad (22)$$

where v is the velocity of the UAV, ρ is the curvature radius of the space curve [33]. Considering the maximum acceleration constraint, the curvature radius needs to satisfy:

$$\rho_{\min} = \min_{t \in [0,1]} \rho(t) \geq \frac{v^2}{a_{\max}}, \quad (23)$$

where ρ_{\min} is the minimum curvature radius, a_{\max} denotes the maximum acceleration of the UAV, and $\rho(t)$ means the curvature radius varying with t :

$$\begin{aligned} \rho(t) &= \frac{E(t)^{3/2}}{\sqrt{A(t)^2 + B(t)^2 + D(t)^2}}, \\ A(t) &= z''(t)y'(t) - y''(t)z'(t), \\ B(t) &= x''(t)z'(t) - z''(t)x'(t), \\ D(t) &= y''(t)x'(t) - x''(t)y'(t), \\ E(t) &= x'^2(t) + y'^2(t) + z'^2(t). \end{aligned} \quad (24)$$

In other words, the maximum acceleration of the UAV depends on the control points of the cubic B-spline curve. So if the maximum acceleration of the UAV is limited, the control points also need to be limited.

6. Simulation result

Because the planning space can be divided into the real 3D space and fake 3D space, two simulation scenarios need to be demonstrated to verify the effectiveness of our modified wolf pack search algorithm.

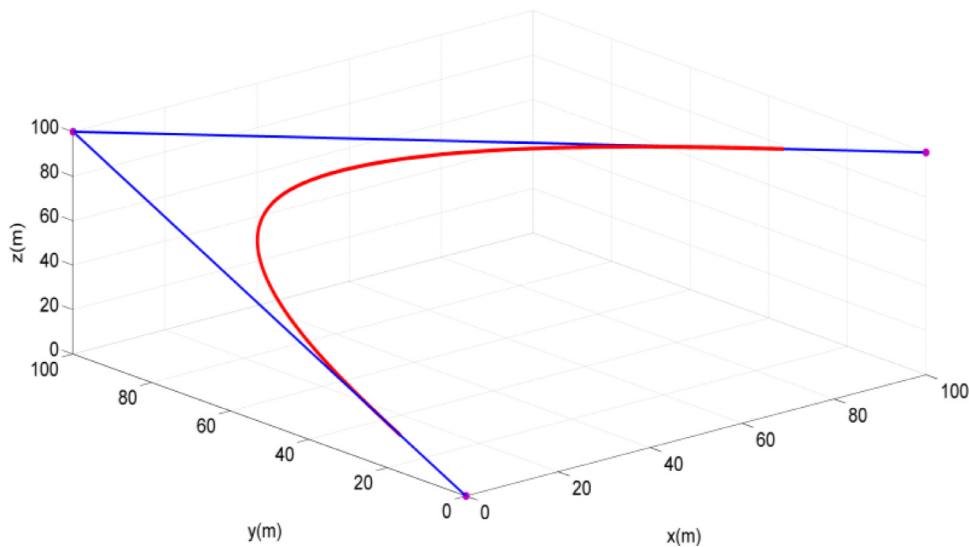


Fig. 8. The cubic B-spline curve.

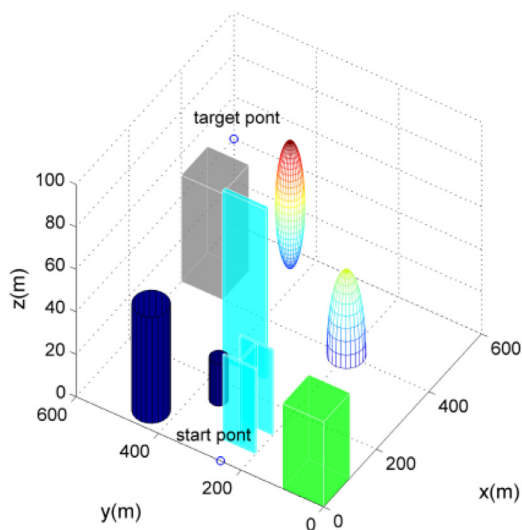


Fig. 9. The planning space 1. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

6.1. Real 3D space simulation

The simulation parameters and the simulation constraint conditions in the examples are as follows:

The length, width and height of the planning space are 600 m × 600 m × 100 m. The starting point coordinates value of the UAV is (0 m, 250 m, 0 m). The target point coordinates are (600 m, 600 m, 40 m). The abstract scene is shown in Fig. 9 [34].

The waypoints in the real 3D space are distributed uniformly. They are distributed in a constant law that can define their distance between each two waypoints. Then, find out the healthy genes (the green points) and delete the others (the red points). In this paper, the waypoints are distributed as follows (Fig. 10):

After these genes are distributed, the modified wolf pack search algorithm is introduced to solve the UAV path planning problem. The initial parameters of the modified wolf pack search algorithm are shown in Table 2:

The modified wolf pack search algorithm as a global probability search algorithm always leads to some random results. Therefore, the simulation results are different often. The randomness of the

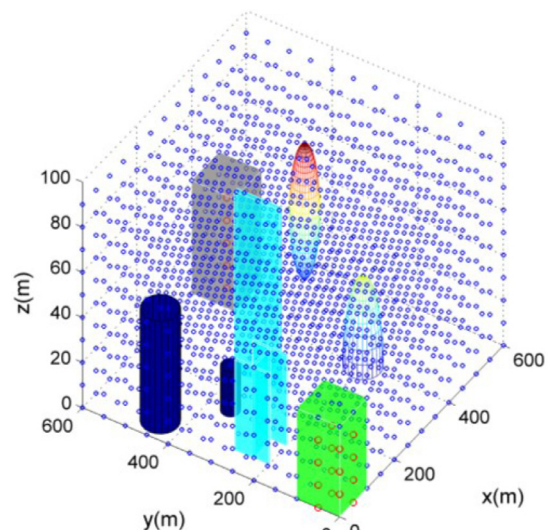


Fig. 10. The dispersed waypoints in the planning space. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

Table 2
Initial parameters 1.

| Parameter | Value | |
|-----------------------------|---------------------|-----------|
| Speed range | Rotor wing UAV | 0–30 m/s |
| | Fixed-wing UAV | 20–50 m/s |
| Maximum acceleration | 50 m/s ² | |
| Population quantity | 20 | |
| Maximum number of iteration | 100 | |
| Safari wolves quantity | 5 | |
| Eliminated quantity | 5 | |
| w1 | 0.5 | |
| w2 | 0.2 | |
| w3 | 0.3 | |

results is from the random initial state. Further, if the initial state of the wolf pack is close to the prey, the result may be good this time. Taken this case into consideration, the simulations need to be repeated for 10 times. Then, the best cost function values in 10 wolf packs changing with the generation are shown as follows (Fig. 11):

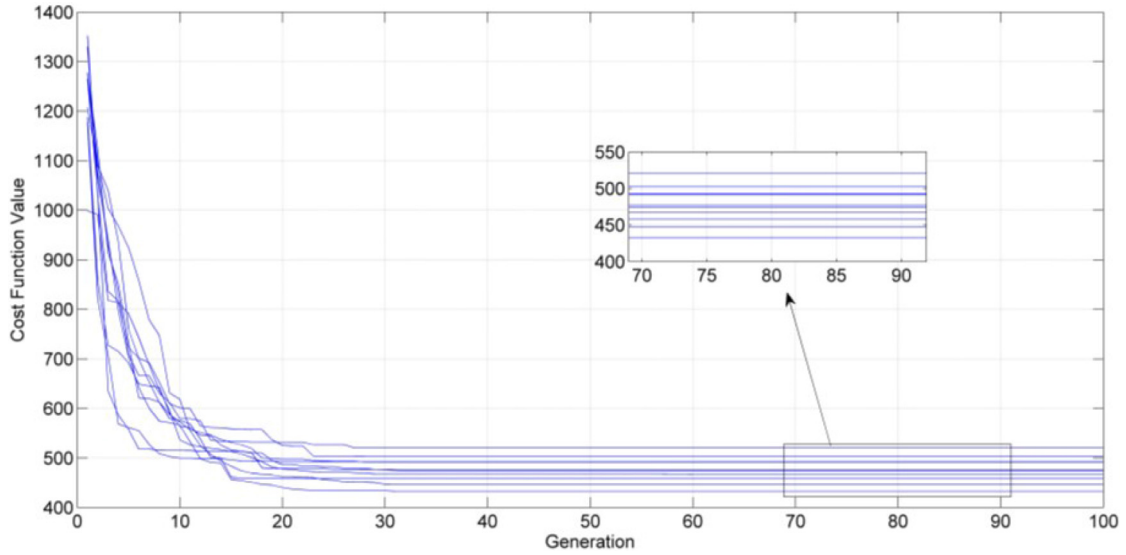


Fig. 11. The best cost function values of 10 different wolf packs.

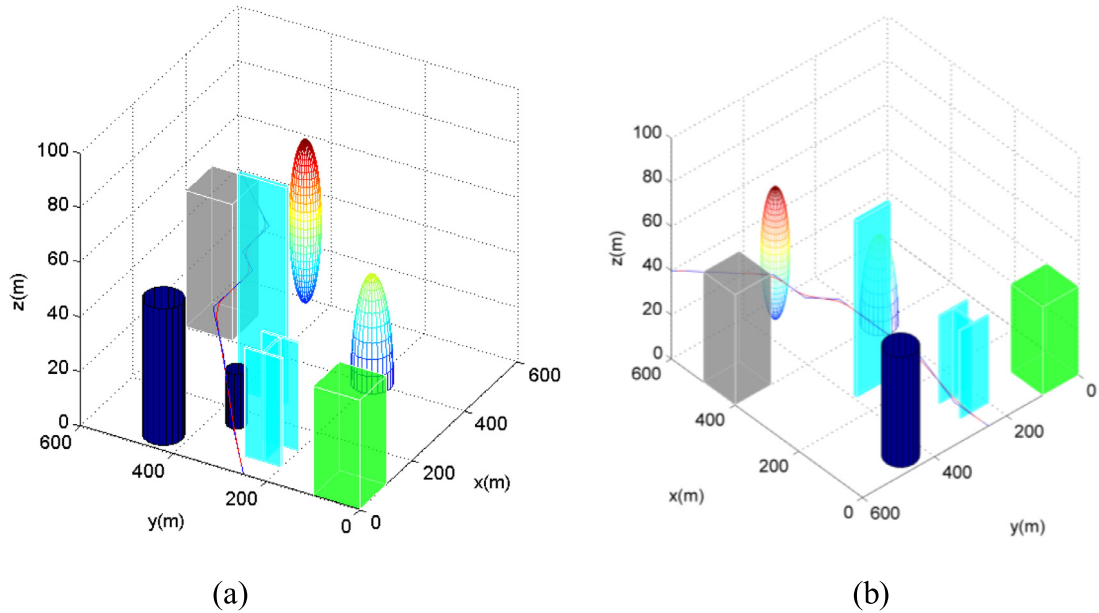


Fig. 12. The results of the real 3D space simulation. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

We can see that the best cost function values of the wolf packs reduce quickly at the initial phase (about 1–10 generations). Then, the cost function values converge to their convergence values at last (above 25 generations) with the number of generations increasing. The convergence values of these 10 wolf packs are limited in the range of [400, 550]. Viewed from the rate of convergence, the modified WPS algorithm is an efficient optimization method.

The best one in the simulation group which includes 10 simulation examples is as follows:

In Fig. 12, we can know that the planning path avoids all obstacles, which means its cost function value is small. The cost function value of this simulation result is 432.8. Of course, the rotor wing UAV can follow the blue path immediately.

After the process of path smoothing, the red line is obtained. Considering the kinematic and dynamic constraints, the fixed-wing UAV cannot follow the blue line. If it flies following the blue line, the acceleration will be infinite. So the fixed-wing UAV can

use the maximum speed 50 m/s to follow the red line. The acceleration is as follows:

From Figs. 12 and 13, we can see that the best planning result is reasonable and suitable for the rotor wing UAV and fixed-wing UAV which verifies the correctness of the modified WPS algorithm in the real 3D space.

6.2. Fake 3D space simulation

The simulation parameters and the simulation constraint conditions in the examples are as follows:

The length, width and height of the planning space are 8000 m × 8000 m × 1200 m. We can see that it is a large planning space and it is suitable for the fake 3D situation. It is observed that this planning space imitates the digital map. Then, the starting point coordinates value of the UAV is (0 m, 500 m, 350 m). The target point coordinates are (7500 m, 7500 m, 300 m). The abstract scene is shown in Fig. 14.

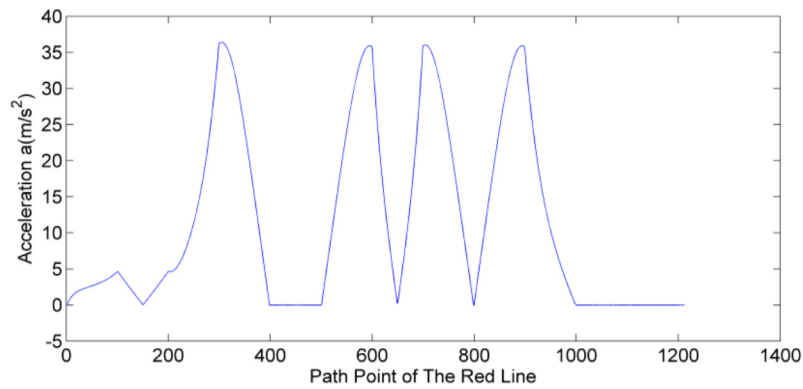


Fig. 13. The acceleration of the UAV.

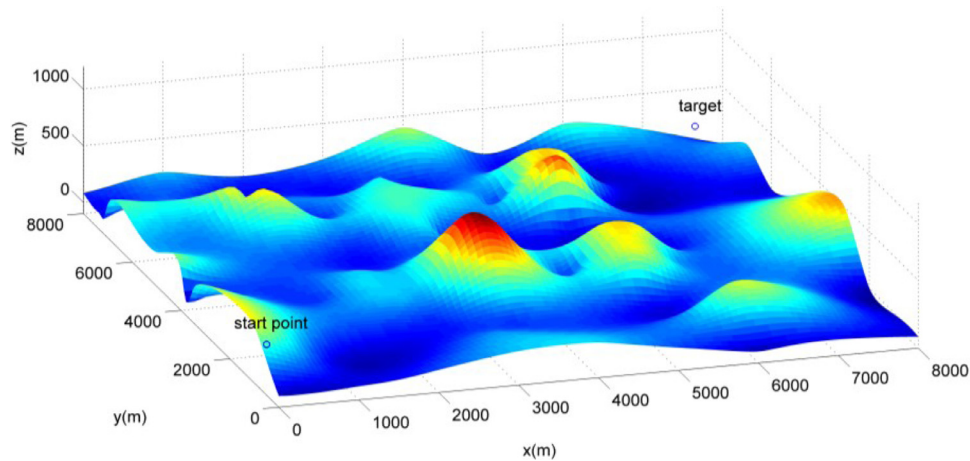


Fig. 14. The planning space 2. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

Table 3
Initial parameters 2.

| Parameter | Value |
|-----------------------------|----------------------|
| Speed range | 100–200 m/s |
| Maximum acceleration | 100 m/s ² |
| Population quantity | 10 |
| Gene number | 30 |
| Maximum number of iteration | 100 |
| Safari wolves quantity | 5 |
| Eliminated quantity | 5 |
| w1 | 0.5 |
| w2 | 0.2 |
| w3 | 0.3 |

The waypoints in the fake 3D space are created according to the obstacles in the planning space. So all waypoints in the fake 3D space are the healthy genes (red points). The modified wolf pack search algorithm is introduced to solve the fake 3D UAV path planning problem. In the large planning space, the high-speed fixed-wing UAV meets this large scale working environment. Then the initial parameters of the wolf pack search algorithm are shown in Table 3:

Similarly, because of the randomness of the modified wolf pack search algorithm, the simulation results in the fake 3D space are also different often. By thinking about this case, the simulations also need to be repeated for many times. Then, the best cost function values of a simulation group are shown as follows (Fig. 15):

With the number of generations increasing, the best cost function values quickly converge to their equilibrium values. Most of the convergence generation numbers are below 60 generations. It

could be explained that the modified wolf pack search algorithm has superior convergence. In the simulation result, the equilibrium cost function values range from 24,962 to 27,986.

The best one in the simulation group which includes 10 simulation examples is as follows (Fig. 16):

The black star-like points mean the genes of the best chromosome, and the black line means the whole chromosome of the best wolf in the whole wolf pack after 100 generations breeding. Then, the red line means the feasible real flying path for the high-speed fixed-wing UAV. Because the scale of the fake 3D space is always large, the big turning radius helps to limit the acceleration of the high speed fixed-wing UAV in a reasonable range easily. When the fixed-wing UAV uses the maximum speed 50 m/s to follow the red line, its accelerations of the whole path are less than 20 m/s².

6.3. Comparison

In order to further verify the effectiveness of the method, the comparisons of the traditional GA, the random search way, the original WPS algorithm and the modified WPS algorithm are introduced under the same conditions. In addition, the mutation probability of the GA is 0.01. The simulation scenario is the real 3D space simulation. Under the same population quantity (20 individuals) and the maximum iteration number (100 generations), the convergent cost functions (10 times) of the three methods are as follows:

In Fig. 17, we can see that the convergent cost functions of the modified WPS algorithm are better than the ones of the others obviously. The sequence of the average convergence rates is as follows: the GA, the modified WPS algorithm, the original WPS algo-

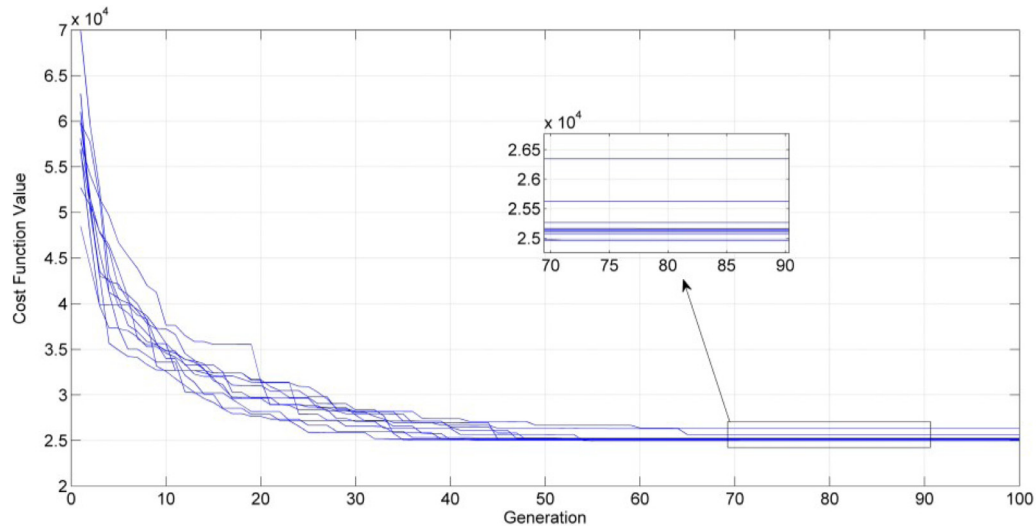


Fig. 15. The best cost function values of 10 different wolf packs.

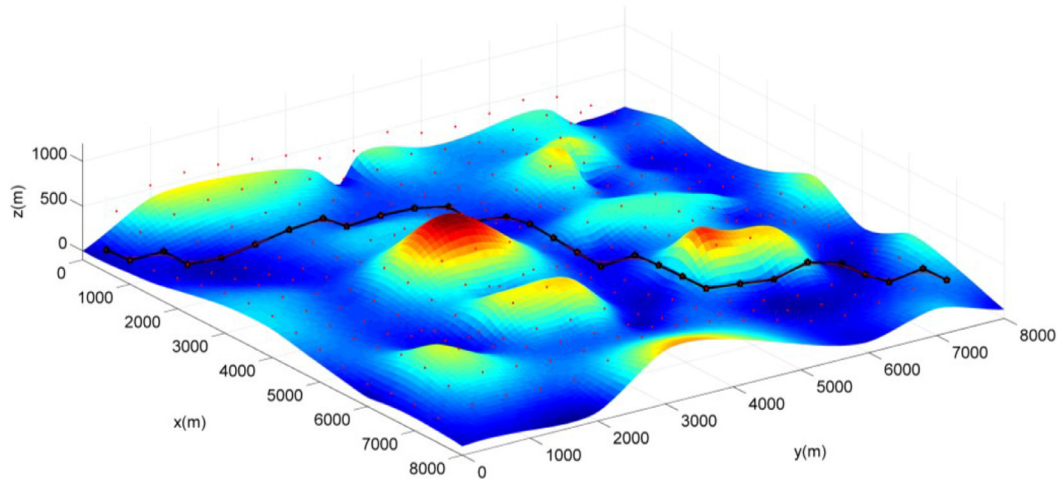


Fig. 16. The planning result of the fake 3D space simulation. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

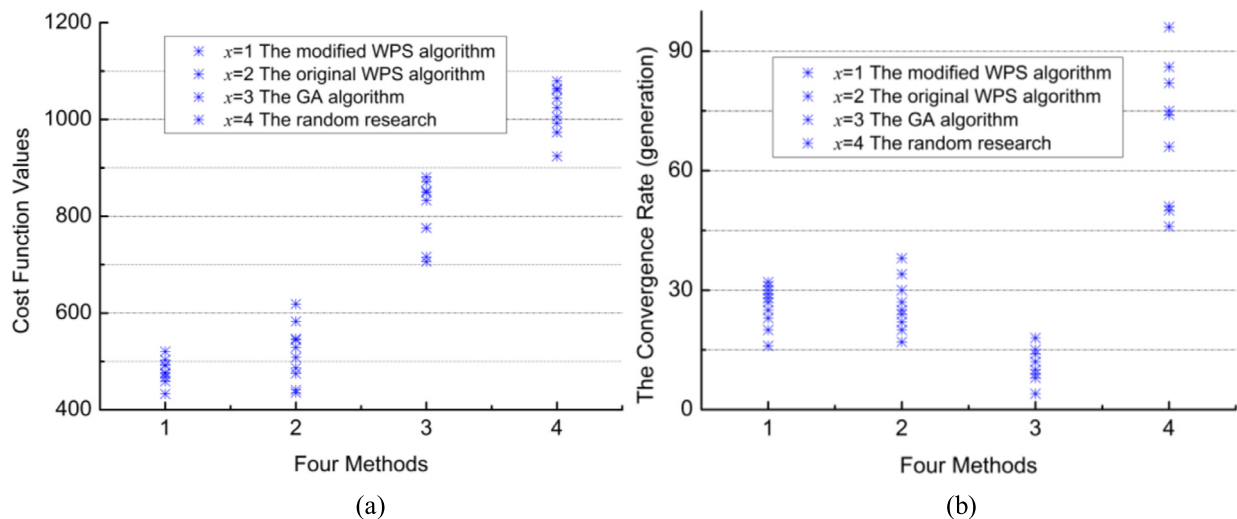


Fig. 17. The comparison among the three methods.

Table 4
simulation system.

| Parameter | Value |
|-----------|--|
| Computer | Aspire 4750 |
| CPU | Intel(R) Core(TM) i3-2350 M CPU @ 2.30 GHz |
| RAM | 4.00 GB, DDR3, 1333 MHz |
| OS | Windows 7 Enterprise, 64 bits |
| IDE | MATLAB R2011b |

rithm and the random search way. So we can see that the traditional GA appears the premature problem. Above all, the modified WPS algorithm has good global search ability and strong convergence ability. Moreover, it can effectively prevent the premature problem.

Then, the computing time analysis of these algorithms is presented. The computational time of an optimization method depends on the complexity of the problem, computer performance and the program structure. The program structures of four algorithms are similar. At the same time, the information of the complexity of the problem and the computer performance is provided as follows.

The simulation system is described in Table 4:

The average simulation times, which includes the obstacle description, optimal process and path smoothing process (without plotting), of these four algorithms in 10 times, respectively, are 149.7 s (modified WPS method), 107.9 s (original WPS method), 120.1 s (traditional GA method) and 51.3 s (random search way). They are obtained by the library function of the MATLAB (tic and toc function). We can see that even though the optimization effect of the modified WPS method is best in four algorithms, it cost the longest simulation time. Of course, the additional cost is acceptable.

6.4. Robustness

In order to verify the robustness of the modified WPS algorithm, the approaching velocity of the chromosome *step* is used for the parameter variation. In Section 3, The approaching velocity *step* of the original manuscript is 1, which means which means the new gene $\mathbf{g}'_i(k)$ is the neighborhood point of the quondam gene $\mathbf{g}_i(k)$. Because the modified WPS algorithm in this paper is closely combined with the path planning problems, the approaching velocity *step* has the fixed physical meaning, which leads that it can only be a positive integer. If the modified WPS method is used in an optimization problem whose independent variables are continuous, the approaching velocity *step* can be more diverse. In our paper, when the approaching velocity *step* is bigger than 1, it means that the interval between the new gene $\mathbf{g}'_i(k)$ and the quondam gene $\mathbf{g}_i(k)$ is *step* in the vertical and lateral directions. The simulation scenario is the real 3D space simulation. The approaching velocity *step* ranges from 1 to 4. Then, under the same parameters, the best ones in 4 groups (Each group includes 10 simulations) are shown as follows:

In Fig. 18, the blue line, green line, purple line and yellow line represent the approaching velocities are 1, 2, 3 and 4, respectively, and their cost function values, respectively, are 432.8, 436.4, 465.6 as well as 441.0. We can see that the best ones in 4 groups all finish the path planning successfully. Their cost functions exist some acceptable gaps. The convergent cost functions (10 times) of the four groups are as follows:

In Fig. 19, we can see that when the approaching velocity is bigger than 3, the simulation results become unstable. The main reason for this phenomenon is that the too big approaching velocity will miss many search ranges in the domain of definition, which makes the higher demand for the initial wolf pack. For this

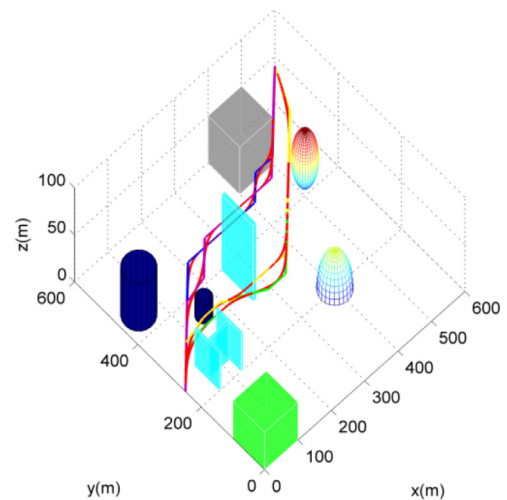


Fig. 18. The best result in four groups. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

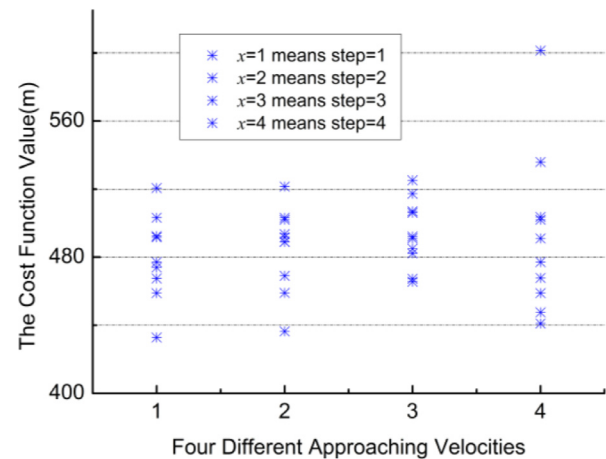


Fig. 19. The cost function values of four groups.

real 3D space problem, the optimization robustness is good when the approaching velocity is smaller than 3 (include 3).

7. Conclusion

- (1) Considering the different description ways of the obstacles in the planning space, the 3D UAV path planning problem which includes the fake 3D problem and the real 3D problem are developed and studied in this paper. The object is to minimize the multi-objective cost function which includes the minimization of the path length, the minimization of the fuel cost and the best smoothness of the planning path.
- (2) At first, the concepts of the chromosome and the gene are introduced into the traditional wolf pack search algorithm. Based on the principle that a chromosome represents a wolf, the operations of the wolf pack are changed into the update of a population. Then, the crossover and mutation operators in the GA are introduced in the modified wolf pack search algorithm.
- (3) In view of the kinematic and dynamic constraints of the UAV, the line segments are smoothed by the cubic B-spline curve. The control points of the cubic B-spline curve are determined by the acceleration limit of the fixed-wing UAV.
- (4) The simulation results show that the modified WPS algorithm can solve the 3D UAV path planning successfully regardless of the different 3D space types. Its performance in these problems is more outstanding than the original WPS algorithm, the GA

and the random search way obviously. Then, the further simulation results show that the modified WPS algorithm has a high robustness ability.

Reference

- [1] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, *Int. J. Robot. Res.* 5 (1) (1986) 90–98.
- [2] M. Garcia, A. Viguria, A. Ollero, Dynamic graph-search algorithm for global path planning in presence of hazardous weather, *J. Intell. Robot. Syst.* 69 (2013) 285–295.
- [3] J.L. Peng, X.X. Sun, F. Zhu, J. Zhang, 3-D path planning with multi-constraints based on genetic algorithm, in: *Proceedings of the 27th Chinese Control Conference, CCC 2008*, Kunming, China, 2008, pp. 94–97.
- [4] Z.H. Peng, L. Sun, J. Chen, J.P. Wu, Path planning of multiple UAVs low-altitude penetration based on improved multi-agent coevolutionary algorithm, in: *Proceedings of the 30th Chinese Control Conference 2011*, Yantai, China, 2011, pp. 4056–4061.
- [5] V. Roberge, T. Mohammed, L. Gilles, Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning, *IEEE Trans. Ind. Inform.* 9 (1) (2013) 132–141.
- [6] O. Cetin, I. Zagli, G. Yilmaz, Establishing obstacle and collision free communication relay for UAVs with artificial potential fields, *J. Intell. Robot. Syst.* 69 (2013) 361–372.
- [7] A. Richards, T. Schouwenaars, J.P. How, F. Eric, Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming, *J. Guid. Control Dyn.* 25 (4) (2002) 755–764.
- [8] W.A. Kamal, D.W. Gu, I. Postlethwaite, MILP and its application in flight path planning, in: *Proceedings of the 16th IFAC World Congress*, 2005, Czech Republic, 2005, pp. 2060–2066.
- [9] E.I. Grøtli, T.A. Johansen, Path planning for UAVs under communication constraints using SPLAT! and MILP, *J. Intell. Robot. Syst.* 65 (2012) 265–282.
- [10] B. Miller, K. Stepanyan, A. Miller, M. Andreev, 3D path planning in a threat environment, in: *Proceedings of the 2011 50th IEEE Conference on Decision and Control and European Control Conference, CDC-ECC*, Orlando, USA, 2011, pp. 6864–6869.
- [11] A. Heydari, S.N. Balakrishnan, Path planning using a novel finite horizon sub-optimal controller, *J. Guid. Control Dyn.* 36 (4) (2013) 1210–1214.
- [12] P.B. Sujit, R. Beard, Multiple UAV path planning using anytime algorithms, in: *Proceedings of the 2009 American Control Conference*, St. Louis, USA, 2009, pp. 2978–2983.
- [13] K.P. Valavanis, *Advances in Unmanned Aerial Vehicles: State of the Art and Road to Autonomy*, Springer, Florida, 2007.
- [14] P. Bhattacharya, M.L. Gavrilova, Roadmap-based path planning using the Voronoi diagram for a clearance-based shortest path, *IEEE Robot. Autom. Mag.* 15 (2) (2008) 58–66.
- [15] K. Yang, S.K. Gan, S. Sukkarieh, A Gaussian process-based RRT planner for the exploration of an unknown and cluttered environment with a UAV, *Adv. Robot.* 27 (6) (2013) 431–443.
- [16] S. Karaman, M.R. Walter, A. Perez, E. Frazzoli, S. Teller, Anytime motion planning using the RRT*, in: *Proceedings of IEEE International Conference on Robotics and Automation, ICRA*, Shanghai, China, 2011, pp. 1478–1483.
- [17] S. Ragi, E.K.P. Chong, Dynamic UAV path planning for multi-target tracking, in: *Proceedings of the 2012 American Control Conference*, Montréal, Canada, 2012, pp. 3845–3850.
- [18] S. Ragi, E.K.P. Chong, UAV path planning in a dynamic environment via partially observable Markov decision process, *IEEE Trans. Aerosp. Electron. Syst.* 49 (4) (2013) 2397–2412.
- [19] X.Y. Xu, J. Xie, K.M. Xie, Path planning and obstacle-avoidance for soccer robot based on artificial potential field and genetic algorithm, in: *Proceedings of World Congress on Intelligent Control and Automation, WCICA 2006*, Dalian, China, 2006, pp. 3494–3498.
- [20] C.G. Yang, X.Y. Tu, J. Chen, Algorithm of marriage in honey bees optimization based on the wolf pack search, in: *Proceedings of the 2007 International Conference on Intelligent Pervasive Computing*, Jeju City, Korea, 2007, pp. 462–467.
- [21] H.A. Abbass, Marriage in honey bees optimization (MBO): a haplotetrisis polygynous swarming approach, in: *Proceedings of Congress on Evolutionary Computation, CEC 2001*, Seoul, Korea, 2001, pp. 207–214.
- [22] T. Shima, S.J. Rasmussen, A.G. Sparks, UAV cooperative multiple task assignments using genetic algorithms, in: *Proceedings of American Control Conference*, 2005, Portland, USA, 2005, pp. 2989–2994.
- [23] Y.Y. Wang, T.T. Wei, X.J. Qu, Study of multi-objective fuzzy optimization for path planning, *Chin. J. Aeronaut.* 25 (1) (2012) 51–56.
- [24] C. Zhang, Z.Y. Zhen, D.B. Wang, M. Li, UAV path planning method based on ant colony optimization, in: *Proceedings of Conference on Control and Decision Conference, CCDC*, 2010, USA, 2010, pp. 3790–3792.
- [25] J.C. Cuo, J.J. Wang, G.Z. Cui, Online path planning for UAV navigation based on quantum particle swarm optimization, in: *Proceedings of the 2009 3rd International Conference on Teaching and Computational Science – Advanced Technology in Teaching*, 2009, pp. 291–302.
- [26] B. Wu, Y.W. Zhao, W.L. Wang, Artificial fish school algorithm for vehicle routing problem, in: *Proceedings of the 3rd International Conference on Electronic Commerce Engineering*, 2003, pp. 329–331.
- [27] Q.B. Deng, J.Q. Yu, N.F. Wang, Cooperative task assignment of multiple heterogeneous unmanned aerial vehicles using a modified genetic algorithm with multi-type genes, *Chin. J. Aeronaut.* 26 (5) (2013) 1238–1250.
- [28] P. Jacobs, J. Canny, Planning smooth paths for mobile robots, in: *Proceedings of IEEE International Conference on Robotics and Automation*, Scottsdale, USA, 1989, pp. 271–342.
- [29] K. Yang, S. Sukkarieh, 3D smooth path planning for a UAV in cluttered natural environments, in: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008*, Nice, France, 2008, pp. 794–800.
- [30] X. Li, J. Xie, M.Y. Cai, M. Xie, Path planning for UAV based on improved heuristic A* algorithm, in: *Proceedings of Electronic Measurement & Instruments, 2009, ICEMI'09*, 9th International Conference on, Beijing, China, 2009, pp. 488–493.
- [31] K. Yang, S.K. Gan, S. Sukkarieh, An efficient path planning and control algorithm for RUAV's in unknown and cluttered environments, *J. Intell. Robot. Syst.* 57 (2013) 101–122.
- [32] A.H. Ren, *Computer Graphics*, Springer-Beihang University Press, Beijing, 2005.
- [33] K.R. Kim, P.T. Kim, J.Y. Koo, Frenet-serret and the estimation of curvature and torsion, *IEEE J. Sel. Top. Signal Process.* 7 (4) (2013) 646–656.
- [34] Y. Chen, G. Luo, Y.S. Mei, J.Q. Yu, X.L. Su, UAV path planning using artificial potential field method updated by optimal control theory, *Int. J. Syst. Sci.* 47 (6) (2016) 1407–1420.



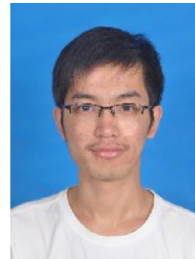
Chen Yongbo received his B.S. degree in Beijing Institute of Technology in 2012. He is currently working toward a Ph.D. degree at the School of Aerospace Engineering, Beijing Institute of Technology, Beijing, China. His research interests include UAV path planning, multi-UAV path planning and UAV mission planning.



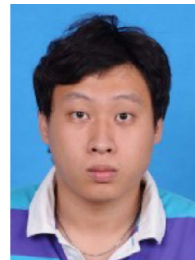
Mei YueSong was born in 1978. He received his Ph.D. degree in flight vehicle design from Beijing Institute of Technology in 2007. He is now a lecture of School of Aerospace Engineering, Beijing Institute of Technology. His research interests include flight vehicle system design, flight dynamics and control.



Yu JianQiao received B.S., M.S. and Ph.D. degrees from Beijing Institute of Technology in 1994, 1997 and 2007, respectively. He is presently a Professor with the school of Aerospace Engineering, Beijing Institute of Technology. His main research interests include flight dynamics and control, cooperative control, flight vehicle system design and robust control.



Su XiaoLong received his B.S. degree in Beijing Institute of Technology in 2012. He is a Post Graduate in flight vehicle design in Beijing Institute of Technology. His research interests include flight dynamics and control and flight vehicle system design.



Xu Nuo received his B.S. degree in Xian University of Architecture and Technology in 2008. He is currently working toward a Ph.D. degree at the School of Aerospace Engineering, Beijing Institute of Technology, Beijing, China. His research interests include flight dynamics and control and flight vehicle system design.