Rotary unmanned aerial vehicles path planning in rough terrain based on multi-objective particle swarm optimization

XU Zhen¹, ZHANG Enze²*, and CHEN Qingwei¹

¹. School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China; ₂. College of Information Engineering, Yangzhou University, Yangzhou 225009, China

Abstract: This paper presents a path planning approach for rotary unmanned aerial vehicles (R-UAVs) in a known static rough terrain environment. This approach aims to find collision-free and feasible paths with minimum altitude, length and angle variable rate. First, a three-dimensional (3D) modeling method is proposed to reduce the computation burden of the dynamic models of R-UAVs. Considering the length, height and tuning angle of a path, the path planning of R-UAVs is described as a tri-objective optimization problem. Then, an improved multi-objective particle swarm optimization algorithm is developed. To render the algorithm more effective in dealing with this problem, a vibration function is introduced into the collided solutions to improve the algorithm efficiency. Meanwhile, the selection of the global best position is taken into account by the reference point method. Finally, the experimental environment is built with the help of the Google map and the 3D terrain generator World Machine. Experimental results under two different rough terrains from Guilin and Lanzhou of China demonstrate the capabilities of the proposed algorithm in finding Pareto optimal paths.

Keywords: unmanned aerial vehicle (UAV), path planning, multi-objective optimization, particle swarm optimization.

DOI: 10.21629/JSEE.2020.01.14

1. Introduction

With the advancement of technology and the rise in labor price, humans are being replaced by autonomy from dangerous and heavy work. In various actual applications, such as forest patrol [1], electric power facilities diagnostic [2], fire dispatching [3], rescue after detection [4] and danger area detection [5], unmanned aerial vehicles (UAVs) have unmatchable advantages over ground robots due to their higher working space and degrees of freedom (DOF). Meanwhile, the free hovering and unconstrained movement features of R-UAVs have irreplaceable advantages over the fixed-wing UAVs. With the rapid development of battery technology and flight control technology, R-UAV has become one of the most popular development tendencies in the field of robotics [6,7]. As one of the most important research themes of the mobile robotics field, the path planning problem of R-UAVs in rough terrains is studied in this paper.

The purpose of path planning is to find a superior path from an initial position to destination without any collision according to certain evaluation standards and constrains. The existing work can be classified into two main groups: classical and heuristic. Among the classical methods, the most representative ones are the skeleton approach [8], cell decomposition [9] and potential field approaches [10], etc. These approaches have their own advantages, however, most of them are computationally expensive and can possibly fall into local minima [11]. Heuristic algorithms, which are based on the cognitive process in nature, have provided an alternative way to solve robot path planning problems. Part heuristic algorithms include genetic algorithms (GA) [12], ant colony optimization (ACO) [13], particle swarm optimization (PSO) [14], rapidly-exploring random trees (RRT) [15], etc. Recent work has demonstrated the superiority of these heuristic algorithms over classical ones in dealing with path planning problems, especially for an uneven rough terrain [16].

PSO, first introduced by Kennedy and Eberhart [17], is a population-based heuristic search technique inspired by the paradigm of birds flocking. Due to its simplicity, fast convergence and population-based feature, PSO has been successfully used for solving the path planning problems of UAVs [11,14,18]. Despite their efforts, these methodologies only consider one objective, or handle the path planning problem as a single-optimization problem by using the weighted sum method. However, multiple objectives such as the length, smoothness and safety of a path should be considered simultaneously in practical path plan-
ning problems. Moreover, the path planning aims at providing a set of equally optimal paths instead of a single one for decision makers to choose according to various tasks. Thus, the path planning in a known rough terrain is better to be considered as a multi-objective optimization problem.

Multi-objective PSO (MOPSO) has been applied to plan the path in different scenarios. Geng et al. [19] applied MOPSO in uncertain environments comparing with centralized algorithms. Geng and Gong [20] applied interval PSO by combining the local optimal criterion in multiple obstacle environments. MOPSO is applied in robot path planning for multi-survivor rescue in limited survival time [21]. Robot path planning is solved in an environment with terrains based on interval MOPSO [22]. From previous studies, we can conclude that even though the path planning problem has been studied for years, dealing with rough terrain scenarios is still an open problem.

In the current work, the path planning problem for R-UA Vs is described as a tri-objective optimization problem, considering simultaneously length, height and tuning angle of a path. An improved MOPSO algorithm is proposed to tackle the problem. The major contributions of this study are as follows:

(i) A three-dimensional (3D) modeling method for trajectories is adopted. This model combines the polar coordinates with the continuous two-dimensional (2D) path planning method to reduce the particle dimension. In the meantime, the angles in this modeling method correspond to the pitch and yaw angles of R-UA V, thus reducing the computation burden and saving the information extraction time of R-UAV platforms.

(ii) A vibration function is introduced into the collided solutions to improve the algorithm efficiency. The collided fragment from an infeasible path is selected and adjusted according to the vibration factor which represents the infeasibility degree of a fragment.

(iii) The reference point method is adopted to guide the algorithm to move toward the preferred region of the objective space, thus providing Pareto optimal paths for decision makers to choose from according to the need of UAV tasks.

(iv) The effectiveness of the proposed approach is validated through experiments conducted in an environment based on true rough terrains.

The remainder of this paper is organized as follows. Section 2 provides details of the modeling of terrains, the encoding of the trajectories, and the evaluative functions for candidate trajectories. In Section 3, the working principle of PSO is described in the beginning, and we present the use of the vibration function and the reference point method respectively. In Section 4, two cases are discussed. This section demonstrates simulation results on the proposed algorithm, and the initial MOPSO algorithm and NSGA-II are used as contrastive experiments to validate the performance of the proposed algorithm. Section 5 concludes this paper.

2. Problem formulation

In 3D modeling of the path planning problem, two key issues need to be considered. First, there is no obvious shortcut in the terrain. Second, code the particle efficiently. Obviously, the two key issues have internal connections. In order to discuss them unambiguously, the two parts are interspersed with each other as shown below. It is worth mentioning that when we use R-UAV as the model, the flight space and angle will be unlimited in principle. Therefore, on the premise of setting aside a safe distance, the author regards R-UAV as a mass point without volume constraint.

2.1 Modeling of the terrain

Some authors use the satellite or radar in terrain modeling for reference, as shown in Fig. 1 [14]. This is a fantastic perspective different from the traditional Euclidean coordinate system and the earth’s curvature is considered when considering long voyages. In this paper, we discuss short distance flight, so the ground can be regarded as a plane. Based on this assumption, we adjust for the modeling approach to Fig. 2 called decomposable polar method (DPM).
We can see from Fig. 1 that the information in the third quadrant is least wanted, because we consider the path planning problem as a bounded convex optimized problem. Thus, we extract the information in the first quadrant to model the terrain which is shown in Fig. 2.

Considering points O and J as the origin and destination respectively and OJ as the bisector of ∠xoy. The space in Fig. 2 can be segmented into n parts slantwise and each of them is perpendicular with OJ axis. The thickness of each piece, after discretization, is \( r = \frac{1}{n} OJ \), which means \( n \) steps are needed for UAV to reach the destination. The size of \( n \) is determined by map accuracy and the requirements of path accuracy. The bigger \( n \) is, the better the solution fits the actual flight path. Considering the resolution of the images in this paper, \( n = 9 \) can achieve good enough accuracy. \( P_i \) is a path point in the \( i \)th section, and \( L_i \) is the projection of \( P_i \) onto the surface \( XOY \). Thus \( P_i \) can be expressed by \( (\rho_{1i}, \rho_{2i}, ir)^T \), where

\[
\begin{align*}
\rho_{1i} &= \angle JOL_i, \quad \text{Left is positive} \\
\rho_{2i} &= \angle P_iOL_i, \quad \text{Up is positive}
\end{align*}
\]

(1)

On the other hand, the dynamics of R-UAV can be concluded as follows.

\[
\begin{align*}
\dot{Z} &= (\cos \phi \cdot \cos \rho_{2i}) \frac{1}{m} u_1 - g \\
\dot{X} &= (\cos \phi \cdot \sin \rho_{2i} \cdot \cos \rho_{1i} + \sin \phi \cdot \sin \rho_{1i}) \frac{1}{m} u_1 \\
\dot{Y} &= (\cos \phi \cdot \sin \rho_{2i} \cdot \sin \rho_{1i} - \sin \phi \cdot \cos \rho_{1i}) \frac{1}{m} u_1
\end{align*}
\]

(2)

where \( \phi \in C \) is the roll angle which has no effect on the path planning problem; \( u_1 \) is the sum of the lift force; \( m \) is the mass of the R-UAV; \( g \) is the acceleration of the gravity, \([X, Y, Z]\) is the location coordinates in Euclidean space.

In this case, point \( P_i = [x_i, y_i, z_i] \) in the \( i \)th surface, and we define it by two angles \( (\rho_{1i}, \rho_{2i}) \). Not only we reduce the dimension of the position matrix, but also the yaw \( \rho_{1i} \) and pitch \( \rho_{2i} \) angles are in accordance with the attitude angles of R-UAV, which will be more convenient to call the compass information in the flight controller than extracting the location information from sensors. The modeling method greatly reduces the computation of iteration.

In order to find the terrains that can reflect the superiority of the algorithm, we use the Google map and World Machine software for help. The terrain example is shown in Fig. 3 which gets from Guilin (25.07, 110.61) in China with 25 km$^2$ in size.

In Fig. 3(a), green and yellow represent vegetation coverage and gullies respectively. In Fig. 3(c), we can acquire a gray level matrix \([A]_{512 \times 512}\) in the floating-point type. The bright and dark parts represent ridge and valley topographies respectively, which is more realistic to show surface fluctuation. The matrixing method also simplifies the problems and improves the adaptability of the algorithm.

![Satellite images and grayscale](image)

Fig. 3 Satellite images and grayscale

2.2 Representation of trajectories

How to design a path as a solution in the search space is the key problem to solve the path planning successfully using PSO. Considering points \( O \) and \( J \) as the origin and destination respectively, they can represent a path combined with intermediate points \([P_1, \ldots, P_i, \ldots, P_{n-1}, J]\).

The path \( \text{path}_i = [O, P_1, \ldots, P_i, \ldots, P_{n-1}, J]_i \in \mathbb{R}^{3(n+1)} \) can be shown in Fig. 4.
A solution is made up of \( n + 1 \) subcomponents, each of which corresponds to a polar position point. \( r \) is the unit length. According to (1), each bit of the subcomponents represents the yaw angle, pitch angle and polar axis length respectively. Fig. 5 is a randomly generated particle corresponding to its position.

\[
\begin{align*}
0 & \quad 0 \quad 0 \quad 0.5236 & -0.1006 & 0.3890 & 0.4230 & 2r & 0.3442 & 0.1273 & 3r & \ldots \\
-0.179 & 0.194 & 4r & 0.0242 & 0.1791 & 5r & -0.3432 & -0.1875 & 6r & -0.0996 & 0.1816 & 7r & \ldots \\
-0.059 & 0.0892 & 8r & 0 & 0 & 9r & r=56.9 & n=9
\end{align*}
\]

\textbf{Fig. 5 A randomly generated particle}

In addition, when judging whether a path is safe without collision or not, we need to transform the angle information into the height information for contrasting with terrain information. Of course, that only exists in the judgment process, and has little influence on the increase of calculation. \( P_i(x(i, y(i, z(i))) \) in Euclidean coordinate system can be calculated as below.

\[
\begin{align*}
x(i) &= i \cdot r \cdot \cos\left(\frac{\pi}{4} + \rho_{i1}\right) / \cos(-\rho_{i1}) \\
y(i) &= i \cdot r \cdot \sin\left(\frac{\pi}{4} + \rho_{i1}\right) / \cos(-\rho_{i1}) \\
z(i) &= i \cdot r \cdot \tan(\rho_{i2})
\end{align*}
\]

where \( r \) is the thickness of each piece; \( |\rho_{i1}| \leq \frac{\pi}{8} \) and \( 0 \leq \rho_{i2} < \frac{\pi}{2} \) based on the terrain and R-UAVs model constraint obviously.

From the above analysis, the author describes the process of terrain modeling and representation of trajectories. The encoding method is simple and visualized, and establishes a reasonable mapping between the solution to the problem and the position of the particle. This method has the following main advantages and disadvantages.

(i) Reduce the computation by reducing the dimension of the particle matrix.

(ii) The yaw and pitch angles are identical with the attitude angle of the R-UAV. It is convenient to call the compass information in the flight controller.

(iii) The proposed coordinate system is a heterogeneity distribution model. On the same surface, the node points in the area of \( \frac{\pi}{8} \leq |\rho_{i1}| \leq \frac{\pi}{4} \) are more than the points in \( |\rho_{i1}| \leq \frac{\pi}{8} \). The paths are relatively dispersive and the diversity of initial population is enhanced.

(iv) From the value of the path angles, we can determine the trend of the path effortlessly.

(v) The author only provides an idea and regards R-UAVs as a mass point without more constraint conditions, but the realistic algorithmic migration may be more complex according to the inertia, wind and GPS signals.

### 2.3 Objective functions

The searchers usually adopt three well-recognized objectives, path length, path smoothness and path safety, for the path planning problem. In this paper, we consider path length as power consumption and the time cost. Meanwhile, we should impose restrictions on large turning angles, which results in exponential consumption of electric power. In terms of security, the threat of radar search can
be grouped with the terrain, and the higher the flight, the easier it is to find by radar. It is worth noting that we do not pay attention to the risk area from radar guided weapons. It will only increase the complexity of the calculation after formulation, which has no virtual help to the improvement of the algorithm. Therefore, we choose the length, the height and the turning angle of a path as the cost function of the MOPSO algorithm, and detailed threat targets and the wind field can be extended if you want to.

A path can be expressed in two coordinate systems according to the angle and the position as

\[
\text{path}_0 = \left((0, 0, 0)^T, (\rho_{11}, \rho_{12}, r)^T, \ldots, (\rho_{n1}, \rho_{n2}, ir)^T, \ldots, (\rho_{n-1,1}, \rho_{n-1,2}, (n-1)r)^T, (0, 0, nr)^T\right)
\]

\[
\text{path}_1 = \left((x_0, y_0, z_0)^T, (x_1, y_1, z_1)^T, \ldots, (x_{n-1}, y_{n-1}, z_{n-1})^T, (x_n, y_n, z_n)^T\right).
\]

The length of the path is expressed as

\[
\cos t_L(\text{Path}_i) = \sum_{j=0}^{n-1} d(P_{j+1}, P_j) \quad (4)
\]

where

\[
d(P_{j+1}, P_j) = \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2 + (z_{j+1} - z_j)^2}
\]

represents the length between points \(P_j\) and \(P_{j+1}\) in Euclidean space.

The height of the path is expressed as

\[
\cos t_H(\text{Path}_i) = \sum_{i=0}^{n} z_i \quad (5)
\]

where \(z_i \geq 0\) is the distance from the horizontal plane to the UAV. This objective we propose cannot be satisfied simultaneously with the length objective in most environments. In practice, the height has a positive effect on the length by highlighting the importance of low altitudes. It prevents the solutions which cross all obstacles along the line \(OJ\) dominating the other solutions.

The turning angle of the path is expressed as

\[
\cos t_T(\text{Path}_i) = \sum_{j=0}^{n} (|\rho_{11}| + \omega_0 |\rho_{12}|). \quad (6)
\]

The turning angle, which represents the rough degree of the path, contains two parts, the yaw \(\rho_{11}\) and the pitch. For R-UAVs, up-and-down motion is easier than side-to-side one, so \(\omega_0 = 0.5\) is practicable.

In this part, we talk about three performance criterions, the height, length and turning angle. Completely conflict goals are not easy to find. It can be seen from the above description that, in most environments, the objectives we propose cannot be satisfied simultaneously and are representarive to some extent. The specific performance will be expressed in the subsequent examples.

3. Improvement of the algorithm

In this part, dominance relation in Section 3.1 is used to update each particle’s \(p\text{best}\) and the external archive. The improvement of the algorithm is carried out from the following two points:

(i) Add in a vibration function to improve the collided solutions.

(ii) Choose an advisable solution as \(g\text{best}\) to lead global particles with the reference point method.

3.1 PSO

PSO has been found to be successful in multiple objectives optimization, proposed by Eberhart and Kennedy [23]. The algorithm, which is based on a metaphor of social interaction, searches a space by adjusting the trajectories of individual vectors. The position of each particle represents a candidate trajectory, and each iterative process is individually based on the previous velocity of the particle, the best position ever occupied by the particle and the best position ever occupied by all the particles in the swarm [24]. Considering the \(i\)th particle’s position and velocity in the swarm, the \(k\)th generation can be updated by the following rules:

\[
\begin{align*}
\vec{v}_i(k+1) &= \omega \times \vec{v}_i(k) + r_1 \times c_1 \times (p\text{best}_i(k) - \vec{A}_i(k)) + r_2 \times c_2 \times (g\text{best}(k) - A_i(k)) \quad (7)
\end{align*}
\]

where \(\omega\) (inertia weight) takes a linear variable value to balance the relationship between exploitation and exploration; \(c_1 = c_2 = 1.496\) 2 control the influences of the \(p\text{best}_i(k)\) and \(g\text{best}(k)\); \(r_{1,2} \in [0,1]\); \(p\text{best}_i(k)\) is the personal best position that the particle \(i\) has had currently; \(g\text{best}(k)\) is a value extracted from the archive.

The greatest challenge in extending the PSO to multi-objective problems is deciphering the notion of guide and selection of \(p\text{best}\) and \(g\text{best}\), as in multi-objective scenarios there is a set of optimal solutions rather than a unique optimum. It means that for each particle, there is more than one candidate for updating its global guidance (the candidates do not dominate each other). Thus we introduce an external archive to record all non-dominated candidates found during the search process and maintain it [25]. The solutions are compared according to the dominance relation in Definition 1 as below.

**Definition 1** Pareto dominance: A vector \(f(A)\) is said to dominate \(g(A)\) (denoted by \(f(A) \leq g(A)\)) if and only
if \( f(A) \) is partially less than \( g(A) \), i.e.,
\[
\forall i \in \{1, \ldots, k\}, \quad f(A) \leq g(A) \cap \\
\exists i \in \{1, \ldots, k\}, \quad f(A) < g(A).
\]
In this problem, \( Path_i \leq Path_j \) if and only if \( cost_i \leq cost_j \), where
\[
\begin{aligned}
\text{cost}_i &= (\text{cost}_L(\text{Path}_1), \text{cost}_H(\text{Path}_1), \text{cost}_T(\text{Path}_0)) \\
\text{cost}_j &= (\text{cost}_L(\text{Path}_1), \text{cost}_H(\text{Path}_1), \text{cost}_T(\text{Path}_0))
\end{aligned}
\]

where \( r_i = |\rho_{1,1} - \rho_{i,1}'| + |\rho_{i+1,1} - \rho_{i+1,1}'|, \quad A_i = (\rho_{1,1}', \rho_{i,2}')^T \) is the feasible fragment closest to \( \rho_i \); \( r_{i,\text{least}} = \min(r_i) \); \( \text{rand}_{1,2} \in [1,2]; k \) is the current iteration; \( \rho_{1,1}(g\text{best}) \) and \( \rho_{i,2}(g\text{best}) \) are the angles of the global optimal solution. \( \rho_{i,1}(p\text{best}) \) and \( \rho_{i,2}(p\text{best}) \) are the angles of the personal optimal solution.

Parameter \( \mu \) represents the infeasibility degree of a fragment. When \( \mu \leq \frac{1}{2} \), the fragment is worth to be corrected based on its own. Thus we increase the height and take random vibrations on the yaw angle by (8) to ensure the diversity. When \( \mu > \frac{1}{2} \), the fragment has big errors which need to be solved combining with the neighboring path and even the \( g\text{best} \) and \( p\text{best} \).

Parameter \( r_{\text{threshold}} \in [0, \frac{3}{4}\pi] \) is the evaluation threshold of similarity degree for two fragments. If \( r_{i,\text{least}} \leq r_{\text{threshold}} \), the two path fragments are similar. It is subjective on account of the time you can sacrifice. Through repeated experimentations, we choose \( r_{\text{threshold}} = 0.1 \max(r) \), and it may reduce iteration time by about 20% in this problem. If \( r_{\text{threshold}} \geq 0.5 \max(r) = \frac{3}{8}\pi \), time increases instead. At the same time, because we split the map into \( n \) bounded pieces artificially, the continuous problems are transformed into discrete problems which have the finite number of solutions. Without loss of generality, the Markov property is preserved in the vibration function. Thus this function does not break the convergence of MOPSO [26].

### 3.2 Vibration function

In the enhanced algorithm, a vibration function is introduced into the collided solutions to improve the algorithm efficiency. The vibration function is defined as below.

**Definition 2** Vibration function: Sample \( p \) points from a fragment \([\rho_{1,1}, \rho_{1,2}]^T, (\rho_{i+1,1}, \rho_{i+1,2})^T\) of \( path_j \) and assume that \( q \) points are collided with the terrain. The vibration factor is defined as \( \mu = \frac{q}{p} \) which represents the infeasibility degree of a fragment. The vibration function \( A_i = \mu(A_i) \) is used to update each bad fragment as follows.

\[
A_i = A_i + \mu \cdot \left[ \frac{0.1 \cdot \text{rand}_1}{k} 0.1 \cdot \text{sgn}(\text{rand}_2 - 0.5) \right]^T, \quad \mu \leq \frac{1}{2}
\]

\[
A_i = \left[ \begin{array}{c}
\rho_{1,1} \\
\rho_{1,2}
\end{array} \right] + \left[ \begin{array}{c}
\frac{0.8r^2}{r_{\text{threshold}}^2} + 0.1 & 0.9 - \frac{0.8r^2}{r_{\text{threshold}}^2}
\end{array} \right]^T, \quad \mu > \frac{1}{2}
\]

\[
A_i = \left[ \begin{array}{c}
\rho_{1,1}(p\text{best}) \\
\rho_{i,2}(p\text{best})
\end{array} \right] + \left[ \begin{array}{c}
\frac{1}{k+1} \\
\frac{k}{k+1}
\end{array} \right]^T, \quad \mu > \frac{1}{2}
\]

### 3.3 Reference point method

The reference point method (RPM) was proposed by Wierzbicki in 1980s [27]. With the help of RPM, we build partially ordered relation rigorously among non-dominance solutions. Not only can we distinguish any two non-dominated solutions according to the decision maker (DM)’s preference, but the dominance relation enhances the selection pressure of the algorithm which accelerates the convergence speed of the algorithm. The definition of the RD is shown as below.

**Definition 3** RD: The weighted Euclidean distance of a certain solution \( p \) to the reference point \( r \) is defined as
\[
RD(C_p, C_r) = \sqrt{\sum_{j=1}^{n} \omega_j (C_{pj} - C_{rj})^2}
\]
where \( C_p = [c_{p1}, \ldots, c_{pj}, \ldots, c_{pn}] \) and \( C_r = [c_{r1}, \ldots, c_{rj}, \ldots, c_{rn}] \in P \) represent the performance criterions of a solution in solution space; \( n \) is the dimension of solution; \( \omega_j \in [0, 1], \sum_{j=1}^{n} \omega_j = 1 \).

The RD between two solutions is defined as
\[
RD(C_p, C_q, C_r) = \frac{RD(C_p, C_r) - RD(C_q, C_r)}{RD_{\text{max}} - RD_{\text{min}}}
\]
where \( C_q \in P, c_{ij} \in [0, 1]; RD_{\text{max}} = \max_{x \in p} RD(x, r); RD_{\text{min}} = \min_{x \in p} RD(x, r). \)
Solution $C_p$ is said to dominate solution $C_r$, either all($c_{pi} \leq c_{ri}$) & $\exists$any($c_{pi} < c_{ri}$) or $RD(C_p, C_q, C_r) < -\delta$, $\delta < [0, 1]$, where $\delta$ is the exploitation pressure.

In order to prevent the loss of diversity of solutions under high selection pressure at the beginning of the iteration, the value of $\delta$ is transformed into a nonlinear form [27].

$$\delta = 1 - (1 - \delta_{DM}) \times (k/k_{max})^p$$  (11)

where $\delta_{DM}$ is the minimum threshold value indicated by the DM; $k$ is the current generation of the evolutionary cycle; $k_{max}$ is the terminal generation; $p \in [1, 10]$ is the nonlinear coefficient.

In the initial stage, $t = 0$, $\delta = 1$, we evaluate the individual solutions based on Pareto dominance relation which retains the exploration ability and ensures the diversity of the search process. With the slow decrease of $\delta$, the particles converge to the desired region faster. The above exponential method ensures the balance between the diversity of the particle group and the preference of the DM, which prevents the algorithm from falling into local optimization.

3.4 Implementation of the proposed algorithm

Based on the aforementioned study, the algorithm is described here.

Step 1 Based on the environment and sensors data, model the landform and workspace by using the method introduced in Section 2.

Step 2 Initialize the population POP and set up the parameters needed in the proposed algorithm.

(i) For $i = 1$ to $S_p$; /* Maximum number of population size $S_p$ */

(ii) Initialize

$POP[i] = [position$, velocity, angle, cost, collide, pbest] on the search space.

(iii) Take each particle as its $p_{best}$ and update the recording of external population (REP).

The selection of $S_p$ should be based on the size of search space; $\omega = 0.73$ is a common choice, and its linear or nonlinear decrease can help to improve the convergence quality of the algorithm. $c_1 = c_2 = 1.4962$ is recommended to set [17].

Step 3 Follow the loop of the iterative process until $k = k_{max}$ or the REP remains unchanged in successive ten generations:

(i) For $k = 1$ to $k_{max}$,

(ii) For $i = 1$ to $S_p$;

(iii) Update $POP[i]$, follow (vii);

(iv) $POP[i] = \mu(POP[i])$, follow (viii);

(v) $i = i + 1$;

(vi) Return to (ii) if $i \leq S_p$;

(vii) Update the global best position $g_{best}(k)$, follow (x);

(viii) Update external archive REP;

(ix) $k = k + 1$;

(x) Return to (i) if $k \leq k_{max}$.

Step 4 Transfer the solutions into optimal paths and guide the R-UAV to the target.

4. Case study

This section is devoted to demonstrating simulation results on the proposed algorithm, and the initial MOPSO and NSGA-II algorithm are used as contrastive experiments to validate the performance of the proposed algorithm [28–30]. The experiments are both coded in MATLAB R2014a software.

4.1 Steep mountain terrain

This terrain example is shown in Section 2.1. The first example considers the characteristic that a mountain blocks the connection between the start and the target, and how to circumvent it is concerned about. The parameters of the simulation are set as follows: the map resolution $res = 512 \times 512$, the population size of the proposed algorithm $S_p = 50$, the maximum number of iterations $k_{max} = 100$, $\omega = 0.73$, $c_1 = c_2 = 1.4962$, the nonlinear coefficient $p = 6$, $\delta_{DM} = 0.5$, $\omega_{4} = (0.4, 0.3, 0.3)$. The reference point is $(110, 6200, 3.08)$.

The population size of NSGA-II $S_n = 50$, the maximum number of iterations $k_{max} = 100$, and the crossover probability $c_p = 0.8$. The dominated parameters are consistent with MOPSO. The cost values of the MOPSO and NSGA-II algorithm are shown in Fig. 6 and Table 1.

![Cost in 3D space](a)
We can see from Table 1 that the proposed algorithm is significantly superior to NSGA-II in the meantime. Also, we note that the several solutions obtained by NSGA-II, shown in boldface, are obviously dominated by the solutions of the proposed algorithm. It indicates that the NSGA-II cannot absolutely guarantee the solutions to be the Pareto front in 100 iterations. The paths in the terrain are shown in Fig. 7.

The experiments are repeated five times and the best results are given in Fig. 7. From Fig. 7(a), we observe two areas, the river valley and peak front, which affect the diversity of the algorithm significantly. The location points obtained by the proposed algorithm are more diverse than the NSGA-II algorithm, and it brings about the better balance of the paths around the mountain.
Fig. 7 Solutions of the proposed algorithm and NSGA-II algorithm

It is worth mentioning that NSGA can get better solutions than before if $k_{\text{max}} > 150$. Because of the choice of reference points, the proposed algorithm has better convergence rate in its neighborhood.

4.2 Loess plateau hill-gully terrain

The terrain example in Fig. 8 is collected from Lanzhou (36.45, 102.71) in China. In order to compare it with the example in Section 4.1, the second example considers a characteristic that the terrains do not have obvious shortcut. No matter how to choose the path, the R-UAV needs to climb the ridges between the peaks many times. The parameters of this simulation are set as follows: the map resolution $res = 512 \times 512$, the population size of MOPSO $S_{p1} = 100$, the proposed algorithm $S_{p2} = 100$ and NSGA-II $S_n = 100$. The maximum number of iterations are $k_{\text{max}} = 500$ equally, $\omega = 0.73$, $c_1 = c_2 = 1.4962$. The reference point is (1 135, 60 000, 4.1). The nonlinear coefficients and crossover probability coincide with the example in Section 4.1.

We can see from Fig. 9 and Table 2 that the solutions in boldface of the proposed algorithm obviously dominate the solutions of MOPSO correspondingly. It is clear from the results that more than half the solutions of MOPSO need to be improved according to the reference point method or traditional dominance relation. On the other hand, when the particle quantity and iteration number are relatively large, the proposed algorithm takes the lead in mean time and the NSGA-II shows better performance in diversity and cost value. The paths in the terrain are shown in Fig. 10. Different from the example in Section 4.1, this example has a sally port in area \( \text{①} \) at the beginning of the simulations which has a great impact on the diversity of solutions. The trajectories from NSGA-II are smooth and distributed evenly. The trajectories from the proposed algorithm, on the other hand, are better in reducing the length and mean time because of the selective pressure from preference. The cost values of the length confirm our conclusion.
XU Zhen et al.: Rotary unmanned aerial vehicles path planning in rough terrain based on multi-objective particle... 139

Fig. 9 Comparison of cost values

Table 2 Some typical costs of the three algorithms

| Algorithm | The proposed algorithm | MOPSO | NSGA-II |
|-----------|------------------------|-------|---------|
|           | Length | Height | Angle | RD   | Length | Height | Angle | Length | Height | Angle | Length | Height | Angle |
| Partial   | 1 139.5 | 67 372 | 4.65 | 0.426 | 0      | 1 407.8 | 68 068 | 5.30 | 1 170.9 | 76 417 | 5.16 | 1 221.7 | 65 440 | 3.976 |
| datum     | 1 186.2 | 69 504 | 4.53 | 0.468 | 3      | 1 577.1 | 52 095 | 4.78 | 944.3 | 81 349 | 5.548 | 4      |       |
|           | 1 137.9 | 72 002 | 4.92 | 0.507 | 5      | 1 577.1 | 52 095 | 4.78 | 944.3 | 81 349 | 5.548 | 4      |       |

Mean time/s 17.092 | 19.350 | 21.335
5. Conclusions

In this paper, we present some advice to handle the path planning problem for known static terrains. The proposed algorithm is easy to use on R-UAV plants and it is fast with desirable performance proved by two representative cases. First, the computation is reduced by designing a new modeling method. Second, the selection pressure is strengthened by introducing DM’s preference which can shorten the convergence time dramatically. To ensure the diversity of the algorithm, a vibration function is introduced which improves the collided solutions rather than forsaking them.

The following work is how to apply the fixed-wing UAV’s constraints into the algorithm, and we are also considering the possibility of extending the algorithm to multiple R-UAVs formation problems.

References

[1] ZHOU M, JIANG L, LIANG S. A UAV patrol system based on bluetooth localization. Proc. of the 2nd Asia-Pacific Conference on Intelligent Robot Systems, 2017: 205 – 209.

[2] TELFORD R, GALLOWAY S. Fault classification and diagnostic system for unmanned aerial vehicle electrical networks based on hidden Markov model. Electrical Systems in Transportation IET, 2015, 5(3): 103 – 111.

[3] MAO Q, ZHANG D. Strategy of control and avoidance of UAVs based on neighborhood tracking and identification. Systems Engineering and Electronics, 2018, 40(9): 2071 – 2078. (in Chinese)

[4] KAKOOEI M, BALEGHI Y. Fusion of satellite, aircraft, and UAV data for automatic disaster damage assessment. International Journal of Remote Sensing, 2017, 38(8): 2511 – 2534.

[5] GU J J, SU T, WANG Q H. Multiple moving targets surveillance based on a cooperative network for multi-UAV. IEEE Communications Magazine, 2018, 56(4): 82 – 89.

[6] PEHLIVANOGLU Y V. A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV. Aerospace Science and Technology, 2012, 16(1): 47 – 55.

[7] GOERZEN C, KONG Z, METTLER B. A survey of motion planning algorithms from the perspective of autonomous UAV
guidance. Journal of Intelligent & Robotic Systems, 2010, 57(1/4): 65 – 100.
[8] SUN T Y, HUO C L, TSAI S J. Optimal UAV flight path planning using skeletonization and particle swarm optimizer. Proc. of the IEEE Congress on Evolutionary Computation, 2008: 1183 – 1188.
[9] CHOJU R C H, KAUFMAN A E, LIANG Z R. An interactive fly-path planning using potential fields and cell decomposition for virtual endoscopy. IEEE Trans. on Nuclear Science, 2002, 46(4): 1045 – 1049.
[10] LUO D L, WU S X. Ant colony optimization with potential field heuristic for robot path planning. Systems Engineering and Electronics, 2010, 32(6): 1277 – 1280. (in Chinese)
[11] MASEHIAN E, SEDIGHIZADEH D. Classic and heuristic approaches in robot motion planning—a chronological review. Proc. of World Academy of Science Engineering & Technology, 2007: 101 – 106.
[12] WANG X F, CHAI J, ZHOU J. Cooperative path planning of multi-UAV based on multi-objective optimization algorithm. Systems Engineering and Electronics, 2017, 39(4): 782 – 787. (in Chinese)
[13] DUAN H B, PEI L, SHI Y H. Interactive learning environment for bio-inspired optimization algorithms for UAV path planning. IEEE Trans. on Education, 2015, 58(4): 276 – 281.
[14] WANG Q, ZHANG A, QI L. Three-dimensional path planning for UAV based on improved PSO algorithm. Proc. of the China Control and Decision Conference, 2014: 3981 – 3985.
[15] LI M,SONG Q, ZHAO Q I. UAV path re-planning based on improved bidirectional RRT algorithm in dynamic environment. Proc. of the 3rd International Conference on Control, Automation and Robotics, 2017: 658 – 661.
[16] Wang B F, LI S, GUO J. Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm. Neurocomputing, 2017, 282: 42 – 51.
[17] COELLO C A C, LECHUGA M S. MOPSO: a proposal for multiple objective particle swarm optimization. Proc. of the IEEE Congress on Evolutionary Computation, 2002, 2: 1051 – 1056.
[18] HUANG C, FEI J Y. UAV path planning based on particle swarm optimization with global best path competition. International Journal of Pattern Recognition and Artificial Intelligence, 2018, 32(06): 132 – 140.
[19] GENG N, MENG Q. How good are distributed allocation algorithms for solving urban search and rescue problems? A comparative study with centralized algorithms. IEEE Trans. on Automation Science and Engineering, 2018, 16(1): 478 – 485.
[20] GENG N, GONG D W. Robot path planning in an environment with many terrains based on interval multi-objective PSO. Proc. of the IEEE Congress on Evolutionary Computation, 2013: 813 – 820.
[21] GENG N, GONG D W, ZHANG Y. PSO-based robot path planning for multi survivor rescue in limited survival time. Mathematical Problems in Engineering, 2014: 187370.
[22] GENG N, SUN X J, GONG D W. Solving robot path planning in an environment with terrains based on interval multi-objective PSO. International Journal of Robotics & Automation, 2016, 31(2): 100 – 110.

Biographies

XU ZHEN was born in 1994. He is a Ph.D. candidate in School of Automation, Nanjing University of Science and Technology. His research interests include the control and path planning of multi-agent system.
E-mail: xz940706@163.com

ZHANG Enze was born in 1989. She received her Ph.D. degree in Nanjing University of Science and Technology in 2017. Now, she is a lecturer in Yangzhou University. Her research interests include multi-objective optimization and particle swarm algorithm.
E-mail: yzzeez8986@yzu.edu.cn

CHEN Qingwei was born in 1963. He received his Ph.D. degree in control theory and control engineering from Nanjing University of Science and Technology in 2005. He is a professor in the School of Automation, Nanjing University of Science and Technology. His research interests include intelligent control and servo system.
E-mail: cqw6061@163.com