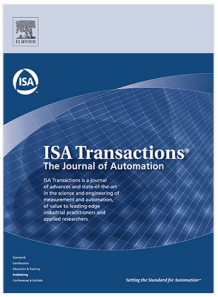
# Journal Pre-proof




Efficient path planning for UAV formation via comprehensively improved particle swarm optimization

Shikai Shao, Yu Peng, Chenglong He, Yun Du

PII: S0019-0578(19)30353-2
DOI: https://doi.org/10.1016/j.isatra.2019.08.018
Reference: ISATRA 3301

To appear in: *ISA Transactions*

Received date : 25 March 2019
Revised date : 5 August 2019
Accepted date : 6 August 2019

Please cite this article as: S. Shao, Y. Peng, C. He et al., Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. *ISA Transactions* (2019), doi: https://doi.org/10.1016/j.isatra.2019.08.018.

# Efficient path planning for UAV formation via comprehensively improved particle swarm optimization

Shikai Shao[a,*], Yu Peng[a], Chenglong He[b], Yun Du[a]

[a] *School of Electrical Engineering, Hebei University of Science and Technology, Shijiazhuang, 050018, China*

[b] *State Key Laboratory of Satellite Navigation System and Equipment Technology, Shijiazhuang, 050081,China*

[*] Corresponding author.

Email address: kdssk@126.com

## Highlights

1. A novel and comprehensively improved PSO is proposed and analyzed.

2. Path planning for UAV formation with 10 vehicles are executed and the formation performance is evaluated by Monte-Carlo simulations.

3. Faster convergence speed and better solution optimality are achieved without additional procedure running time.

# Efficient path planning for UAV formation via comprehensively improved particle swarm optimization

**Abstract**: Automatic generation of optimized flyable path is a key technology and challenge for autonomous unmanned aerial vehicle (UAV) formation system. Aiming to improve the rapidity and optimality of automatic path planner, this paper presents a three dimensional path planning algorithm for UAV formation based on comprehensively improved particle swarm optimization (PSO). In the proposed method, a chaos-based Logistic map is firstly adopted to improve the particle initial distribution. Then, the common used constant acceleration coefficients and maximum velocity are designed to adaptive linear-varying ones, which adjusts to the optimization process and meanwhile improves solution optimality. Besides, a mutation strategy that undesired particles are replaced by those desired ones is also proposed and the algorithm convergence speed is accelerated. Theoretically, the comprehensively improved PSO not only speeds up the convergence but also improves the solution optimality. Finally, Monte-Carlo simulation for UAV formation under terrain and threat constraints are carried out and the results illustrate the rapidity and optimality of the proposed method.

**Key words:** UAV formation; Path planning; Particle swarm optimization (PSO); Mutation strategy

## 1. Introduction

Unmanned aerial vehicle (UAV) is an aircraft without human pilot onboard. With the development of technology, UAV has been widely applied to both military and civilian tasks, such as intelligence, surveillance, reconnaissance, rescue and commercial performance [1-3]. Compared to single vehicle, UAV formation has unique advantages of low cost, better robust and can carry out complicated tasks beyond the ability of single UAV.

Path planning is one of the most important techniques for autonomous UAV formation flying and is also a main challenge for autonomous UAV application in

engineering [4,5]. The path planning problem for a UAV can be formulated as an optimization problem which aims to find a feasible path from the starting position to the destination position under different optimization criteria and mission constraints, which may include minimal flight length, minimal flight time, UAV state constraints, environmental constraints, etc. [6,7]. Considering the fundamental and crucial role in improving the autonomy and intelligent level of unmanned system, the research for UAV path planning has drawn increasing attention.

In the past few years, a variety of methods have been proposed for UAV or autonomous robot path planning. Graph-based algorithms, such as Voronoi diagram algorithm [8], A* algorithm [9], probabilistic roadmaps algorithm [10,11], rapidly-exploring random trees based algorithm [12,13], are one kind of simple path planning methods. However, UAV kinematic and dynamic constraints are seldom considered by these algorithms, thus they usually can not be utilized for practical situations. Besides, the algorithms of this kind are based on the cost map, which should be produced and stored in advance, resulting the cost map production being time-consuming. Potential field-based method is another kind of effective path planning methods. Artificial potential field algorithm [14,15] and interfered fluid dynamical system algorithm [16] are two typical examples. In order to generate the flyable path, such algorithms need to globally establish the interaction between the attractive field and the repulsive field. As a result, they are easy to be trapped into local minimum and sometimes no feasible path could be guaranteed when the target and obstacles are near. With the development of swarm intelligent technology, the population based evolutionary algorithms have made much progress [17,18] recently, and they maintain strong ability to search the optimal solution in a more efficient and flexible fashion. More and more researchers have paid attention to UAV path planning by using this kind of method. The commonly used algorithms mainly include artificial bee colony algorithm (ABC) [19], the ant colony algorithm (ACO) [20], the genetic algorithm (GA) [21] and the particle swarm algorithm (PSO) [22,23].

Among these algorithms, PSO maintains the advantages of simple implementation principle and being able to access global optimum for all particles, thus it is often utilized for UAV path planning and other optimization tasks. In Ref.

[24], PSO was adopted to solve optimal scheduling for a swarm robotic systems. In Ref. [25], the dynamic hybrid PSO combined with the trim-maneuver library technology was proposed for UAV motion planning. In Ref. [26], the authors considered the drawback of basic PSO that some high-quality waypoints in previous candidate paths might not be exploited for further evolution, and proposed a new idea of separately evaluating and evolving waypoints for two dimensional UAV path planning. Similar idea was extended by Ref.[27], where the authors introduced an optimal path competition strategy between the single waypoint selection and the integrated waypoints selection, and completed a three dimensional UAV path planning demonstration. In order to overcome the defects of local minimum and premature of PSO, the authors in Ref.[28] proposed an improved PSO combining with an adaptive decision operator for three dimensional UAV path planning, and comparison results illustrated the advantages of the proposed strategy. Apart from the utilization in UAV path planning, some researchers have also focused on PSO theoretical analysis and improvements. In Ref.[29], the authors introduced chaotic maps and Gaussian mutation mechanism into a standard PSO to solve the local minimum and premature problems. Furthermore, the authors comprehensively analyzed the factors affecting the performance of PSO, and pointed out that swarm initialization, parameter selection, topology structure and combination with hybrid methods were four main aspects in improving the efficiency of PSO. The authors in Ref. [30] designed a new component in the velocity update rule and took the similar view as Ref. [29] to improve PSO performance.

After the generation of feasible flying path, two critical indicators, namely the convergence speed and the solution optimality, are often compared to evaluate the performance of the planned paths. For practical applications, the paths with faster convergence speed and better solution optimality are always preferred. Although some PSO related path planning results have been obtained, the convergence speed and solution optimality is still not satisfactory for practical flying. Based on existing results, how to further speed up convergence and improve solution optimality is a main motivation of this research. Besides, considering that most existing results are focused on single UAV path planning, which may be good enough for one application,

however, may not be good enough for another due to different environmental constraints. A formation system usually contains a variety of members that each one may face different environment, how to improve the existing PSO and evaluate the performance of the whole formation system is also another important issue for applications.

In this paper, the particle initial distribution, the parameters and update topology are comprehensively considered and a more accurate and faster PSO path planning algorithm is proposed. According to PSO evolution principle, particles with more uniform initial distribution are easier to obtain global optimal solution. Thus, in this paper, a chaos-based Logistic map, which maintains better uniform distribution property than random function, is introduced for particle initialization. During PSO evolution process, acceleration coefficients and maximum velocity are key parameters in influencing the convergence speed and solution optimality. In this research, an adaptive linear-varying strategy is proposed for both acceleration coefficients and maximum velocity, which ensures the designed PSO algorithm achieve global optimal solution with much less iterations. Besides, the position update topology is also an important factor in influencing PSO performance. Herein, according to the fitness value of the whole particles, a mutation update strategy is proposed. The undesired particles are replaced by some mutants of the desired ones, thus the influence of desired particles is strengthened and that of undesired particles is weakened. Finally, the comprehensively improved PSO is utilized for UAV formation with 10 vehicles under different environmental constraints. Different from single UAV path planning, the formation average convergence speed and average optimal solution are evaluated and compared via the Monte-Carlo simulation, which is more fair and convincing than the comparison of single UAV and has been seldom investigated by researchers.

The main contributions of the paper are as follows: Firstly, three improvements for PSO, namely the chaos-based initialization, the adaptive varying parameter and the mutation update strategy, are introduced and analyzed. Secondly, a more accurate and faster PSO algorithm is proposed. Comparing with basic PSO and modified genetic algorithm, the convergence speed and solution optimality are effectively improved. Thirdly, the proposed PSO is successfully adopted for UAV formation path

planning under terrain, threat and collision avoidance constraints. The formation average performance is comprehensively evaluated and the results is more convincing than single UAV.

The rest of the paper is organized as follows: Section 2 formulates mathematical model for UAV path planning, including environmental representation, objective function and UAV constraints. In Section 3, the three improvements and analysis for the proposed PSO are introduced and the procedure for UAV path planning is given. Experimental simulations and comparisons with different scenarios are carried out in Section 4. Finally, detailed conclusions are drawn in Section 5.

**Nomenclature**

| | |
|---|---|
| $x_{\min}, x_{\max}, y_{\min}, y_{\max}, z_{\min}, z_{\max}$ | Flying space……km |
| $x_k^m, y_k^m, z_k^m$ | Coordinate of the $k$ th mountain……km |
| $\left(x_k^{m0}, y_k^{m0}\right), h_k, x_k^t, y_k^t$ | The $k$ th mountain parameters |
| $\left(x_k^r, y_k^r, z_k^r\right), R_k$ | Radar center and radius…… ..km |
| $f_L$ | Flying path length cost |
| $f_T$ | Terrain cost |
| $f_R$ | Radar detection cost |
| $f_C$ | Collision cost among UAVs |
| $N_w$ | Waypoint number |
| $\delta$ | Intensity of the radar |
| $\overline{d}$ | Safe distance between UAVs…… ..km |
| $\mathbf{p}_i$ | The particle position |
| $\mathbf{v}_i$ | The particle velocity |
| $w$ | Inertia weight |
| $c_1, c_2$ | Acceleration coefficients |
| $T$ | Total iterations |
| $r_1, r_2$ | Random values in $(0,1)$ |
| $V_{\max}$ | Particle maximum velocity |
| $w_{\max}, w_{\min}$ | Inertia weight parameters |
| $\mu$ | Bifurcation coefficient |
| $c_{\max}, c_{\min}$ | Acceleration coefficient parameters |

| | |
|---|---|
| $V_1, V_2$ | Maximum velocity parameters |
| $a$ | Position offset parameter |
| $(x_s, y_s, z_s)$ | UAV starting position |
| $(x_d, y_d, z_d)$ | UAV destination position |
| $S$ | Swarm size |
| $D$ | Particle dimension |

## 2. Problem description

When planning paths for each UAV in the formation, some important factors need to be taken into consideration, such as the environment of mission area, UAV safety and the cost of each path. In general, the mission area environment mainly include the flying range, the terrain condition, the threat level and so on. The objective function should consider all the necessary environmental elements and also should be able to evaluate how they influence the performances. Then, the optimization problem for UAV path planning is formulated and solved by the proposed PSO. In the following, the environmental constraints and objective function are respectively introduced.

### 2.1 Flying space and path representation

The goal for UAV path planning is to find a feasible and optimal path to the target area under complex environment and state constraints. In the paper, we denote $(x, y, z)$ as the coordinate of a path waypoint in three dimensional environment. The flying space for UAV can be expressed as [27]

$$\left\{(x, y, z) \mid x_{\min} \le x \le x_{\max}, y_{\min} \le y \le y_{\max}, z_{\min} \le z \le z_{\max}\right\} \tag{1}$$

where $x_{\min}, x_{\max}, y_{\min}, y_{\max}, z_{\min}, z_{\max}$ denote the flying bounds respectively.

### 2.2 Terrain and threat model

In common applications, accurate and updatable terrain maps exist and terrain constraints can be known in advance. In the paper, we suppose mountains as the main terrain feature and each UAV can not fly through mountains. The simplified model for a mountain is descripted as [27]

$$z_k^m = h_k * \exp\left(\frac{\left(x_k^m - x_k^{m0}\right)^2}{x_k^t} + \frac{\left(y_k^m - y_k^{m0}\right)^2}{y_k^t}\right) \tag{2}$$

where $x_k^m, y_k^m, z_k^m$ represent the three dimensional coordinate of the $k$ th

mountain, $\left(x_k^{m0}, y_k^{m0}\right)$ is the horizontal center of the mountain, and $h_k, x_k^t, y_k^t$ are the designed parameters, which can affect the height and slope of a mountain.

For a flying UAV under penetration condition, it may face high risk threat. In this paper, we consider the radar detection as the risk threat. Because the defense scope of a radar is omnidirectional, the mathematical model of the threat is treated as a geometric sphere. The simplified mathematical model for a radar threat is written as [27]

$$T_k = \left(x_k^r, y_k^r, z_k^r, R_k\right) \tag{3}$$

where $\left(x_k^r, y_k^r, z_k^r\right)$ is the center of the $k$ th radar threat and $R_k$ is the radius of the threat. The radar threat is formed as half sphere.

**2.3 Objective function design**

The search for feasible UAV formation flying paths is a complicated optimization problem with multi-objective. Taking the path length, environmental constraints and collision avoidance into consideration, the evaluation function for UAV formation path planning can be written as follows

$$f = f_L + f_T + f_R + f_C \tag{4}$$

where $f_L$ is the minimal flying path length cost, $f_T$ is the terrain cost, namely the safe flying cost around the mountain, $f_R$ is the radar detection cost and $f_c$ is the collision cost among flying UAVs.

(1) Minimal flying path length cost

For flying missions, shorter path length is always preferred than longer path length, because shorter paths usually result in less fuel consumption and can reduce the probability of encountering an unexpected threat. To characterize this factor, path length ratio (PLR) is used for evaluating the path length, which is given as follows [26]

$$f_L = \frac{\sum_{i=2}^{N_w} \sqrt{\left(x_i - x_{i-1}\right)^2 + \left(y_i - y_{i-1}\right)^2 + \left(z_i - z_{i-1}\right)^2}}{\sqrt{\left(x_{N_w} - x_1\right)^2 + \left(y_{N_w} - y_1\right)^2 + \left(z_{N_w} - z_1\right)^2}} \tag{5}$$

where $N_w$ is the number of total path waypoints, including the staring waypoint $\left(x_1, y_1, z_1\right)$, the destination waypoint $\left(x_{N_w}, y_{N_w}, z_{N_w}\right)$, and the designed waypoints $\left(x_i, y_i, z_i\right)$ with $i = 2, \ldots, N_{w-1}$. The value of PLR always satisfies

$f_L \geq 1$, and the smaller value means the shorter path.

(2) Mountain terrain cost

For UAV safe flying, the planned path should not go through into the terrain and must avoid all the mountains in 3D environment. The solutions should be penalized if at least one point of the path is inside the mountain. The cost function for mountain constraint is depicted as [28]

$$f_T = \sum_{i=2}^{N_w} \sum_{j=1}^{n} A_{i,j} \quad \text{with} \quad A_{i,j} = \begin{cases} 1, & if \quad z_{i,j} \leq z_{k,i,j}^m \\ 0, & otherwise \end{cases} \tag{6}$$

where $A_{i,j}$ is a binary value. For each segment between the $(i-1)$ th and the $i$ th waypoint, it is divided into $n$ piecewise parts, denoted as $(x_{i,j}, y_{i,j}, z_{i,j})$. $z_{k,i,j}^m$ is the return height for horizontal position $(x_{i,j}, y_{i,j})$ of the $k$ th mountain. The piecewise number $n$ reflects the tradeoff between computation cost and path accuracy, which should be set according to different situations.

(3) Radar threat cost

When a UAV is flying into the detection range of a radar, it may encounter the risk of being discovered or being attacked. The longer distance that the waypoints and the radar center maintains, the lower probability that the UAV is discovered. The cost function of radar detection threat is [26]

$$f_R = \sum_{i=2}^{N_w} \sum_{j=1}^{n} B_{i,j} \quad \text{with} \quad B_{i,j} = \begin{cases} \left(\delta / D_{i,j}\right)^4, & if \quad D_{i,j} \leq R_k \\ 0, & otherwise \end{cases} \tag{7}$$

where

$$D_{i,j} = \sqrt{\left(x_{i,j} - x_k^r\right)^2 + \left(y_{i,j} - y_k^r\right)^2 + \left(z_{i,j} - z_k^r\right)^2} \tag{8}$$

denotes the distance between path point $(x_{i,j}, y_{i,j}, z_{i,j})$ and the radar center $(x_k^r, y_k^r, z_k^r)$, $\delta$ denotes the scale of the intensity of the radar.

(4) UAV collision cost

For multi-UAV formation flying, it is essential to take collision avoidance into consideration during the path planning. Each UAV should keep a relatively large distance with others. The UAV collision cost function is designed as follows

$$f_C = \sum_{i=2}^{N_w} \sum_{j=1}^{n} C_{i,j} \quad \text{with} \quad C_{i,j} = \begin{cases} 1, & if \quad d_{i,j} \leq \bar{d} \\ 0, & otherwise \end{cases} \tag{9}$$

where $d_{i,j}$ is the shortest distance between $\left(x_{i,j}, y_{i,j}, z_{i,j}\right)$ and other path points with different UAV number. $\bar{d}$ is the predesigned safe distance between each UAV.

## 3. Improved PSO path planning

### 3.1 Standard PSO path planning

Particle swarm optimization is a population-based stochastic optimization algorithm. In PSO framework, the algorithm starts from a randomly initialization with candidate solutions and could find a global optimal solution via iteration based on position and velocity updating [26-28]. For each particle, the velocity represents the searching direction and is dynamically updated by its previous value, the personal best position and the global best positon. And the new position is mainly dependent on its previous value and the current velocity. If the newly obtained position of a particle is better than its previous value according to the fitness function, then the new one will be restored as the personal best position. If the new position is better than all other particles', then it will be restored as the global best position.

For a three dimensional path planning problem, suppose the waypoint number of each particle is $D$, then the position and velocity vector for the $i$ th particle can be respectively written as

$$\mathbf{p}_i = \left(\mathbf{p}_{i,1},...,\mathbf{p}_{i,D}\right)^T = \left(\left(p_{i,1}^x, p_{i,1}^y, p_{i,1}^z\right),...,\left(p_{i,D}^x, p_{i,D}^y, p_{i,D}^z\right)\right)^T \tag{10}$$

$$\mathbf{v}_i = \left(\mathbf{v}_{i,1},...,\mathbf{v}_{i,D}\right)^T = \left(\left(v_{i,1}^x, v_{i,1}^y, v_{i,1}^z\right),...,\left(v_{i,D}^x, v_{i,D}^y, v_{i,D}^z\right)\right)^T \tag{11}$$

where $\left(p_{i,j}^x, p_{i,j}^y, p_{i,j}^z\right), \left(v_{i,j}^x, v_{i,j}^y, v_{i,j}^z\right), j \in 1,...,D$ denote the $j$ th waypoint's position and velocity of the $i$ th particle in three dimensional space. If the total particle number is $S$, then the whole swarm can be expressed as

$$\left(\left(\mathbf{p}_1, \mathbf{v}_1\right), \left(\mathbf{p}_2, \mathbf{v}_2\right), ..., \left(\mathbf{p}_S, \mathbf{v}_S\right)\right) \tag{12}$$

For PSO with $S$ particles, there are the same number personal best positions and one global best position, which are written as

$$\mathbf{P}_{i,best} = \left(\mathbf{p}_{i,1,best},...,\mathbf{p}_{i,D,best}\right)^T = \left(\left(p_{i,1,best}^x, p_{i,1,best}^y p_{i,1,best}^z\right),...,\left(p_{i,D,best}^x, p_{i,D,best}^y p_{i,D,best}^z\right)\right)^T \tag{13}$$

$$\mathbf{G}_{best} = \left(\mathbf{g}_{1,best},...,\mathbf{g}_{D,best}\right)^T = \left(\left(p_{1,best}^x, p_{1,best}^y p_{1,best}^z\right),...,\left(p_{D,best}^x, p_{D,best}^y p_{D,best}^z\right)\right)^T \tag{14}$$

where $i \in 1,...,S$ is the label of particle number. The velocity and position of each particle are updated according to the following formula [27-30]

$$\begin{cases} \mathbf{v}_{i,j}(t+1) = w \cdot \mathbf{v}_{i,j}(t) + c_1 \cdot r_1 \cdot \left(\mathbf{p}_{i,j,best}(t) - \mathbf{p}_{i,j}(t)\right) + c_2 \cdot r_2 \cdot \left(\mathbf{g}_{j,best}(t) - \mathbf{p}_{i,j}(t)\right) \\ \mathbf{p}_{i,j}(t+1) = \mathbf{p}_{i,j}(t) + \mathbf{v}_{i,j}(t+1) \end{cases} \tag{15}$$

where $i \in 1,...,S$, $j \in 1,...,D$ are the label of particle number and waypoint number. $t \in 1,...,T$ is the current iterations with $T$ denoting the total iterations. $w$ is the inertia weight used for balancing the global and local searching, $c_1, c_2$ are acceleration coefficients reflecting the ability of learning from the particle itself and the whole warm. $r_1, r_2$ are two random numbers distributed in $(0,1)$, indicating the randomness for PSO in searching global optimal solution. The updating velocity is usually restricted by $[-V_{\max}, V_{\max}]$. In order to accelerate the swarm convergence, some researchers have adopted linear-varying inertia weight, which is as [29]

$$w = w_{\max} - \frac{(w_{\max} - w_{\min})t}{T} \tag{16}$$

The standard PSO possesses the convenience of few parameters and easy implementation, however, some drawbacks accompany standard PSO, such as it may fall into local minimum and premature. Besides, the total iterations for global optimal solution is usually quite large and the whole evaluation is time consuming.

### 3.2 Comprehensively improved PSO

In order to search the global optimal solution in a faster and more accurate way, we propose several improvements for standard PSO, including improving the particle initial distribution, adaptively adjusting parameters according to evaluation and changing the structure topology for updating best positions.

#### 3.2.1 Chaos-based particle initialization

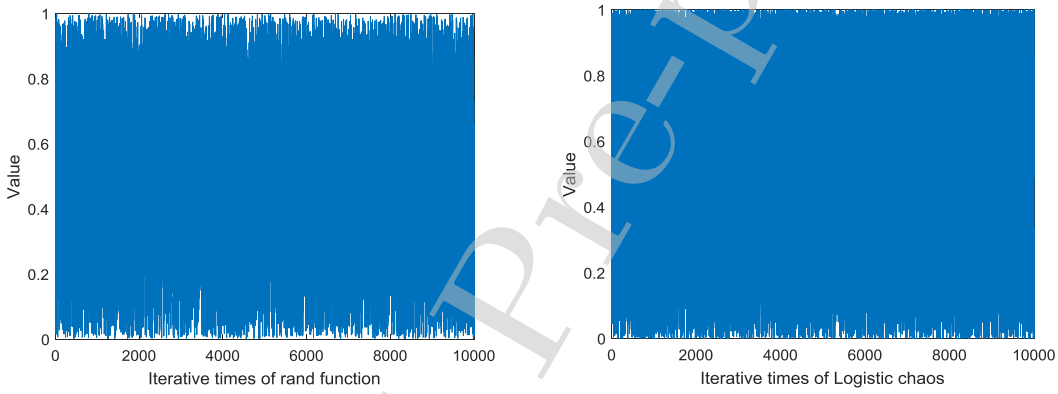
As is well known, PSO is initialized with a population of random distributed solutions. The distribution quality of particle initialization plays a critical role in the searching for optimal solution. The more uniform the initial distribution is, the richer diversity the swarm maintains, and thus the better chance and faster convergence of acquiring an optimal solution are obtained. Inspired by [29], some researchers have theoretically proposed chaos-based swarm initialization for standard PSO. Compared with random initialization, the chaotic initialization is a powerful strategy to diversify the swarm and improve the performance of PSO. In this paper, in order to enrich the diversity of particle initialization in UAV path planning, the Logistic map, which is

one of the simplest and the most widely used maps, is introduced as follows [29]

$$x_{n+1} = \mu x_n \left(1 - x_n\right) \tag{17}$$

where $\mu$ is the bifurcation coefficient, $x_n$ is the $n$ th chaotic variable. When $\mu \in \left[3.57, 4\right]$ the system evolves into the chaotic state. When $\mu = 4, x_0 \in \left(0,1\right)$ the system will produce chaotic signal $x_n \in \left(0,1\right)$ uniformly, which will be utilized for particle initialization.

In order to better illustrate the advantage of Logistic map initialization compared with random initialization, the simulated figures for 10000-time iterations by Matlab are shown in Fig.1. It is obvious that the line distribution of Logistic chaos map is more uniform than that of rand function, which would enrich the diversity of the candidate flying paths for UAV.




(a) Rand function map (b) Logistic chaos map

Figure 1 Comparison of rand function and Logistic map

### 3.2.2 Adaptive parameter adjustment

Apart from inertia weight, acceleration coefficients $c_1, c_2$ are another two critical parameters. They indicate the weight of stochastic acceleration terms that pull each particle to local and global best positions and play important role in adjusting the convergence speed and searching direction. For most PSO related research, the coefficients are set as constant values, however, from the view of swarm evaluation, the PSO can maintain better performance if the coefficients are adaptively adjusted with the evaluation process.

Because of the randomness of the searching process, it is difficult to carry out precise quantitative analysis between coefficient values and iterative times. However, we can always analyze the mapping relationship between them in a qualitative way. For PSO evolution, searching and convergence are the two main objectives during the

process. Searching means each particle should enrich the diversity of the swarm and convergence means the swarm should finally achieve a same value, namely, the global optimal solution. Theoretically, according to the PSO searching and convergence process, it is reasonable to conclude that searching process should predominate the evolution at the first half stage while convergence process should predominate at the last half stage. Thus, the acceleration coefficients $c_1, c_2$ are adaptively designed as

$$c_1 = c_{\max} - \frac{\left(c_{\max} - c_{\min}\right)t}{T} \tag{18}$$

$$c_2 = c_{\min} + \frac{\left(c_{\max} - c_{\min}\right)t}{T} \tag{19}$$

where $T$ is the total iterative times and $t$ is the current iterative times, $c_{\max}, c_{\min}$ are constant values with $c_{\max} > c_{\min} > 0$. In the simulation, $c_1$ and $c_2$ for conventional PSO are set as a same constant value as in the researches [27-29], while $c_{\max}, c_{\min}$ for the proposed PSO are set according to the principle $\left(c_{\max} + c_{\min}\right)/2 = c_1^c = c_2^c$, where $c_1^c = c_2^c$ are the conventional PSO values. This principle guarantees that the average $c_1$ and $c_2$ for the proposed PSO are the same as those for conventional PSO, and makes the simulation comparisons much more fair and well-founded.

It is obvious that $c_1$ and $c_2$ are respectively linearly decreasing and linearly increasing, and the sum satisfies $c_1 + c_2 = c_{\max} + c_{\min}$, meaning that the ability for particle searching and convergence is constant. For Eqs.(18)~(19), it is satisfied that $c_1 > c_2$ at the first half stage, thus the searching diversity is strengthened, which would be benefit for fast obtaining global optimal solution and avoiding local minimum. $c_1 < c_2$ is satisfied at the last half stage and has the benefit of fast convergence to the same global optimal solution. In a word, the adaptive coefficients would improve the optimality and rapidity of PSO.

### 3.2.3 Maximum velocity design

Maximum velocity $V_{\max}$ is also an important factor for PSO performance, which has been seldom paid attention by researchers. Generally, $V_{\max}$ is taken as constant value in PSO and is not sufficient to precisely reflect the searching diversity and accuracy. For PSO evolution, it is expected that the searching range should be

large enough to enrich the solution diversity at the beginning and the range should be relatively small to improve the accuracy of optimal solution at the end. A constant maximum can hardly meet the requirements simultaneously. In this section, a linear-varying strategy for $V_{\max}$ is designed as follows

$$V_{\max} = V_1 - \frac{(V_1 - V_2)t}{T} \tag{20}$$

where $T$ and $t$ are the total iterative times and the current iterative times, $V_1 > V_2 > 0$ are constant values. In the simulation, $V_1$ and $V_2$ for the proposed PSO are selected satisfying the principle $(V_1 + V_2)/2 = 2V_{\max}^c$, where $V_{\max}^c$ is the corresponding conventional PSO value, to make the comparisons fair and well-founded.

For the velocity varying strategy, $V_{\max} = V_1$ when $t = 0$, $V_{\max} = V_2$ when $t = T$, and $V_{\max}$ linearly decreases from $V_1$ to $V_2$ as iterative times $t$ evolves. The designed strategy can take both searching range and searching accuracy into consideration. In the beginning stage, the velocity with bigger value can guarantee a wider searching range, thus enriches the solution diversity and improves the probability of meeting global optimal solution. At the final stage, the velocity with smaller value can help to refine the optimal solution by smaller searching range and remove unnecessary searching of non-optimal solutions. Theoretically, the designed adaptive strategy guarantees better performance than constant $V_{\max}$.

### 3.2.4 Position updating strategy

Position updating topology is an important aspect in the designing of PSO algorithm, and also a main issue for researchers to improve PSO performance. Some researchers have introduced adaptive sensitivity decision operator to update the best position [28], but other computation models would be brought into the algorithm and computation complexity would be increased. In order to improve the convergence speed with a simple way, a new position updating strategy is proposed in the following.

Firstly, carry out the standard PSO for one time and calculate the fitness values of all particles. Rearrange the particles whose fitness value is from small to large. Then according to the fitness value, the position updating strategy is proposed as follows

$$\begin{cases} \mathbf{p}_{large} = \mathbf{p}_{small} + a \cdot r_3 \\ \mathbf{v}_{large} = \mathbf{v}_{small} \end{cases} \tag{21}$$

where $\mathbf{p}_{large}$ and $\mathbf{p}_{small}$ respectively represent the half of swarm particles with large fitness values and small fitness values, $r_3$ is a random number in $[0,1]$, $a$ is a constant value reflecting the position offset extent of the newly obtained mutant particles to the particles with small fitness value. The value for $a$ should be dependent on the searching space and the convergence of all particles. Ideally, $a$ should be relatively large at the beginning and becomes increasing smaller as all the particles evolve. To avoid additional running procedure and for brevity, the value is set constant in the simulation.

The above equation shows that the particles with large fitness values will be replaced by those with small fitness values with an offset value, and the particle velocities with large fitness values would remain the same. For PSO path planning, the particles with large fitness values means that they are far from the shortest path or some of the waypoints are within the threat or collision areas, thus using them for subsequent evolution is not meaningful and the evolution for optimal solution would be time consuming, so we propose the position replacement strategy to accelerate the convergence. Meanwhile, in order to keep the diversity of the swarm, an offset value $a$ is added in the replacement and the velocity is not affected by the position replacement. This proposed strategy would theoretically improve the convergence speed of swarm evolution and remain swarm diversity.

**Remark 1**: The conventional PSO is formulated by Eqs.(15)~(16) while the proposed comprehensively improved PSO is formulated by Eqs.(15)~(21). It is obvious that the main structure of the two PSOs are almost the same, but the proposed PSO refines the parameter settings by Eqs.(17)~(21). By qualitative analysis, the proposed PSO can improve both the convergence speed and the solution optimality. Meanwhile, little extra running time is needed for the proposed PSO due to the same main structure.

### 3.3 Path planning algorithm for UAV formation

In this section, we give the detailed steps for UAV formation path planning by using the comprehensively improved PSO.

***Step 1***: Construct the 3D mission area, terrain and threat model according to Eqs.(1)~(3). Set the starting and destination points for UAV formation.

***Setp 2:*** PSO parameters initialization for each UAV, including swarm scale $S$, particle dimension $D$, inertia weight $W$, acceleration coefficients $c_1, c_2$, maximum velocity $V_{\max}$, maximum iterative time $T$ and swarm initialization by Eq.(17).

***Setp 3:*** Construct the fitness function for each UAV by Eqs.(4)~(9). Carry out the PSO algorithm for one time.

***Step 4:*** Calculate the fitness function, figure out the particle best position and swarm best position.

***Step 5:*** Update parameters by designed strategies, including inertia weight $W$ by Eq.(16), swarm initialization by Eq.(17), acceleration coefficients $c_1, c_2$ by Eqs.(18)~(19) and maximum velocity $V_{\max}$ by Eq.(20).

***Step 6:*** Sort the fitness values of all particles and update particle positions by Eq.(21).

***Step 7:*** Evolve the improved PSO algorithm and repeat step 4~step 6 until the maximum iterative times is satisfied.

***Step 8:*** Finish the evolution and plot the optimal path.

Table 1 Pseudo code of the proposed method

| | |
|---|---|
| | **// Environment construction** |
| 1 | Set flying space by Eq.(1); |
| 2 | Set two mountains by Eq.(2); |
| 3 | Set one radar by Eq.(3); |
| 4 | Set starting position and destination position for 10 UAVs |
| | **// PSO initialization** |
| 5 | Set the swarm parameters $S, D, w_{\max}, w_{\min}, c_{\max}, c_{\min}, V_1, V_2, T$; |
| 6 | Set UAV number as 10; |
| 7 | Set chaos parameter $\mu = 4$; |
| 8 | Initialize particles using Eq.(17) |
| 9 | Set a relative large value for fitness value; |
| 10 | Calculate particle fitness value and select gbest and pbest |
| | **// Main loop** |

For $i = 1$ *to UAVnumber*
  For $j = 1$ *to* $T$
    Update PSO parameters by Eq.(16) and Eqs.(18)~(20);
    Generate new particle position and velocity by Eq.(15);
    Calculate particle fitness value;
    Sort fitness value of all particles;
    Select gbest and pbest;
    Remember the value and the iteration for gbest;
    Adopt mutation strategy and generate new particle position by Eq.(21);
    $j = j + 1$
  End
  $i = 1 + 1$;
End

**// Output**

Output the planned paths for the formation

## 4. Simulation analysis

In order to illustrate the performance of the proposed method, simulations and comparisons are carried out in this section. The simulations are implemented in MATLAB environment and are run on a server with 3.6GHz CPU, 4.00GB of RAM and 64-bit operating system.

### 4.1 Parameter setting

Tabel 2 Mountain and radar model parameters

| Model kind | Parameter setting (km) |
|---|---|
| Flying space | $20\times20\times10$ |
| $z_{m,1}$ | $h_1 = 4, x_1^{m0} = 7, y_1^{m0} = 6, x_1^t = 4, y_1^t = 4$ |
| $z_{m,2}$ | $h_2 = 5, x_2^{m0} = 8, y_2^{m0} = 15, x_2^t = 4, y_2^t = 4$ |
| $\boldsymbol{T}_1$ | $x_1^r = 12, y_1^r = 10, z_1^r = 0, R_1 = 3.7$ |

In the simulation scenario, a formation with 10 UAVs are demanded to fly simultaneously from the starting points to the destination points under the certain environment. Two mountains and one radar are contained in the space which is modeled by Eqs.(2)~(3). The corresponding environment parameters are given in

Table 2. The starting and destination points for UAVs are shown in Table 3, where the starting points are in the shape of a straight line and the destination points are in the shape of a circle and they are matched by a PSO based task allocation algorithm in advance. The three dimensional environment and the endpoints are shown in Fig.1.

Tabel 3 UAV starting and destination positions

| Coordinate | No.1 | No.2 | No.3 | No.4 | No.5 | No.6 | No.7 | No.8 | No.9 | No.10 (km) |
|---|---|---|---|---|---|---|---|---|---|---|
| $x_s$ | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 |
| $y_s$ | 14 | 18 | 10 | 4 | 2 | 6 | 8 | 12 | 16 | 20 |
| $z_s$ | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 | 1.8 |
| $x_d$ | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| $y_d$ | 8.618 | 7.618 | 6.382 | 5.382 | 5 | 5.382 | 6.382 | 7.618 | 8.618 | 9 |
| $z_d$ | 3.1756 | 3.9021 | 3.9021 | 3.1756 | 2 | 0.8244 | 0.0979 | 0.0979 | 0.8244 | 2 |

where $(x_s, y_s, z_s)$ and $(x_d, y_d, z_d)$ respectively represent the three dimensional coordinates of the starting and destination points, $No.i, (i \in 1,...,10)$ is the UAV number label.

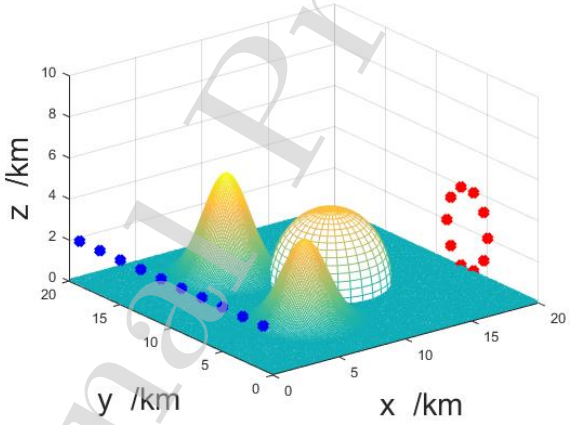



Figure 2 The flying area and endpoints

The parameters for the PSO algorithm are designed as follows: The swarm size is $S = 500$, the maximum iteration is $T = 120$. The waypoints, besides the endpoints, of each path is 20, the X-coordinate is obtained via uniformly dividing the range $[0.8, 20]$ into 20 segments, and the dimension for each particle is $D = 40$. For cost function parameters, the scale of intensity is $\delta = 3.5$, the safe distance for collision avoidance is $\bar{d} = 100m$.

In order to make a clear explanation and comparison, some abbreviations for different PSOs are used, where SPSO is the standard PSO, LCPSO is the PSO with linear-varying coefficients, LVPSO is the PSO with linear-varying maximum velocity,

CBPSO is the PSO with chaos-based initialization, PMPSO is the PSO with position mutation strategy, and CIPSO is the comprehensively improved PSO. The related parameters for different PSOs are shown in Table 4.

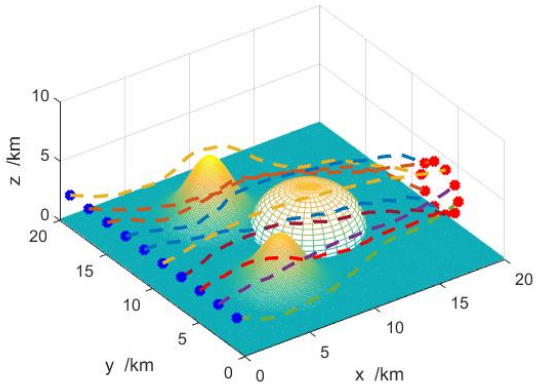
Tabel 4 Parameters for different PSOs

| PSO kind | Parameters |
|---|---|
| SPSO | $w_{\max}=0.9, w_{\min}=0.4, c_1=2, c_2=2, V_{\max}=0.3$ |
| LCPSO | $w_{\max}=0.9, w_{\min}=0.4, c_{\max}=3.5, c_{\min}=0.5, V_{\max}=0.3$ |
| LVPSO | $w_{\max}=0.9, w_{\min}=0.4, c_1=2, c_2=2, V_1=0.5, V_2=0.1$ |
| CBPSO | $w_{\max}=0.9, w_{\min}=0.4, c_1=2, c_2=2, V_{\max}=0.3, \mu=4$ |
| PMPSO | $w_{\max}=0.9, w_{\min}=0.4, c_1=2, c_2=2, V_{\max}=0.3, a=2$ |
| CIPSO | $w_{\max}=0.9, w_{\min}=0.4, c_{\max}=3.5, c_{\min}=0.5, V_1=0.5, V_2=0.1, \mu=4, a=2$ |

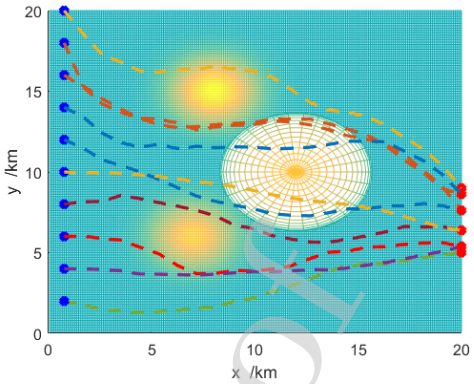
**4.2 Results and comparisons**

In this section, to evaluate the performance of PSO, we carry out several comparisons on different critical indicators, such as solution optimality, convergence speed, success rate and procedure running time. The simulation results for these different PSOs are given and analyzed, meanwhile the comparison of CIPSO with a recently proposed modified genetic algorithm is also carried out. All the comparison results clearly illustrate the effectiveness and advantages of the proposed improvements. The related results are shown in Figs.3~14 and Tables 5~8.

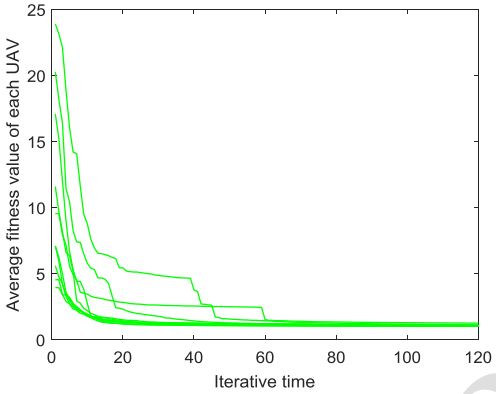
Figs.3~4 are the formation path planning results by CIPSO and SPSO, respectively, where the first two figures are the planned paths in three and two dimensional views. It is clear that feasible flying paths with different shapes are guaranteed by both methods and there is no confliction with the mountains and radar. Figs.3(c) and 4(c) show the evolution process for fitness value (FV), which is vital to evaluate PSO performance. It is straightforward that the FV of CIPSO converges faster than that of SPSO, thus the CIPSO maintains faster convergence speed than SPSO. Figs.3(d) and 4(d) are the UAV distances between each other, one can see that UAV distances during the flying process are always larger than the safe distance, thus collision avoidance is guaranteed by both methods.



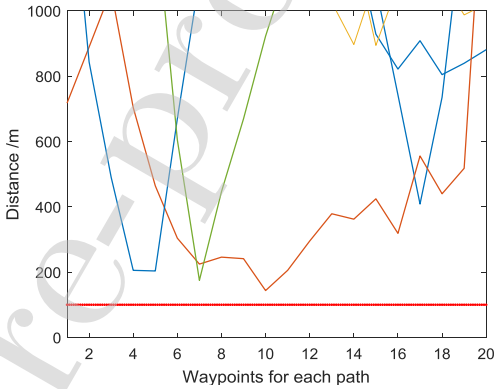

(a) Three dimensional view




(b) Two dimensional view

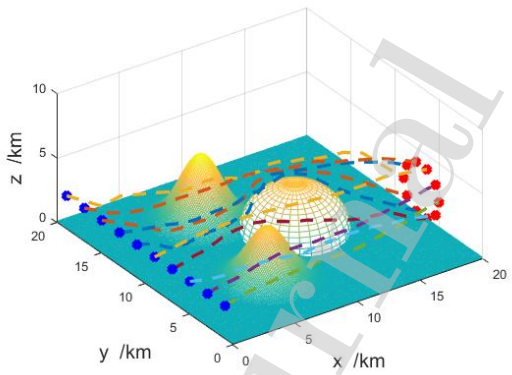



(c) Fitness value

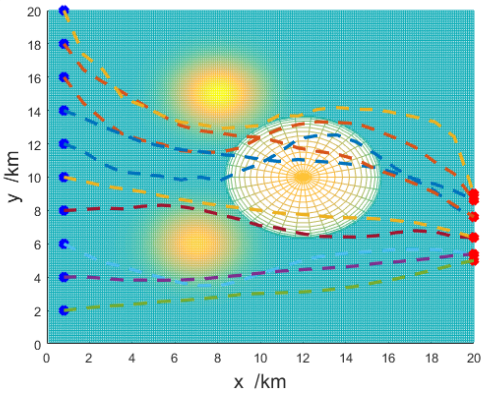



(d) UAV distance

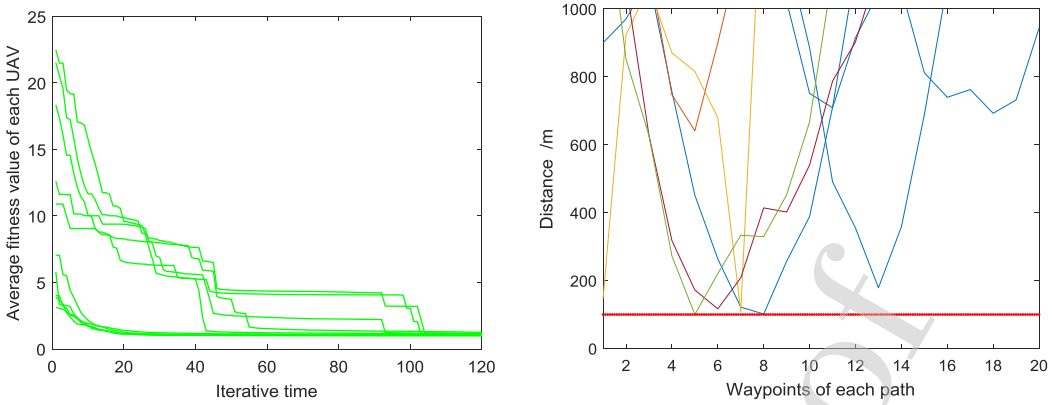
Figure 3 Evolution results of UAV formation by CIPSO




(a) Three dimensional view




(b) Two dimensional view

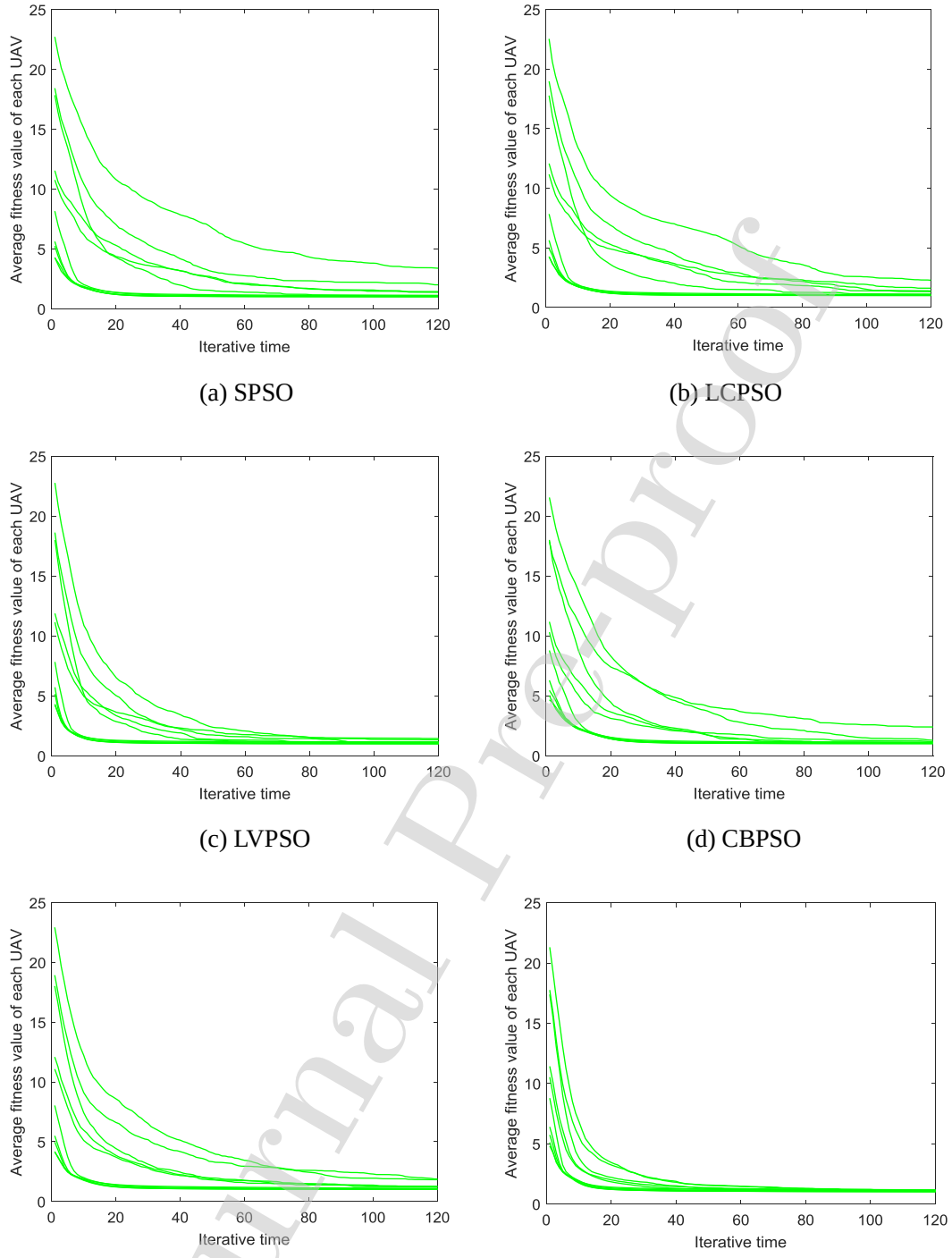



(c) Fitness value (d) UAV distance

Figure 4 Evolution results of UAV formation by SPSO

Due to the randomness of PSO searching process, it is not enough to evaluate the performance of different PSOs by running only one time. In order to make fair and convincing comparisons and to better show the advantages of the proposed improvements, we also carry out Monte-Carlo simulations for the proposed PSOs. In the following, the Monte-Carlo simulation times are set as 50 and the related results are shown from Figs. 5~10.

As a vital and comprehensive indicator for both convergence speed and solution optimality, the average fitness values (AFVs) for each UAV by different PSOs are firstly calculated in Fig.5. From the result, one can see that the LCPSO, LVPSO, CBPSO and PMPSO either have faster convergence speed or the final values are more centralized than those of SPSO, which means the convergence speed or solution optimality is enhanced. Besides, one can see that the CIPSO has the most centralized final values and the fastest convergence speed compared with SPSO, LCPSO, LVPSO, CBPSO and PMPSO, indicating that the CIPSO possesses the best performance among these PSOs. In the following, to show the advantages of CIPSO more clearly, solution optimality, convergence speed, success rate and running time are respectively analyzed and compared in detail.




(a) SPSO (b) LCPSO

(c) LVPSO (d) CBPSO

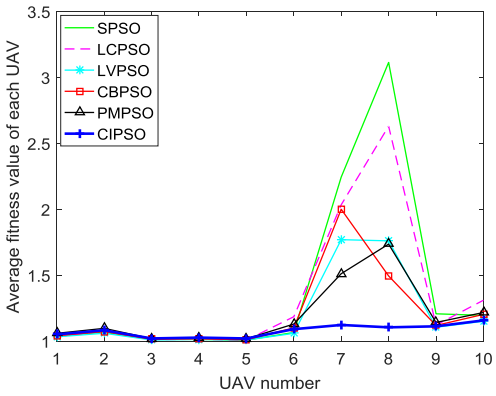
(e) PMPSO (f) CIPSO

Figure 5 Evolution of AFV by different PSOs

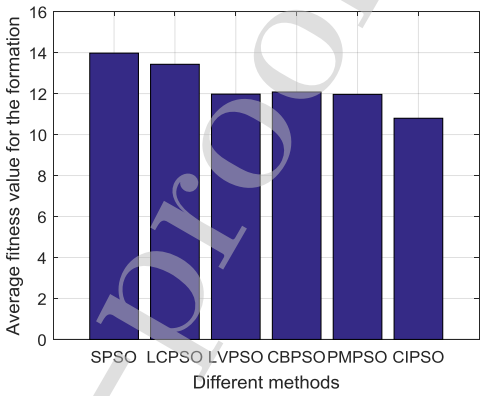
### 4.2.1 Solution optimality comparison

As the FV is closely related to PSO solution optimality and is a main objective in the design of PSO, we further carry on FV comparisons among different PSOs and UAVs. Fig.6 and Table 5 respectively show the AFV for each UAV and the formation during the Monte-Carlo simulations. From Fig.6 (a), it can be seen that the AFVs for

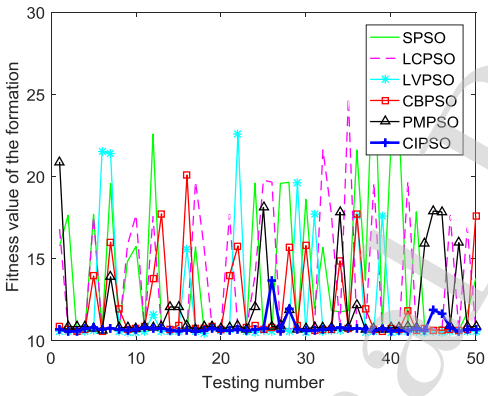
UAV 1~5 and UAV 6~10 vary in a large extent, this is mainly due to different environment constraints and will be analyzed with the combination of Fig.7 in Section 4.2.1. From Fig.6(a) and Table 5, one can see that the AFVs of UAV 1~5 are almost the same by different PSOs, while for UAV 6~10 the SPSO and CIPSO respectively maintain the maximum and minimum AFV and the AFVs by other methods are mainly in the middle.



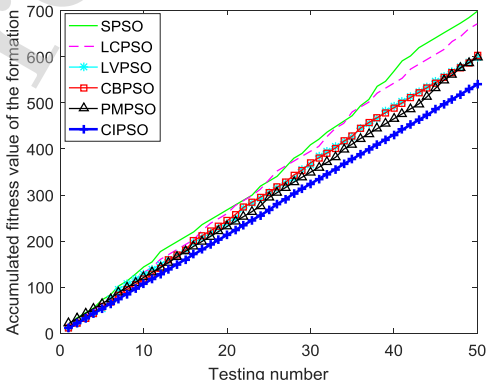

(a) AFV of each UAV




(b) AFV of the formation




(c) FV of the formation




(d) Accumulated FV of the formation

Figure 6 FV comparisons by different PSOs

From single UAV perspective, it is shown that the CIPSO does not always maintain less AFV than SPSO. However, from the formation perspective, one can see from Fig.6(b) and formation average fitness value (FAFV) in Table 5 that the CIPSO has the least AFV and the SPSO has the largest AFV. For UAV formation flying application, it is reasonable to insist that the formation AFV is more important and fair than single UAV AFV, since the formation AFV is an overall reflection of all members, while single UAV AFV is easily affected by different environment constraints and is not sufficient to evaluate the performance of the proposed PSO. Comparing with SPSO, the formation AFVs by LCPSO, LVPSO, CBPSO, PMPSO

and CIPSO are respectively reduced by 3.94%, 14.24%, 13.60%, 14.32%, 22.69%, thus one can conclude that each proposed improvement is effective in reducing formation AFV and enhancing the solution optimality, and the CIPSO performances the best among the PSOs. Next, by viewing the formation FV during the Monte-Carlo simulations in Fig.6(c), one can see that the formation FV by adopting each improvement is lower and more stable than that of SPSO, while the CIPSO is the lowest and most stable. The accumulated formation FV in Fig.6(d) also indicates that CIPSO maintains the least FV accumulated value, which is another illustration of better solution optimality of CIPSO.

Table 5 AFV comparison for each UAV and the formation

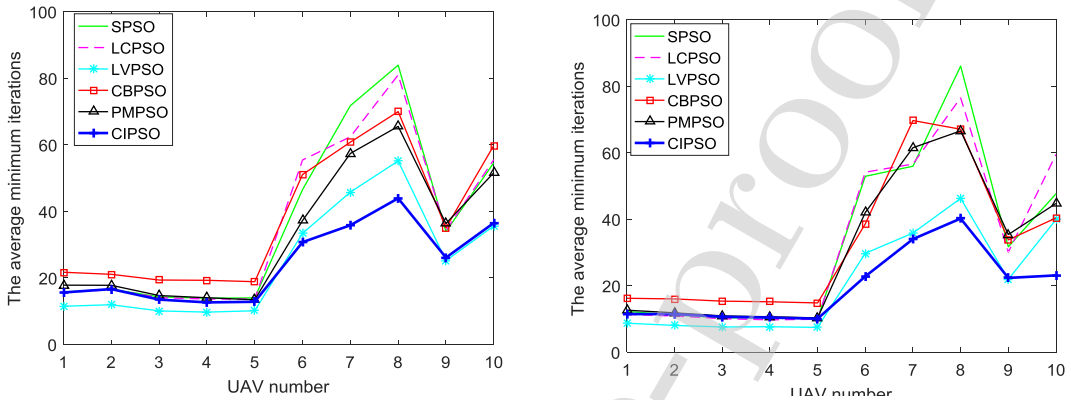
| Label | SPSO | LCPSO | LVPSO | CBPSO | PMPSO | CIPSO |
|---|---|---|---|---|---|---|
| UAV 1 | 1.04 | 1.03 | 1.03 | 1.04 | 1.06 | 1.05 |
| UAV 2 | 1.07 | 1.08 | 1.06 | 1.07 | 1.10 | 1.08 |
| UAV 3 | 1.01 | 1.01 | 1.01 | 1.01 | 1.02 | 1.02 |
| UAV 4 | 1.01 | 1.01 | 1.01 | 1.02 | 1.03 | 1.03 |
| UAV 5 | 1.01 | 1.01 | 1.01 | 1.01 | 1.02 | 1.02 |
| UAV 6 | 1.07 | 1.19 | 1.06 | 1.10 | 1.13 | 1.10 |
| UAV 7 | 2.24 | 2.03 | 1.78 | 2.00 | 1.51 | 1.12 |
| UAV 8 | 3.11 | 2.63 | 1.76 | 1.50 | 1.74 | 1.11 |
| UAV 9 | 1.21 | 1.12 | 1.11 | 1.12 | 1.14 | 1.11 |
| UAV 10 | 1.20 | 1.31 | 1.15 | 1.20 | 1.22 | 1.16 |
| FAFV | 13.97 | 13.42 | 11.98 | 12.07 | 11.97 | 10.8 |
| IP | N/A | 3.94% | 14.24% | 13.60% | 14.32% | 22.69% |

where FAFV means the formation AFV, and IP is the improved percentage of formation system compared with SPSO.

### 4.2.2 Convergence speed comparison

After the comparisons on FV, we carry on to analyze convergence speed among different PSOs, and adopt minimum iteration (MI), namely the necessary iteration to generate a feasible solution, as an indicator to evaluate the convergence speed. To show this property, the average minimum iteration (AMI) of each PSO during the Monte-Carlo simulations is calculated in Fig. 7. In the simulation, two constant values $F=1.5$ and $F=1.8$ are respectively set as criteria of feasible solution. From Fig.7, it is obvious the AMIs of UAV 1~5 are in the same level, while those for UAV 6~10 differ a lot, meaning that the searching speed for UAV 1~5 by different methods is almost the same but that for UAV 6~10 is different. Combining with Fig.6(a), Table 5

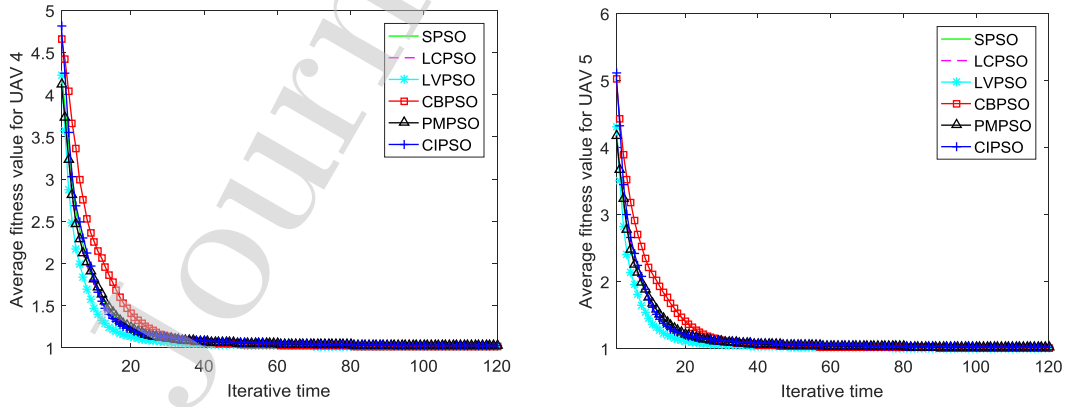
and the endpoints distribution of the formation, it is easy to deduce the reason is that the flying area for UAV 1~5 is relatively simple while that for UAV 6~10 is affected by the mountains and radar. For UAV 1~5 in Figs.7(a) and (b), the LVPSO maintains a little smaller AMIs than CIPSO, but for UAV 6~10 the CIPSO maintains much smaller AMIs than LVPSO, meaning the searching ability of CIPSO is better than LVPSO in complicated environment.



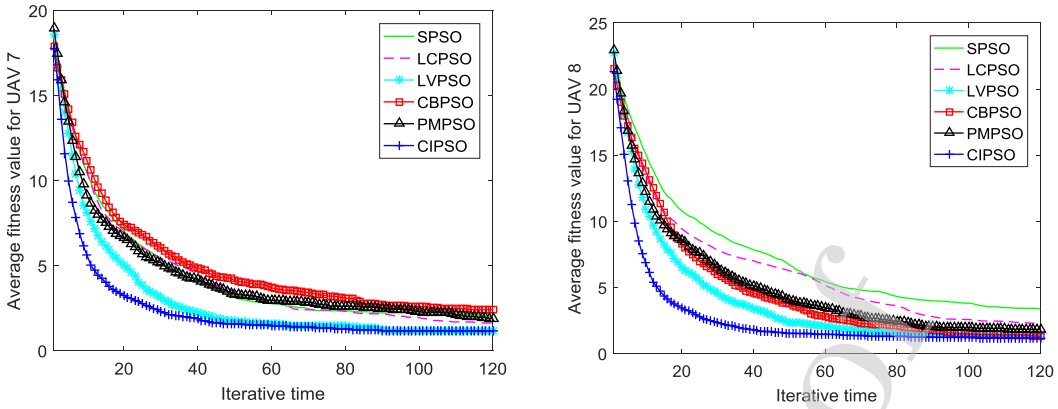

(a) AMI under $F=1.5$ (b) AMI under $F=1.8$

Figure 7 AMIs comparison by different PSOs

To further illustrate the performance of CIPSO, several different UAV AFVs are respectively shown in Fig.8. For space limit, only UAV 4~5 and UAV 7~8 are chosen as the candidates. One can see that the CIPSO and the SPSO maintain almost the same AMIs and AFVs for UAV 4~5 while the CIPSO shows better performance than SPSO both in AMI and AFV for UAV 7~8, which illustrates the faster convergence speed and better solution optimality of CIPSO.




(a) AFV of UAV 4 (b) AFV of UAV 5




(c) AFV of UAV 7 (d) AFV of UAV 8

Figure 8 AFV comparisons for different UAVs

Table 6 AMI for formation by different methods

| Indicator | SPSO | LCPSO | LVPSO | CBPSO | PMPSO | CIPSO |
|---|---|---|---|---|---|---|
| FAMI | 365.5 | 362.2 | 248.7 | 377.2 | 326.1 | 244.5 |
| IP | N/A | 0.89% | 31.97% | -3.20% | 10.79% | 33.11% |

where FAMI means the formation AMI, and IP is the improved percentage compared with SPSO.

Besides these results, the total AMI for the formation system is also given and compared in Table 6. One can see the CIPSO maintains the least AMI for the whole formation system and the improved percentage for CIPSO is 33.11%, which is a significant advantage compared with SPSO. Comprehensively, considering the AMI comparisons in above, the CIPSO guarantees the fastest convergence speed for the formation system among all the proposed PSOs.

**4.2.3 Success rate and running time comparison**

Apart from the solution optimality and convergence speed, success rate for flying mission and procedure running time are also important factors in evaluating the performance of PSO. Herein, we give the related results and show them in Figs.9~10, where Fig.9 is the total failure path number during the Monte-Carlo simulations, and Fig.10 is the average running time for each UAV. The failure path is defined as the one whose final FV satisfies $f_{(T=120)} > 2$, which means the path is too long or the UAV flies into the mountain or radar area or UAV collision happens. From Fig.9, it is straightforward that SPSO has the most failures, each improved PSO has less failures than SPSO, and CIPSO has the least failures, which is about one seventh of SPSO. In Table 7, the total path failure rate during the Monte-Carlo simulations is also given, one can see that the formation path failure rate for CIPSO is only 0.80%, meaning

almost all planned paths are feasible and the improved percentage compared with SPSO is 85.71%. We can conclude that the CIPSO guarantees significant improvement and can achieve the highest success rate. For running time in Fig.10, it is obvious the average running time for the several PSOs are almost the same and are all about 1.5s, proving that the proposed improvements do not increase additional time consuming and the CIPSO can still be realized in a time efficient way.

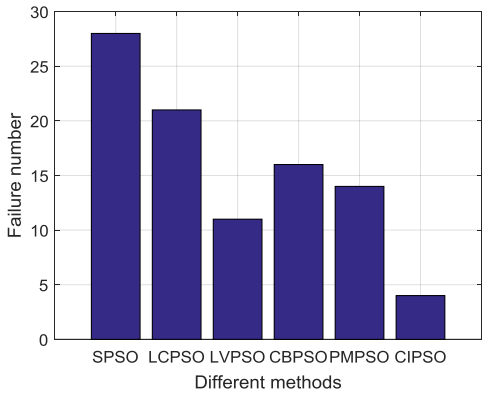



Figure 9 Failure path number comparison

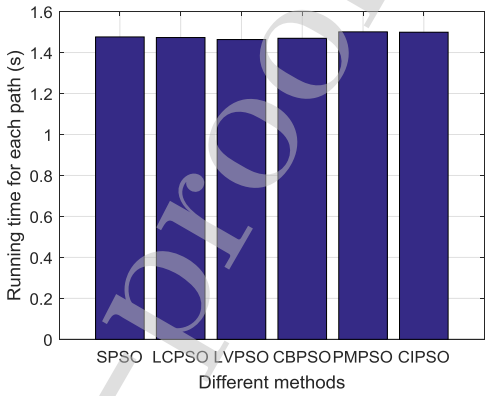



Figure 10 Average running time comparison

Table 7 Formation path failure rate and comparison

| Indicator | SPSO | LCPSO | LVPSO | CBPSO | PMPSO | CIPSO |
|---|---|---|---|---|---|---|
| FPFR | 5.60% | 4.20% | 2.20% | 3.20% | 2.80% | 0.80% |
| IP | N/A | 25.00% | 67.71% | 42.86% | 50.00% | 85.71% |

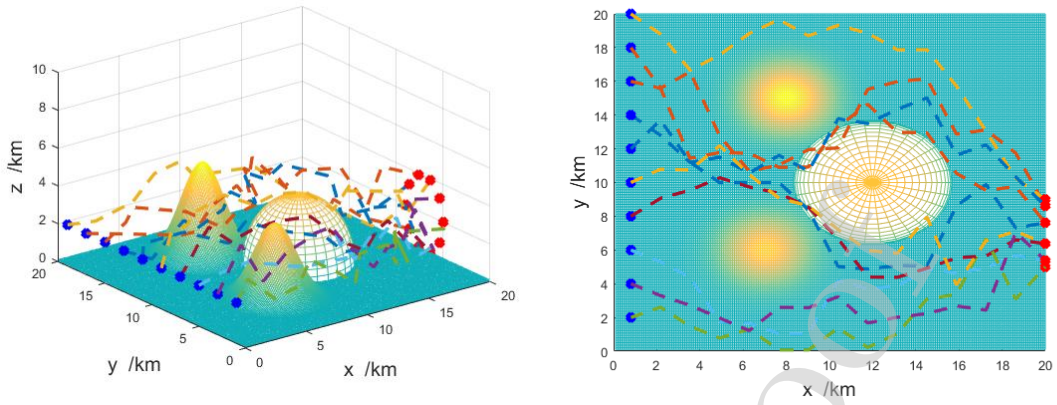
where FPFR means the formation path failure rate, and IP is the improved percentage compared with SPSO.

Finally, for the UAV formation system, it is concluded that the proposed improvements can improve solution optimality, speed up convergence and raise success rate within the same running time. Compared with SPSO, the CIPSO maintains significant advantages and can produce a much better and faster feasible path.

**4.2.4 Comparison with other method**

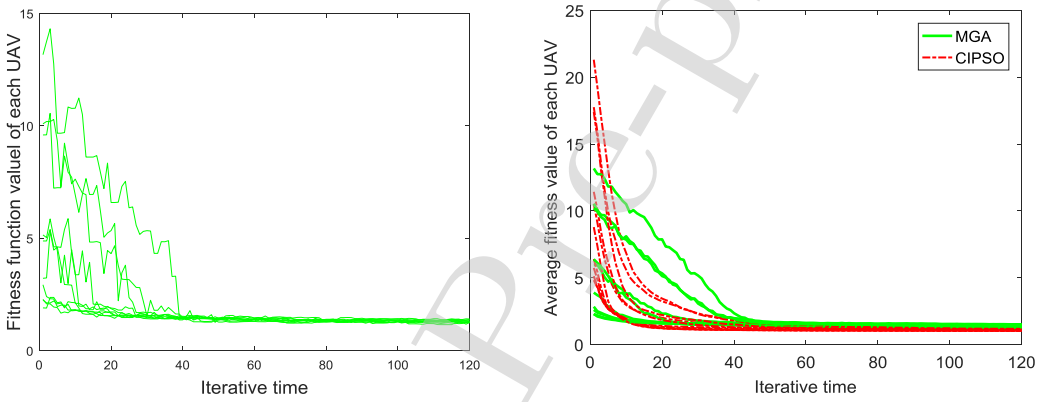
Besides the comparisons with different PSOs, we have also made some comparisons with the commonly used genetic algorithm. A recently proposed modified genetic algorithm (MGA) in Ref [31] is adopted here, which increases the exploration capability of the standard GA. For fair comparison, the population size and maximum iterations are set the same as those of CIPSO, namely, $S = 500$ and $T = 120$. By trial and error, the generation gap is chosen as 0.9 and the mutation probability is 0.05. The Monte-Carlo simulation times is also 50. The following

Figs.11~14 and Table 8 show the comparison results.




(a) Three dimensional view (b) Two dimensional view

Figure 11 Formation paths by MGA

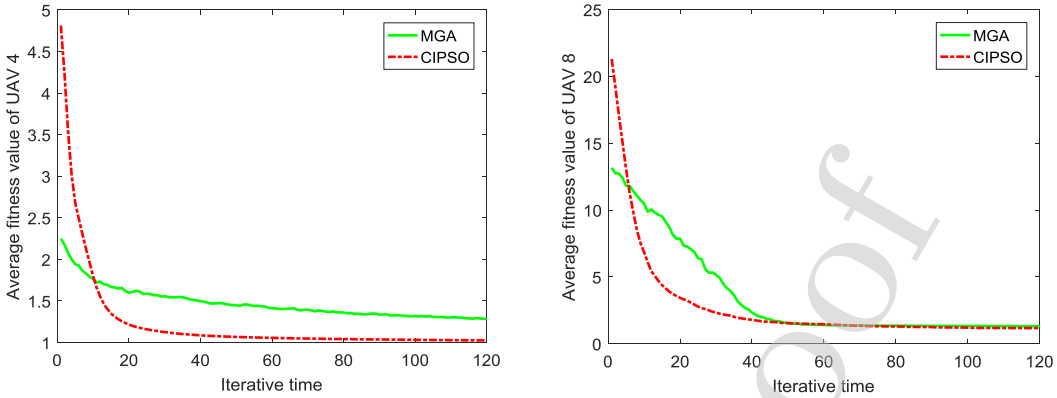



(c) FV by MGA (d) AFV comparison between MGA and CIPSO

Figure 12 FV and comparison

Fig.11 is the formation paths generated by MGA in three and two dimensional views. Compared with Fig.4, it seems that the paths of MGA are not smooth and much longer than those of CIPSO. Fig.12 shows the FV of MGA and the AFV comparison with CIPSO, where Fig.12(a) is the FV result by running MGA one time, and Fig.12(b) is the AFV during Monte-Carlo simulations and the comparison with CIPSO. Intuitively, the convergence speed and solution optimality of MGA is not as well as those of CIPSO. For clarity, UAV 4 and 8 are chosen as candidates for further comparison in the limited space. One can see that CIPSO maintains faster convergence speed and better solution optimality than MGA. In Table 8, the exact values of AFV for MGA and CIPSO are also given. From both single UAV and formation perspectives, the improved percentage of AFV by CIPSO is about 20%. In Fig.14, the AMI under the same criterion $F=1.5$ and the average procedure running

time are respectively shown, which further illustrate the faster convergence of CIPSO and also indicates CIPSO is much more time-saving than MGA.



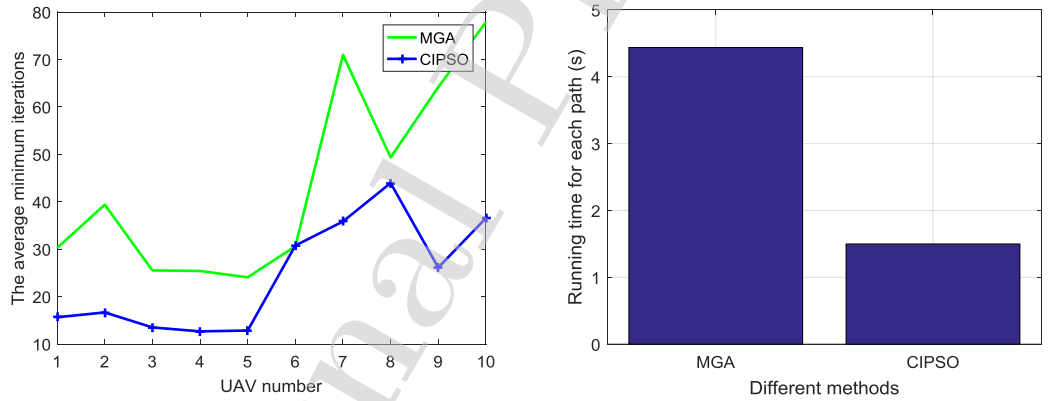

(c) AFV comparison of UAV 4 (d) AFV comparison of UAV 8

Figure 13 AFV comparison for UAV 4 and 8

Table 8 AFV comparison for each UAV

| Label | No.1 | No.2 | No.3 | No.4 | No.5 | No.6 | No.7 | No.8 | No.9 | No.10 | FAFV |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MGA | 1.33 | 1.42 | 1.28 | 1.28 | 1.26 | 1.32 | 1.51 | 1.32 | 1.44 | 1.49 | 13.65 |
| CIPSO | 1.05 | 1.08 | 1.02 | 1.03 | 1.02 | 1.10 | 1.12 | 1.11 | 1.11 | 1.16 | 10.8 |
| IP | 21.1% | 24.0% | 20.3% | 19.5% | 19.1% | 16.7% | 25.9% | 15.9% | 22.9% | 22.1% | 20.9% |

where FAFV means the formation AFV, and IP is the improved percentage compared with MGA.




(c) AMI comparison (d) Running time comparison

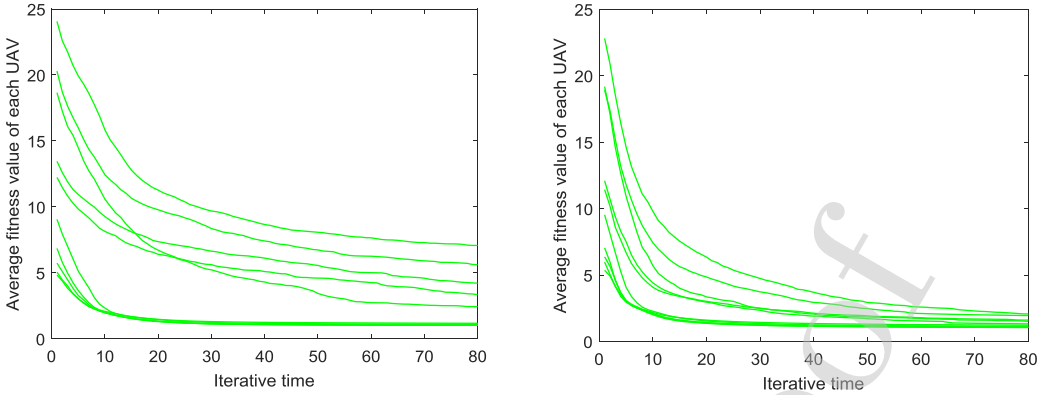
Figure 14 AMI and running time comparison between MGA and CIPSO

In a word, by evaluating convergence speed, solution optimality and procedure running time, the CIPSO shows better performance than MGA, which illustrates the advantages of the proposed CIPSO.

### 4.3 Results with less iterations and swarm size

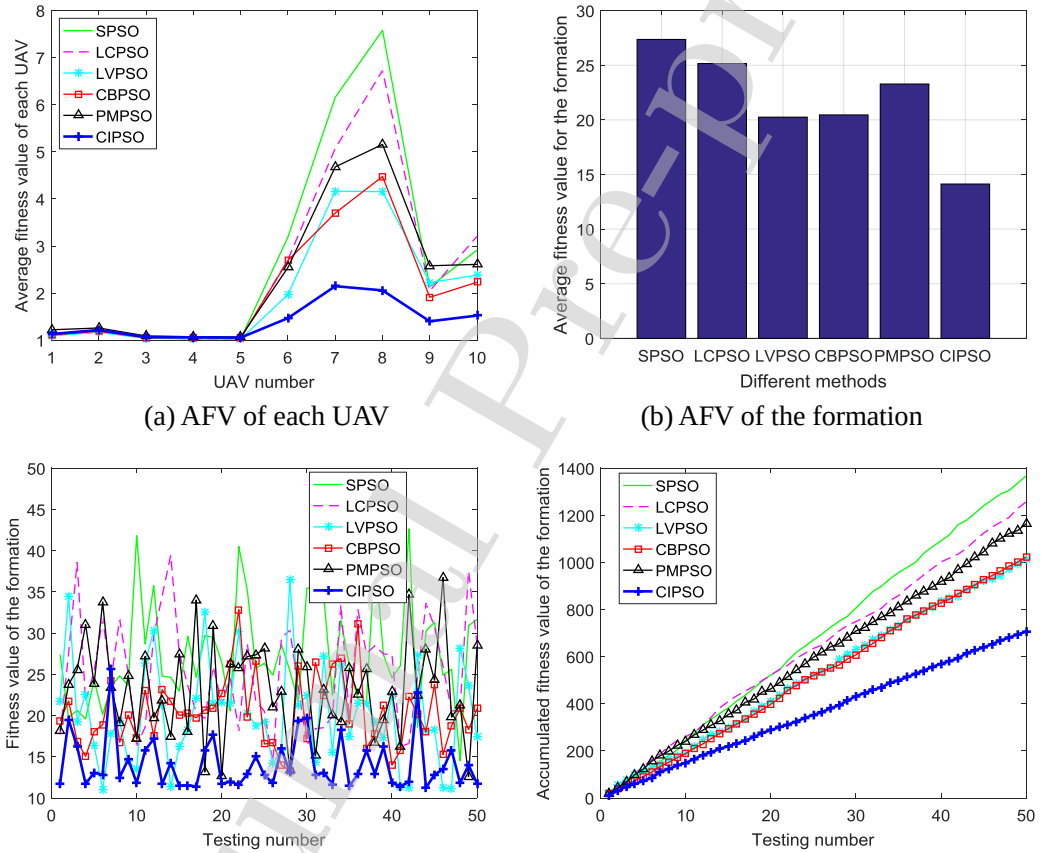
In order to further analyze and compare the performance of the proposed improvements and CIPSO, another group of simulations with much less iterations and smaller swarm size are carried out. The maximum iterations and swarm size are respectively reduced as $T = 80$ and $S = 200$. Other parameters are the same as

previous. The corresponding results are shown in Figs.15~19.




(a) SPSO (b) CIPSO

Figure 15 Evolution of AFV by SPSO and CIPSO




(a) AFV of each UAV (b) AFV of the formation

(c) FV of the formation (d) Accumulated FV of the formation

Figure 16 FV comparison by different PSOs

In Fig.15, the evolution of AFV by SPSO and CIPSO is given. It is natural that the result is not as well as that in Fig.5, however, similar trends and advantages are still maintained by CIPSO comparing with SPSO. In Fig.16, the FV and accumulated FV comparisons by different PSOs are provided. It is clear that each proposed improvement can contribute to reduce the FV and the CIPSO maintains the smallest

FV for both single UAV and the formation system, which is in accordance with that in Section 4.2.

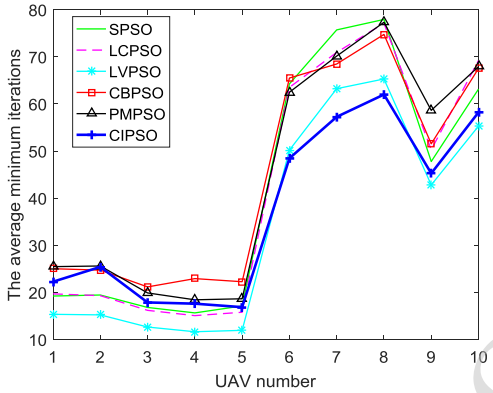



Figure 17 AMIs comparison by different PSOs

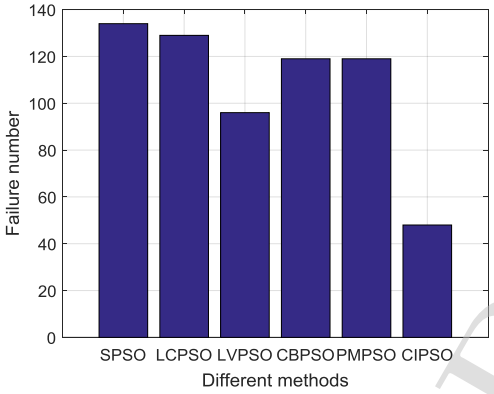



Figure 18 Failure path number comparison

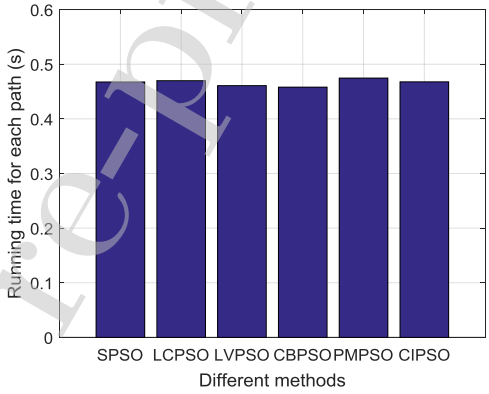



Figure 19 Average running time comparison

From Figs.17~19, the AMI, the total failure number and the running time comparisons are carried out. Obviously, the results are in the same trends as those in Section 4.2. Due to the reduction of maximum iterations and swarm size, the total path failures increase extremely, but CIPSO still maintains the least failures. The raised failure number also results in the increasing of AMI, which weakens the advantage of CIPSO, but the AMI of CIPSO is still the smallest for UAV 6~10. Finally, all PSOs take almost the same time to run the procedure.

Still, we can conclude the proposed improvements are effective and CIPSO shows the best even with insufficient maximum iterations and swarm size.

## 5. Conclusion

This paper researches the comprehensively improved PSO algorithm for UAV formation path planning under environmental and collision avoidance constraints. The chaos-based initialization, the liner-varying maximum velocity and acceleration coefficients and the position mutation strategy are respectively proposed and analyzed

in the paper. For fair and convincing evaluation, Monte-Carlo simulations for the formation with 10 UAVs under different environment constraints are carried out. Plenty of comparisons with the SPSO and MGA illustrate the advantages of the proposed CIPSO. The key findings are summarized as follows:

1). The CIPSO can achieve better solution optimality than the SPSO and the MGA for the formation system.

2). The formation convergence speed by CIPSO is accelerated and much faster than that by SPSO and MGA.

3). The formation success rate is also improved and no additional running time is needed for CIPSO.

In the future, we will focus on formation path planning with the combination of UAV 3-DOF model. Path re-planning when unexpected threats or moving obstacles are encountered would also be considered for future work. Besides, how to establish the relationship between the flying time and the planned path, such as investigating the PSO based pseudo-spectral method, would also be an interesting and meaningful research issue.

## References

1. George J, Sujit P B, Sousa J B. Search strategies for multiple UAV search and destroy missions. Journal of Intelligent & Robotic Systems, 2011; 61(1-4): 355-367.

2. Ortiz-Peña H J, Sudit M, Hirsch M, et al. A multi-perspective optimization approach to UAV resource management for littoral surveillance. In: Proceedings of the 16th international conference on information fusion. IEEE; 2013, p. 492-498.

3. Tomic T, Schmid K, Lutz P, et al. Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. IEEE Robotics & Automation Magazine, 2012; 19(3): 46-56.

4. Zheng C, Li L, Xu F, et al. Evolutionary route planner for unmanned air vehicles. IEEE Transactions on Robotics, 2005; 21(4): 609-620.

5. Besada-Portas E, de la Torre L, Jesus M, et al. Evolutionary trajectory planner for multiple UAVs in realistic scenarios. IEEE Transactions on Robotics, 2010; 26(4): 619-634.

6. Ma Y, Hu M, Yan X. Multi-objective path planning for unmanned surface vehicle with

currents effects. ISA transactions, 2018; 75: 137-156.

7. Chen YB, Mei YS, Yu JQ, et al. Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm. Neurocomputing, 2017; 266: 445-457.

8. Pehlivanoglu YV. 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.

9. Bayili S, Polat F. Limited-Damage A*: A path search algorithm that considers damage as a feasibility criterion. Knowledge-Based Systems, 2011, 24(4): 501-512.

10. Baumann M, Léonard S, Croft E A, et al. Path planning for improved visibility using a probabilistic road map. IEEE Transactions on Robotics, 2010; 26(1): 195-200.

11. Brubaker M A, Geiger A, Urtasun R. Map-based probabilistic visual self-localization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016; 38(4): 652-665.

12. Kothari M, Postlethwaite I. A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees. Journal of Intelligent & Robotic Systems; 2013, 71(2): 231-253.

13. Moon C, Chung W. Kinodynamic planner dual-tree RRT (DT-RRT) for two-wheeled mobile robots using the rapidly exploring random tree. IEEE Transactions on Industrial Electronics, 2015; 62(2): 1080-1090.

14. Chen YB, Yu JQ, Su XL, et al. Path planning for multi-UAV formation. Journal of Intelligent & Robotic Systems, 2015; 77(1): 229-246.

15. Chen Y, Luo G, Mei Y, et al. UAV path planning using artificial potential field method updated by optimal control theory. International Journal of Systems Science, 2016; 47(6): 1407-1420.

16. Yao P, Wang H, Su Z. Cooperative path planning with applications to target tracking and obstacle avoidance for multi-UAVs. Aerospace Science and Technology, 2016, 54: 10-22.

17. Ma H, Shen S, Yu M, et al. Multi-population techniques in nature inspired optimization algorithms: A comprehensive survey. Swarm and Evolutionary Computation, 2019; 44: 365-387.

18. Zhao Y, Zheng Z, Liu Y. Survey on computational-intelligence-based UAV path planning. Knowledge-Based Systems, 2018; 158: 54-64.

19. Bansal JC, Gopal A, Nagar A K. Stability analysis of artificial bee colony optimization

algorithm. Swarm and Evolutionary Computation, 2018; 41: 9-19.

20. Gao C, Zhen Z, Gong H. A self-organized search and attack algorithm for multiple unmanned aerial vehicles. Aerospace Science and Technology, 2016; 54: 229-240.

21. Fister I, Fister Jr I, Yang XS, et al. A comprehensive review of firefly algorithms. Swarm and Evolutionary Computation, 2013; 13: 34-46.

22. Xia X, Xing Y, Wei B, et al. A fitness-based multi-role particle swarm optimization. Swarm and Evolutionary Computation, 2019; 44: 349-364.

23. Lin A, Sun W, Yu H, et al. Global genetic learning particle swarm optimization with diversity enhancement by ring topology. Swarm and Evolutionary Computation, 2019; 44: 571-583.

24. Lee W, Kim DE. Adaptive approach to regulate task distribution in swarm robotic systems. Swarm and Evolutionary Computation, 2019; 44: 1108-1118.

25. Karimi J, Pourtakdoust SH. Optimal maneuver-based motion planning over terrain and threats using a dynamic hybrid PSO algorithm. Aerospace Science and Technology, 2013; 26(1): 60-71.

26. Yang P, Tang K, Lozano JA, et al. Path planning for single unmanned aerial vehicle by separately evolving waypoints. IEEE Transactions on Robotics, 2015; 31(5): 1130-1146.

27. Huang C, Fei JY. UAV path planning based on particle swarm optimization with global best path competition. International Journal of Pattern Recognition and Artificial Intelligence, 2018; 32(06): 1859008, DOI.org/10.1142/S0218001418590085.

28. Liu Y, Zhang X, Guan X, et al. Adaptive sensitivity decision based path planning algorithm for unmanned aerial vehicle with improved particle swarm optimization. Aerospace Science and Technology, 2016; 58: 92-102.

29. Tian D, Shi Z. MPSO: Modified particle swarm optimization and its applications. Swarm and Evolutionary Computation, 2018; 41: 49-68.

30. Xia X, Xing Y, Wei B, et al. A fitness-based multi-role particle swarm optimization. Swarm and Evolutionary Computation, 2019; 44: 349-364.

31. Elhoseny M, Tharwat A, Hassanien A E. Bezier curve based path planning in a dynamic field using modified genetic algorithm. Journal of Computational Science, 2018, 25: 339-350.

## Conflict of interest

We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.