Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Cooperative 3D Path Planning of Multi-UAV Via Improved Fruit Fly Optimization

Yongjie Mao
Southwest Jiaotong University School of Electrical Engineering
Deqing Huang
Southwest Jiaotong University School of Electrical Engineering
Na Qin (qinna@swjtu.edu.cn)
Southwest Jiaotong University School of Electrical Engineering https://orcid.org/0000-0003-3227-974X
Lei Zhu
Southwest Jiaotong University School of Electrical Engineering
Jiaxi Zhao
Southwest Jiaotong University School of Electrical Engineering

Research Article

Keywords: Unmanned Aerial Vehicle (UAV), Path Planning, Time Stamp Segmentation (TSS), Fruit Fly Optimization Algorithm

Posted Date: December 22nd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1124343/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Cooperative 3D Path Planning of Multi-UAV Via Improved Fruit Fly Optimization

Yongjie Mao[1], Deqing Huang[1], Na Qin[1*], Lei Zhu[1] and Jiaxi Zhao[1]

School of Electrical Engineering, Southwest Jiaotong University, Chengdu, 611756, Sichuan, China.

*Corresponding author(s). E-mail(s): qinna@swjtu.edu.cn;
Contributing authors: Maoyj@my.swjtu.end.cn; elehd@home.swjtu.edu.cn;
zl861940030@163.com; zjx19980602@163.com;

Abstract

Path planning of multiple unmanned aerial vehicles (UAVs) is a crucial step in cooperative operation of multiple UAVs, whose main difficulties lie in the severe coupling of time and three-Dimensional (3D) space as well as the complexity of multi-objective optimization. For this purpose, the time stamp segmentation (TSS) model is first adopted to resolve the time-space coupling among multiple UAVs. Meanwhile, the solution space is reduced by transforming the multi-objective problem to a multi-constraint problem. In consequence, based on the elite retention strategy, a novel improved fruit fly optimization algorithm (NIFOA) is proposed for multi-UAV cooperative path planning, which overcomes the shortcomings of basic fruit fly optimization algorithm in slow convergence speed and the potentials to fall into local optima. In particular, the multi-subpopulations evolution mechanism is further designed to optimize the elite subpopulation. At last, the effectiveness of the proposed NIFOA has been verified by numerical experiments.

Keywords: Unmanned Aerial Vehicle (UAV), Path Planning, Time Stamp Segmentation (TSS), Fruit Fly Optimization Algorithm

1 Introduction

With the development of aviation and aerospace industry and Internet technology, unmanned aerial vehicle (UAV) plays an important role in military reconnaissance and strike (Nex and Remondino, 2014), aerial photography (Flores et al., 2017), agricultural plant protection (Khan et al., 2019), police security (Li and Zhen, 2021), oilfield inspection (Ge et al., 2020), etc. Due to the increased difficulty of the task and the complexity of the environment, single UAV operations have been unable to meet the task requirements. Therefore, much effort has been put forward towards the cooperative operation of multiple UAVs, where multi-UAV cooperative path planning is one of the research hotspots.

In response to the complex path planning issue of UAVs, researchers have proposed all kinds of methods, e.g., the methods based on graph analysis, such as voronoi diagram (Candeloro et al., 2017, Niu et al., 2019), Dijkstra algorithm (Zhu and Sun, 2021), A* algorithm (Lai et al., 2021, Mandloi et al., 2021), D* algorithm (Dakulović et al., 2011, Yao et al., 2021), artificial potential field algorithm (APF) (Kumar et al., 2019, Luo

et al., 2015), and the Dubins curve method (Shanmugavel et al., 2010). Although these algorithms are outstanding in computational speed and timeliness, the motion characteristics of the UAVs are not considered. To resolve this, swarm intelligence algorithms are increasingly used to address the corresponding multi-objective optimization problems (Alejo et al., 2013, Han et al., 2016, He et al., 2021, Li et al., 2014, 2021a,b, Liu et al., 2011, Miao et al., 2021, Shao et al., 2019, Xu et al., 2020, Zhang and Duan, 2015, 2018), including particle swarm optimization (PSO) algorithm (Alejo et al., 2013, He et al., 2021, Shao et al., 2019), ant colony optimization (ACO) algorithm (Li et al., 2021a,b, Miao et al., 2021), artificial bee colony (ABC) algorithm (Li et al., 2014, Xu et al., 2020), pigeon-inspired optimization (PIO) algorithm (Zhang and Duan, 2015, 2018), wolf colony algorithm (WCA) (Han et al., 2016, Liu et al., 2011), fruit fly optimization algorithm (FOA) (Bai et al., 2021, Pan, 2012, Zhang et al., 2018), and their improved versions. Notice that swarm intelligence algorithms are frequently used to solve the Non-deterministic Polynomial hard problems, and most of them have the shortcomings of slow convergence speed and local optimization.

FOA is a new swarm intelligence algorithm proposed by Pan in 2011 (Pan, 2012), which realizes the iterative search of the solution space by simulating the process of fruit fly using smell and vision to prey. It has the advantages of easy-to-understand principle, simple process, and strong local search ability. Therefore, FOA is widely used in fault diagnosis (Li et al., 2021c), image processing (Nazir et al., 2021), data prediction (Li and Jiang, 2019), path planning (Bai et al., 2021, Pan, 2012, Zhang et al., 2018), and many other fields. In addition, FOA can optimize neural network parameters (Li and Jiang, 2019), optimize support vector machines parameters (Gu et al., 2021, Luo et al., 2019), adjust controller parameters (Wang et al., 2017), and solve traveling salesman problems (Iscan and Gunduz, 2017). Recently, much effort has been devoted to improving the premature and local optimization performance of FOAs. For instance, (Yuan et al., 2017) proposed an improved fruit fly optimization algorithm (IFOA), which combines the average learning and step-change strategy to balance the algorithm's global and local search capabilities; (Yuan et al., 2014) proposed a multi-group FOA, in which the multiple subgroups move independently in the search space to explore the global optimal solution. Although these algorithms render to remarkable optimization performance, certain limitations still exist in the aspects of large search space for feasible solutions and large number of invalid solutions (Tian and Li, 2019).

Recently, more improved FOAs have been proposed for path planning, and achieved remarkable results. Among them, (Li, 2014) proposed a 3D path planning scheme for robots in virtual environment based on the improved FOA, which promoted the planning efficiency. Meanwhile, (Zhang et al., 2018) proposed a phase angle-coded FOA with mutation adaptive mechanism (θ-MAFOA) for path planning of single UAV. The cooperative path planning of multi-UAV based on multi-group FOA is proposed in (Shi et al., 2020), which considers the strategy of cooperative collision avoidance among UAVs. In addition, (Li et al., 2020) proposed an ORPFOA algorithm for path planning of multiple UAVs, where the tasks are changed online, and the mission completion time and the collision avoidance strategy among UAVs are taken into account simultaneously. Nevertheless, although these methods have improved the classic FOA algorithm in different aspects and been applied to 3D path planning, the performance constraints (e.g flight speed, pitch angle, turn angle) and simultaneous arrival problems of UAVs are not considered comprehensively.

In this paper, a Novel Improved Fruit fly Optimization Algorithm (NIFOA) is proposed to solve the aforementioned problems. First, the greedy and restart strategies are adopted to guide individual search and avoid falling into local optima. Second, the multiple subpopulation evolution mechanism is introduced into the better population to achieve the mutation optimization population. Third, the final result is obtained by integrating with the cooperative strategy. Finally, the algorithm is applied with time stamp segmentation model to solve the cooperative path planning problem of multiple UAVs.

The remainder of the paper is organized as follows: Section 2 introduces the TSS model and formulates the UAV planning problem mathematically. Section 3 presents the key idea of basic FOA and its improved version. Section 4 shows

the experimental results. Section 5 concludes the work.

2 Problem Formulation

Multi-UAV cooperative path planning is based on single-UAV path planning, considering the multi-constraint and multi-objective problem of mutual cooperation and restriction among multiple UAVs, as shown in Fig. 1. The constraints include the limitations of the UAV (maximum yaw angle, maximum pitch angle, maximum flight speed, minimum flight speed, etc.), information sharing, collision avoidance, etc, and the objective function needs to consider the minimum total cost of the UAV cluster and the potential of reaching the destination at the same time.

Path planning is the first problem to be solved in the cooperative path planning of multiple UAVs. Given a starting point and a targeting point, the main task of path planning is to plan one or more paths that meet the constraint requirements. In this paper, the method of generating route control points is used to connect the control points in turn to generate the track. Let $S(S_{m,x}, S_{m,y}, S_{m,z})$ and $T(T_{m,x}, T_{m,y}, T_{m,z})$ denote the starting and targeting points of the mth UAV. Next, the straight line ST is divided into $D+1$ segments by the straight line $L_k, k = 1, ..., D$ perpendicularly. Then, take one point on each straight line L_k, and connect these points in turn to get the path from S to T, as shown in Fig. 2. As such, the path planning problem is transformed into a key path point optimization problem. In order to speed up the optimization process and complete the synergy constraints, this work quotes the time stamp segmentation model (TSS) (Zhang and Duan, 2018).

2.1 Time Stamp Segmentation Model

In this work, the TSS model is used to process the time coordination of multi-UAV cooperative path planning, which divides the flight time into several time periods by setting the same take-off time and arrival time for all UAVs. Let T_a and T_g be the common take-off time and arrive time of all UAVs, and D the number of timestamps. Then, the time stamp t_s is defined as $t_s = (T_g - T_a)/D$. Further, it is necessary to determine two components in

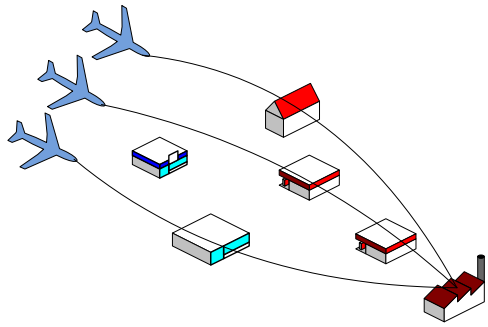

Fig. 1 The environment of multi-UAV

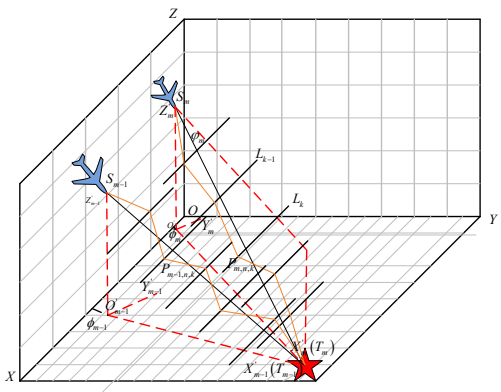

Fig. 2 The path representation in 3D view of multi-UAV

the TSS model beforehand, namely the velocity component

$$V_{m,n} = [V_{m,n,x}, V_{m,n,y}, V_{m,n,z}]$$

and the position component

$$P_{m,n} = [P_{m,n,x}, P_{m,n,y}, P_{m,n,z}],$$

where $V_{m,n,x}, V_{m,n,y}, V_{m,n,z}, P_{m,n,x}, P_{m,n,y}$ and $P_{m,n,z}$ represent the velocity components and position components in x-axis, y-axis, and z-axis, respectively. The relationship between $P_{m,n}$ and $V_{m,n}$ can be expressed as

$$P_{m,n} = P_{m,n-1} + V_{m,n} * t_s \tag{1}$$

where $V_{m,n}$ is the average velocity of the nth time stamp of the mth UAV, where $m = 1, 2, \cdots, M$ represents the number of UAVs, and

$n = 1, 2, \cdots, D$ represents the number of timestamps.

In the TSS model, the flight speed and the number of timestamps of UAV are determined firstly. Then, the time stamp is marked on the x-axis between the take-off point and the targeting point of the UAV (without loss of generality, assuming that $d_x > d_y$ with d_x and d_y being the x-axis and y-axis distance from the take-off point to the targeting point). Moreover, assume that $[V_{min}, V_{max}]$ is the speed range of the UAV, and $l_{m,x}$ is the x-axis Euclidean distance between the take-off point and the targeting point of the mth UAV. Then, the range of the time stamp is:

$$\begin{cases} t_s \in \bigcap_{m=1}^{M} [T_{min}, T_{max}] \\ T_{m,min} = \frac{l_{m,x}}{V_{max} D} \\ T_{m,max} = \frac{l_{m,x}}{V_{min} D} \end{cases} \tag{2}$$

As shown in Fig. 3, in order to improve the search efficiency of the best path, a new local coordinate system is constructed. In the new local coordinate, the z-axis is unchanged, $S_m T_m$ is set as the x-axis, S_m coincides with the new coordinate origin O'_m, and the angle between $O_m X_m$ and $O'_m X'_m$ is ϕ_m, that is, ϕ_m is the angle at which $O_m X_m$ rotates counterclockwise to $O'_m X'_m$. Then the transformation between the two coordinate systems is:

$$\begin{bmatrix} \overline{P}_{m,n,x} \\ \overline{P}_{m,n,y} \\ \overline{P}_{m,n,z} \end{bmatrix} = \begin{bmatrix} \cos(\phi_m) & \sin(\phi_m) & 0 \\ -\sin(\phi_m) & \cos(\phi_m) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} P_{m,n,x} \\ P_{m,n,y} \\ P_{m,n,z} - P_{m,0,z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ P_{m,0,z} \end{bmatrix} \tag{3}$$

In the local coordinate system, assume that the velocity of UAV in the x-axis direction is constant (when the starting point and the targeting point is with the same x-axis coordinate, the velocity in the y-axis direction is chosen to be constant), that is $\overline{V}_{m,n,x} = l_{m,x}/Dt_s$. In such a way, the position component of $P_{m,n,x}$ in the x-axis direction is also known. Then, the velocities of UAV in the y-axis direction and z-axis direction are as follow:

$$\overline{V}_{m,n,y} = \overline{V}_{m,n,x} \cdot \tan(\beta_{m,n}) \tag{4}$$

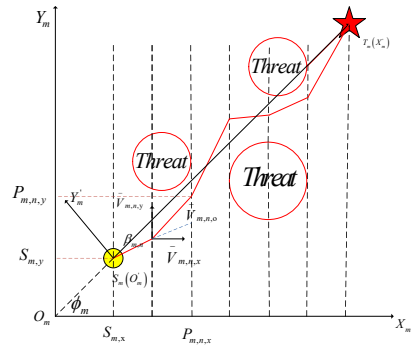

Fig. 3 The path representation of 2D view

$$P_{m,n,y} = \overline{V}_{m,n,y} \cdot t_s + P_{m,n-1,y} \tag{5}$$

$$\overline{V}_{m,n,z} = \sqrt{\overline{V}^2_{m,n,x} + \overline{V}^2_{m,n,y}} \cdot \tan(\alpha_{m,n}) \tag{6}$$

$$P_{m,n,z} = \overline{V}_{m,n,z} \cdot t_s + P_{m,n-1,z} \tag{7}$$

where $\beta_{m,n}$ is the horizontal turning angle of the mth UAV at the nth time stamp, and $\alpha_{m,n}$ is the pitch angle of the mth UAV at the nth time stamp. Note that $P_{m,0}$ represents the starting point and $P_{m,D+1}$ represents the targeting point. By virtue of (3)-(7), it renders to

$$P_{m,n,x} = V_{m,n,x} * t_s + P_{m,n-1,x} \tag{8}$$

$$V_{m,n,y} = V_{m,n,x} \tan(\beta_{m,n} + \phi_m) \tag{9}$$

$$V_{m,n,z} = \sqrt{V^2_{m,n,x} + V^2_{m,n,y}} * \tan(\alpha_{m,n}) \tag{10}$$

$$P_{m,n,y} = V_{m,n,y} * t_s + P_{m,n-1,y} \tag{11}$$

$$P_{m,n,z} = V_{m,n,z} * t_s + P_{m,n-1,z} \tag{12}$$

From (8)-(12), it can be seen that $P_{m,n}$ is only related to $\alpha_{m,n}$ and $\beta_{m,n}$ when $V_{m,n,x}$ is given. Hence, the key path point optimization problem is

transformed into a problem of finding the optimal $\alpha_{m,n}$ and $\beta_{m,n}$.

In the following, the objective function and restrictions design for the multi-UAV cooperative path planning is conducted.

2.1.1 Objective Function and Restrictions Design

Designing reasonable multi-restraints conditions and multi-objective optimization function is crucial for solving the path planning problem, where terrain threats, radar threats, path length cost as well as the UAV's performance and coordination cost, etc., should be considered together.

(1) The cost of terrain threat. UAVs cannot collide with obstacles such as mountains in flight, so a feasible path should be able to avoid all obstacles. Assume that the 3D coordinates of the mth UAV at the kth path point is $H_{m,k}(x_{m,k}, y_{m,k}, z_{m,k})$, and $f(x_{m,k}, y_{m,k})$ denotes the height of the current point $(x_{m,k}, y_{m,k})$ in the map. The terrain threat cost can be expressed mathematically as follows:

$$J_{m,k}^{terrain} = \begin{cases} Q, & z_{m,k} < f(x_{m,k}, y_{m,k}) \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where $Q > 0$ is the penalty constant.

(2) The cost of radar threat. The threat area of a UAV is denoted by a hemisphere with a radius of R_{max}, where the probability of a UAV being threatened is related to its distance from the radar $R_{m,k}$. The radar threat cost can be formulated as (Belew et al., 1990):

$$J_{m,k}^{radar} = \begin{cases} \dfrac{R_{max}^4 Q}{R_{max}^4 + R_{m,k}^4}, & R_{m,k} \le R_{max} \\ 0, & R_{m,k} > R_{max} \end{cases} \tag{14}$$

where Q is same as in (13).

(3) The cost of path length. The path length cost of a UAV can be considered as the sum of all path segments of the UAV, where the path length of the mth UAV at the kth time stamp is

$$J_{m,k}^{length} = \big((x_{m,k} - x_{m,k-1})^2 + (y_{m,k} - y_{m,k-1})^2 + (z_{m,k} - z_{m,k-1})^2\big)^{1/2} \tag{15}$$

(4) The cost of path smoothing. To describe the smoothness of the path, the angle between the adjacent path segments is taken as the path smoothing cost, namely, for the mth UAV at the kth time stamp

$$J_{m,k}^{smoothing} = \arccos\left(\frac{S_{m,k} S_{m,k-1}}{|S_{m,k}||S_{m,k-1}|}\right) \tag{16}$$

where $S_{m,k} = (x_{m,k+1} - x_{m,k}, y_{m,k+1} - y_{m,k}, z_{m,k+1} - z_{m,k})$ is the directed track segment corresponding to the kth time stamp.

(5) The cost of path altitude. The altitude cost reflects the flying altitude of UAV on the ground. When flying at low altitudes, UAV prefers to using terrain cover to avoid radar detection. Owing to the existence of safe altitude h_{safe}, the path height cost can be calculated in the following way.

$$J_{m,k}^{altitude} = |h_{m,k} - h_{safe}| \tag{17}$$

where $h_{m,k}$ is the height of the mth UAV at the kth time stamp.

(6) The cost of UAV's constraint. Since UAV is always constrained by its own structure during flight, such as flight speed, yaw angle, and pitch angle, the self-constrained cost of the mth UAV at the kth time stamp can be expressed as:

$$J_{m,k}^{constraint} = J_{m,k}^{turn} + J_{m,k}^{slope} + J_{m,k}^{velocity}, \tag{18}$$

with

$$J_{m,k}^{turn} = \begin{cases} 0, & -\gamma_{min} \le \beta_{m,k} \le \gamma_{max} \\ Q, & \text{otherwise} \end{cases} \tag{19}$$

$$J_{m,k}^{slope} = \begin{cases} 0, & -\varphi_{min} \le \alpha_{m,k} \le \varphi\_max \\ Q, & \text{otherwise} \end{cases} \tag{20}$$

$$J_{m,k}^{velocity} = \begin{cases} 0, & v\_m, min \le v\_m, k \le v_{m,max} \\ Q, & \text{otherwise} \end{cases} \tag{21}$$

where $\gamma_{min} > 0, \gamma_{max} > 0, \varphi_{min} > 0, \varphi_{max} > 0, 0 < v_{m,min} < v_{m,max}$ are all pre-specified constants, and Q is same as in (13) and (14).

(7) The cost of coordination. UAVs cannot collide with each other during the flight, otherwise the coordination cost is expensive. When the mth UAV collides with the nth UAV, the cost of

coordination can be expressed as:

$$J_{m,n,k}^{coordination} = \begin{cases} Q, & |P_{m,k} - P_{n,k}| \leq l_{uav,max} \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

where $P_{m,k} = (x_{m,k}, y_{m,k}, z_{m,k})$ is the position of the mth UAV at the kth time stamp, $|P_{m,k} - P_{n,k}|$ is the Euclidean distance between $P_{m,k}$ and $P_{n,k}$, and $l_{uav,max} > 0$ is a given threshold.

In summary, the objective function of UAV path planning is the weighted sum of the above 7 cost functions (He et al., 2021), i.e.,

$$\begin{aligned} J_m =& w_1 \sum_{i=1}^{D} J_{m,k}^{terrain} + w_2 \sum_{i=1}^{D} J_{m,k}^{radar} + w_3 \sum_{i=1}^{D} J_{m,k}^{length} \\ &+ w_4 \sum_{i=1}^{D} J_{m,k}^{smothing} + w_5 \sum_{i=1}^{D} J_{m,k}^{altitude} \\ &+ w_6 \sum_{i=1}^{D} J_{m,k}^{constraint} + w_7 \sum_{i=1}^{D} J_{m,k}^{coordination} \end{aligned} \tag{23}$$

where $w_i, i = 1, 2, \cdots, 7$ are the weight coefficients of the cost function J_m. Notice that variations of the weighting coefficients could lead to huge differences in the optimization result. In the following, to optimize the paths for UAVs more efficiently, some cost function components are processed as constraints, e.g., the terrain threat, the radar threat and the UAV's constraint.

The remaining components are then stratified, namely, the path length/smoothing/height cost is used as the first/second/third level cost function. When the difference in path length cost of the same path is less than a certain threshold, a second level comparison is performed. Further, if the difference of path smoothing cost is less than a certain threshold, the third level comparison is performed.

3 The Proposed Algorithm

In the multi-UAV cooperative path planning, the TSS-based path planning model is adopted to resolve the spatial-temporal coupling problem of multiple UAVs. In detail, the model transforms the path planning into an optimization problem of the angles between the UAV velocity and the x-axis/xoy-plane, i.e., the horizontal rotation angle and the pitch angle. As illustrated in Section 2, the path points corresponding to the time stamps can be obtained, once the horizontal yaw angle and the pitch angle are determined and the velocity components corresponding to each axis of the UAV are available. Now, we are at the position of exploiting NIFOA to search the optimal solutions of the path planning model.

3.1 The Basic Fruitfly Optimization Algorithm

FOA is a global optimization algorithm by imitating the foraging behavior of fruit flies. Fruit flies are superior to other species in perception, especially in smell and vision. The olfactory organs of fruit flies are good at collecting various odors floating in the air, and can even smell food sources 40 km away. The process of searching for food by fruit flies can be divided two phases: on one hand, the fruit flies smell the food source using olfactory organs and then try to fly there; on the other hand, when the fruit flies are close to food, keen vision is adopted to make sure where the food and their companions are. Inspired by the characteristics of fruit flies in searching for food, FOA is proposed and the pseudo code is summarized in Algorithm 1.

Step 1: Initialization of parameters, e.g., the maximum number of iterations (gen_{max}), the population size (M_{pop}). Meanwhile, randomly initialize the location (X_axis, Y_axis) of the fruit fly population:

$$\begin{aligned} X\_axis &= rand(-1, 1) \\ Y\_axis &= rand(-1, 1) \end{aligned} \tag{24}$$

where $rand(-1, 1)$ is a uniformly distributed random number in $(-1, 1)$.

Step 2: Each fruit fly is with a random distance and direction, and the sense of smell is used to find food along the following way:

$$\begin{aligned} X_i &= X\_axis + rand(-1, 1) \\ Y_i &= Y\_axis + rand(-1, 1) \end{aligned} \tag{25}$$

where X_i and Y_i, $i = 1, 2, \cdots, M_{pop}$ denote the position coordinate components of the ith fruit fly, and the function $rand(-1, 1)$ is same as in (24).

Step 3: Considering that the location of the food is unknown, the distance ($Dist$) of the fruit fly to the origin is first estimated via (26), then

Algorithm 1 Pseudo code of Standard FAO Algorithm

for $gen = 1$ to gen_{max} **do**
  for $M = 1$ to $M = M_{pop}$ **do**
    Random initial the position of fruitfly population.
    $X\_axis = rand(-1, 1)$
    $Y\_axis = rand(-1, 1)$.
    Give fruit fly individuals a random distance and direction to randomly search for food using their sense of smell using (25).
    Calculate the distance from the fruit fly to the origin (26)
    Compute the taste concentration smell using (27).
    Evaluate the taste concentration ($Smell_i$) using (28).
    Record bestSmell and bestIndex using (29).
    Update X_axis and Y_axis
  end for
  if $Smell_i > bestSmell$ **then**
    $X\_axis = X(bestIndex)$
    $Y\_axis = Y(bestIndex)$
  end if
end for

(27) is used to calculate the taste concentration judgement value (S), which is actually the reciprocal of the distance.

$$Dist_i = \sqrt{X_i^2 + Y_i^2} \tag{26}$$

$$S_i = \frac{1}{Dist_i} \tag{27}$$

Step 4: With the taste concentration judgement value S, the taste concentration ($Smell$) at the individual location of fruit fly is calculated by the pre-defined taste concentration decision function (28).

$$Smell_i = function(S_i) \tag{28}$$

Step 5: Seek the fruit fly with the highest/lowest taste concentration among the fruit fly population:

$$[bestSmell, bestIndex] = max(Smell) \tag{29}$$

where $bestSmell$ is the highest taste concentration, $bestIndex$ is the index of the fruit fly who has the highest/lowest taste concentration, and $Smell$ is the set of the taste concentrations of all fruit flies.

Step 6: Keep the best taste concentration and the corresponding position coordinate such that the fruit fly population can fly there immediately.

$$\begin{aligned} Smellbest &= bestSmell \\ X\_axis &= X(bestIndex) \\ Y\_axis &= Y(bestIndex) \end{aligned} \tag{30}$$

where $X(bestIndex)$ and $Y(bestIndex)$ denote the position coordinate of the fruit fly who has the highest (the lowest) smell concentration, and $Smellbest$ the global optimal smell concentrate.

Step 7: Iterative optimization. Judge whether the taste concentration in the current iteration is greater than that in the previous iteration. If so, perform Step 6; otherwise, repeat Steps 2-5.

3.2 The Novel Improved Fruitfly Optimization Algorithm

In order to speed up the search speed of fruit fly, improve the global search ability of FOA, and avoid premature convergence and falling into local optima, the original FOA is improved from the following aspects:

(1) Greedy and restart strategies. In the original FOA, fruit flies search for food randomly within a fixed range, which might lead to low search efficiency and convergence precision. For this reason, the presented NIFOA adopts the greedy strategy to reduce the search scope. To be specific, the search scope is divided into two phases during the whole evolutions by introducing a new smell contraction judgement parameter

$$S_i = \begin{cases} 1/2, & rand(0,1) < P \\ \dfrac{1}{Dist_i}, & \text{otherwise} \end{cases} \tag{31}$$

where $rand(0, 1)$ is a uniformly distributed random number in $(0, 1)$, $0 < P < 1$ is a constant, and the parameter $Dist_i$ is same as (27). In order to avoid falling into local optima, the restart strategy is introduced, i.e., when the fruit flies fail to find the food source within certain trials during the search, the fruit flies will restart the search again from the starting point.

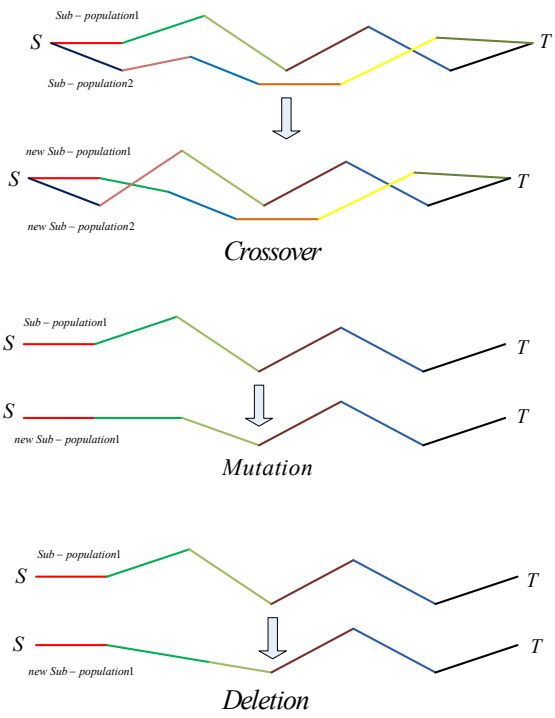

Fig. 4 Crossover, mutation and deletion of the evolution operators

(2) Multiple populations simultaneous search. The population is divided into several sub-populations of the same size so that the sub-populations move independently in the search space at the same time, searching for the global optimum. This strategy improves the diversity of solutions to avoid local optimum or premature convergence.

(3) Evolution mechanism. In order to make full use of the sub-optimal solution information in the population, the crossover, mutation and deletion operations are introduced to improve the quality of the solution, as shown in Fig. 4. As an efficient evolutionary operation for superior fruit flies, this technique makes full use of the knowledge of sub-optimal individuals by exchanging information so that the quality of optimal individual and the diversity of solutions are enhanced to avoid premature.

(4) Cooperative strategy. For cooperation of UAVs, traditional methods prefer to comparing all path points of any two UAVs, which greatly increases the time cost of path planning. To resolve this, the cooperation strategy is introduced. By taking the paths corresponding to the sub-populations in the optimal population as alternative paths, judge the distance between the path points corresponding to each time stamp to determine whether the synergy is satisfied, and finally combine the best sub-populations that satisfy the synergy to get the solution of path planning.

Under the framework of NIFOA, the implementation procedure of multi-UAV cooperative path planning is illustrated in Algorithm 2.

Algorithm 2 Pseudo code of Multi-UAV Cooperative Path Planning of NIFAO

Input: input the starting points and targeting points of all UAVs, environment map, the number of time stamp D

Output: The path point

/*Initialization*/
Set the maximum number of iterations $Itermax$, the fruit fly population M_{pop}, the fruit fly sub-population gr, and fruit fly individuals in each sub-population D, initialize the location of individual fruit flies randomly by (24).
for $N = 1$ to Num **do**
    Calculate the time stamp t_s, the velocity of fixed axis V_N, and the initial angle $angled(N)$
    for $NC = 1$ to $Itermax$ **do**
        for $m = 1$ to M_{pop} **do**
            while $g = 1$ to gr **do**
                while $d = 1$ to D **do**
                    for $i = 1$ to 2 **do**
                        $X_i = X_{axis} + (-1 + 2 * rand())$
                        $Y_i = Y_{axis} + (-1 + 2 * rand())$
                        $S = \frac{1}{\sqrt{X_i^2 + Y_i^2}}$
                    end for
                    while (1) **do**
                        if $S \leq 1$ **then** break
                        else
                            $X_i = X_{axis} + (-1 + 2 * rand())$
                            $Y_i = Y_{axis} + (-1 + 2 * rand())$
                        end if
                    end while

if $rand() \leq P$ **then**
$S = 1/2$
end if
$angle = (2 * S(odd) * \beta_{max} + \beta_{max}) + angle$
$angleu = \alpha_{max} + 2 * S(eve) * \alpha_{max}$
Calculating the vehical and position by (4)-(10)
if L(current point to targeting) $< L$(starting to targeting) and $z(x, y) > map(x, y)$ **then**
$d = d + 1$, $r = 0$
else
$r = r + 1$
end if
if $r \geq 30$ **then**
$d = 1, angle = angled(N)$
end if
end while
if $-\beta_{max} \leq \beta_D \leq \beta_{max}$ **then**
$J_{length} = \sum_{i=0}^{n+1} J_k^{length}$
else
$J_{length} = 2000$
end if
calculate $J_{altitude}$ and $J_{smoothing}$,
$g = g + 1$
end while
end for
Find out gr the best sub-population, and keep their locations.
operate mutation, crossover and deletion
uptate J_{length}, $J_{altitude}$, $J_{smoothing}$, and X_i, Y_i
end for
repeat line 5-line 48
end for
The sub-populations in the optimal population of each UAV are used as alternative paths, and the best path that satisfies the requirements of all UAVs is outputted by the cooperative strategy.

Step 1: Initialization. First set the maximum number of iterations $Itermax$, the fruit fly population M_{pop}, the fruit fly sub-population gr, and the number of fruit fly individuals in each sub-population D. Next, initialize the location of individual fruit flies randomly by (24).

Step 2: According to (25), individuals use smell to search for food with random direction and distance. The taste concentration values are calculated by (26) and (31).

Step 3: Substitute the taste concentration value (S_i) into fitness function or objective function so as to find the fitness function value ($Smell_i$) of the individual location of fruit fly by (28).

Step 4: According to (29), the sub-population with the smallest (biggest) taste concentration is found.

Step 5: Crossover, mutation and deletion operators are performed on the individuals in the best population. If it is better than the initial value, the original individuals are replaced with new ones, otherwise the original ones are kept.

Step 6: Keep the best taste concentration value and the X and Y coordinates. Meanwhile, the fruit fly colony uses vision to fly to this position by (30).

Step 7: Repeat Steps 2-6, until the maximum number of iterations is reached or the target requirements are met.

Considering that, by virtue of NIFOA, the output result of multi-UAV cooperative path planning is a series of path points, the path needs to be smoothed to meet the requirements of UAV flight. To address this, the B-spline curve is introduced and derived from the Bezier curve, and has the advantages of geometric invariance, convexity and continuous curvature (Elbanhawi et al., 2015). The B-spline curve can be structured by

$$P(u) = \sum_{j=i-k}^{i} P_i N_{i,k}(u) \tag{32}$$

where i is the track node number, P_i is control point, and $N_{i,k}(u)$ is the blending function that can be defined recursively as follows (Bai et al., 2018):

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, u_i \leq u \leq u_{i+1} \\ 0, textotherwise \end{cases} \\ N_{i,k}(u) = \frac{u-u_i}{u_{i+k}-u_i} N_{i,k-1}(u) + \frac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} N_{i+1,k-1}(u) \\ define \quad \frac{0}{0} = 0 \end{cases} \tag{33}$$

Table 1 The starting and targeting points of UAVs in Scenario 1

Number	Starting	Targeting
UAV_1	(10, 85, 22)	(67, 180, 6)
UAV_2	(155, 26, 6)	(76, 198, 10)
UAV_3	(126, 176, 10)	(140, 20, 6)
UAV_4	(30, 14, 10)	(200, 178, 10)
UAV_5	(104, 20, 8)	(140, 180, 15)
UAV_6	(9, 9, 12)	(160, 170, 20)
UAV_7	(32, 20, 9)	(32, 190, 5)

In this paper, the cubic B-spline method is used to ensure that the UAV track has better curve smoothness. In order to make the smoothed track as equal as possible to the original track, four adjacent control points are used to determine the spline curve that is flyable for UAV.

4 Experimental Evaluation and Comparison

In this section, two simulation experiments are conducted to evaluate the effectiveness of the proposed algorithm, namely, Scenario 1: multiple UAVs fly to different targeting points at the same time, and Scenario 2: multiple UAVs reach the same targeting points at the same time. The 3D virtual environment model is built on the Matlab platform, which facilitates the observation of the planned paths.

4.1 Scenario 1: Multiple UAVs fly to different targeting points at the same time

In the simulation experiment, the experimental environment is the entire space of the 3D map with a size of 200*200*85, including mountain terrain, radar, severe weather, and other environments. To verify the algorithm performance for different path conditions, 7 UAVs flying to 7 different targeting points are considered. The starting and targeting points corresponding to each UAV are shown in Table 1. The 2D and 3D view results of path planning are shown in Figs. 5 and 6, respectively, from which we can see that the proposed algorithm can effectively generate feasible paths that meet all the pre-specified constraints. In particular, all the obstacles are avoided, and there are no collisions among UAVs.

Table 2 The starting and targeting points of UAVs in Scenario 2

Number	Starting	Targeting
UAV_1	(197, 32, 5)	(32, 190, 5)
UAV_2	(32, 20, 9)	(32, 190, 5)
UAV_3	(22, 22, 5)	(200, 178, 10)
UAV_4	(104, 20, 8)	(200, 178, 10)
UAV_5	(190, 73, 5)	(129, 190, 5)
UAV_6	(140, 20, 6)	(129, 190, 5)

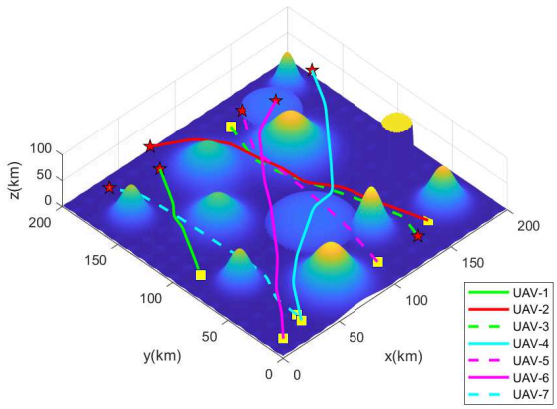

Fig. 5 3D view of the planned path in Scenario 1

4.2 Scenario 2: Multiple UAVs fly to the same targeting points at the same time

In the experiment, six identical UAVs are simulated, where the starting and targeting points of UAVs are set in Table 2. The 2D and 3D views of path planning are demonstrated in Figs. 7 and 8. Similar to Scenario 1, the proposed NIFOA can generate paths that satisfy all the constraints for UAVs.

4.3 Algorithm Comparison

In order to verify the superiority of the proposed algorithm, a comparison of PSO, FOA, IFOA and NIFOA, in generating multi-UAV paths, has been done under the condition of Scenario 1. In the experiment, the total path cost function that combines the path length cost, the path smoothing cost and the path height cost is used as the objective function, and each algorithm is tested 50 times to calculate the best cost, worst cost, average cost and variance, as summarized in Table 3. Figs. 9 and 10 show the convergence

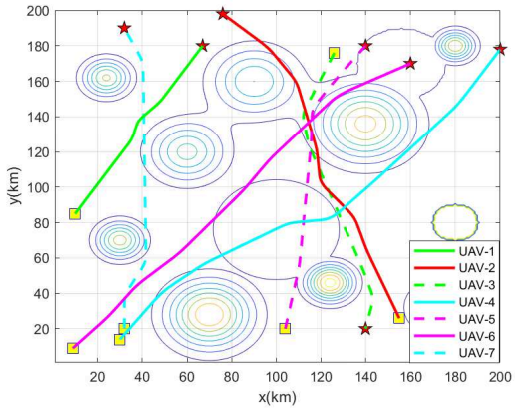

Fig. 6 2D view of the planned path in Scenario 1

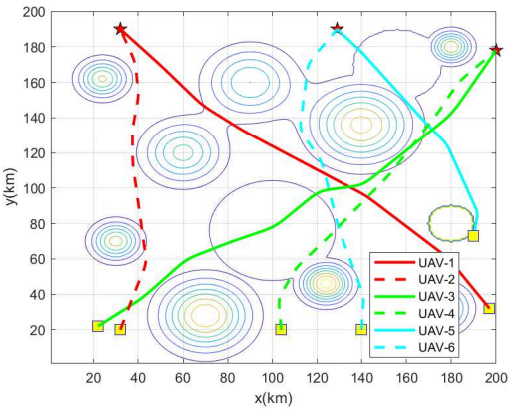

Fig. 8 2D view of the planned path in Scenario 2

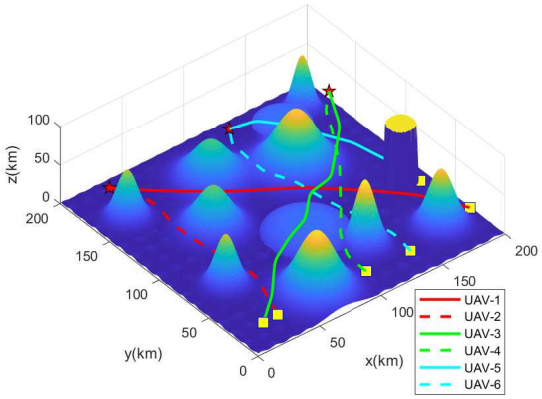

Fig. 7 3D view of the planned path in Scenario 2

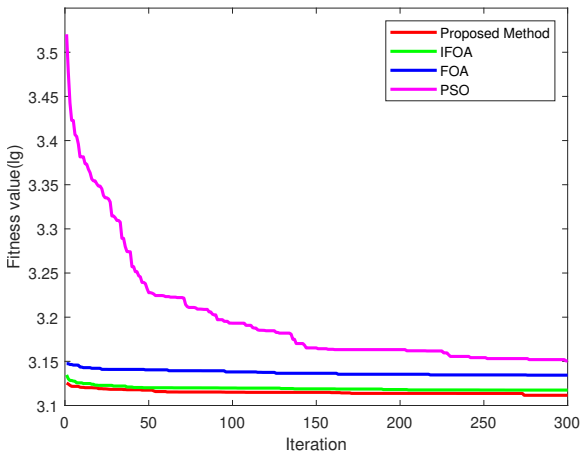

Fig. 9 Convergence speed comparison of the four algorithms

Table 3 The comparison results of the four algorithms

Algorithm	Best	Worst	Mean	Std(10^{-4})
Proposed Method	3.1116	3.1156	3.1138	7.0222
IFOA	3.1140	3.1175	3.1162	7.3710
FOA	3.1262	3.1343	3.1304	19
PSO	3.1505	3.2514	3.1759	252

curves and the path lengths of the four algorithms in 50 experiments, where NIFOA achieves the fastest convergence speed, the smallest cost and the shortest path length. The 2D, 3D views and height diagram for one of the planned paths are shown in Figs. 11, 12 and 13, respectively, implying that the planned path corresponding to NIFOA has the shortest path distance, the lowest average height, and the best smoothness. In such sense, the efficiency of the proposed NIFOA algorithm has been verified clearly.

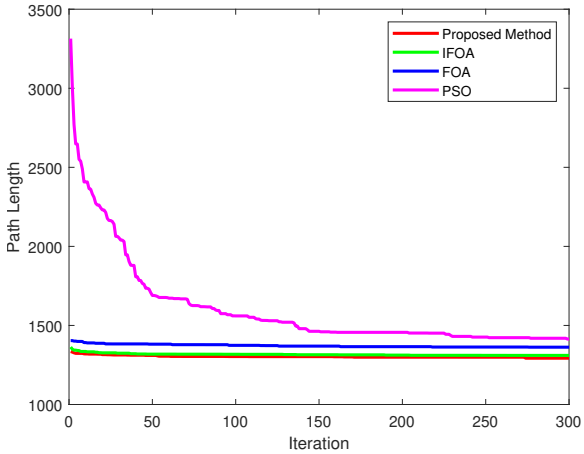

Fig. 10 Path length comparison of the four algorithms

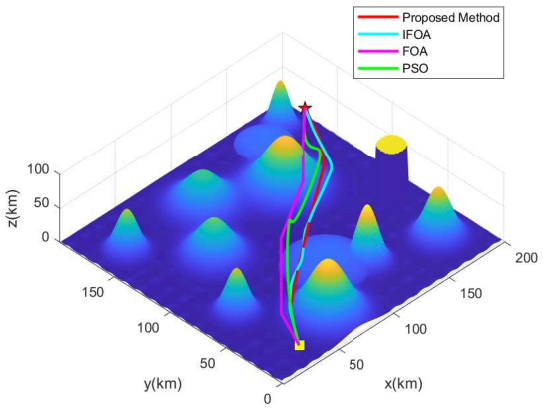

Fig. 11 The single path generated by the four algorithms in 3D view

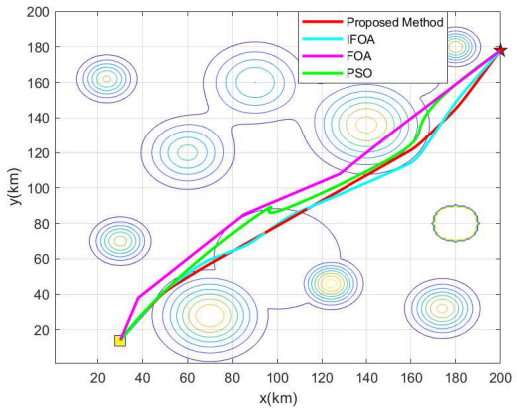

Fig. 12 The single path generated by the four algorithms in 2D view

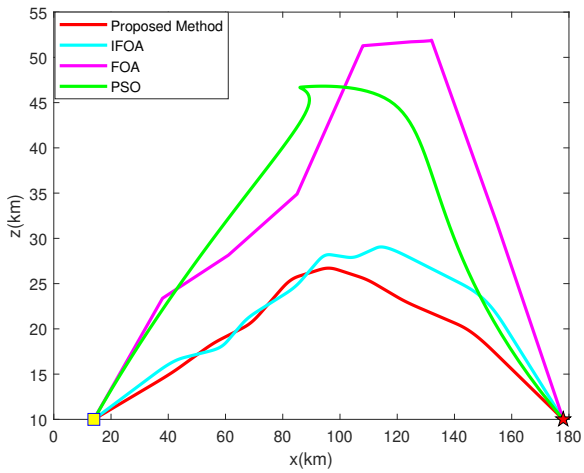

Fig. 13 The altitude of the single path generated by the four algorithms

5 Conclusions

This paper proposes a TSS-based NIFOA algorithm for 3D-space multi-UAV cooperative path planning. The TSS model is introduced to resolve the time-space coupling among multiple UAVs, and the multi-objective problem is transformed to a multi-constrain problem. In the proposed algorithm, to enhance the search efficiency of the algorithm, some novel strategies are integrated. Specifically, the greedy and restart strategies are used to speed up the convergence and avoid falling into local optima, and the evolution strategy of the optimum population is used to exchange the information of sub-populations to improve the convergence accuracy of the algorithm. The effectiveness of NIFOA has been proven by detailed comparisons with the other three optimization algorithms, namely PSO, FOA, and IFOA. The main idea of the proposed algorithm can be extended for dynamic and/or real-time path planning of multiple UAVs, which will be addressed in our future work.

Declarations

- Funding The work was partially supported by the National Natural Science Foundation of China under Grants U1934221, 61773323, and the Sichuan Science and Technology Program under Grants 2021JDJQ0012 and 2020YFQ0057.
- Conflict of interest The authors declare that they have no conflict of interest.
- Ethics approval This article does not contain any studies with human participants or animals performed by any of the authors.
- Consent to participate Not applicable.
- Consent for publication Not applicable.
- Availability of data and materials Not applicable.
- Authors' contributions All authors contributed to the study conception and design.

References

Alejo, D., Cobano, J., Heredia, G., Ollero, A., 2013. Particle swarm optimization for collision-free 4D trajectory planning in unmanned aerial vehicles, in: 2013 international conference on

unmanned aircraft systems (ICUAS), IEEE. pp. 298–307.

Bai, W., Wu, X., Xie, Y., Wang, Y., Zhao, H., Chen, K., Li, Y., Hao, Y., 2018. A cooperative route planning method for multi-UAVs based-on the fusion of artificial potential field and B-Spline interpolation, in: 2018 37th Chinese Control Conference (CCC), IEEE. pp. 6733–6738.

Bai, Y., Luo, M., Pang, F., 2021. An algorithm for solving robot inverse kinematics based on FOA optimized BP neural network. Applied Sciences 11, 7129.

Belew, R.K., McInerney, J., Schraudolph, N.N., 1990. Evolving networks: Using the genetic algorithm with connectionist learning, in: In, Citeseer. pp. 1–47.

Candeloro, M., Lekkas, A.M., Sørensen, A.J., 2017. A Voronoi-diagram-based dynamic path-planning system for underactuated marine vessels. Control Engineering Practice 61, 41–54.

Dakulović, M., Horvatić, S., Petrović, I., 2011. Complete coverage D* algorithm for path planning of a floor-cleaning mobile robot. IFAC Proceedings Volumes 44, 5950–5955.

Elbanhawi, M., Simic, M., Jazar, R.N., 2015. Continuous path smoothing for car-like robots using B-Spline curves. Journal of Intelligent & Robotic Systems 80, 23–56.

Flores, D.A., Saito, C., Paredes, J.A., Trujillano, F., 2017. Aerial photography for 3D reconstruction in the peruvian highlands through a fixed-wing UAV system, in: 2017 IEEE International Conference on Mechatronics (ICM), IEEE. pp. 388–392.

Ge, F., Li, K., Han, Y., Xu, W., et al., 2020. Path planning of UAV for oilfield inspections in a three-dimensional dynamic environment with moving obstacles based on an improved pigeon-inspired optimization algorithm. Applied Intelligence , 1–18.

Gu, Q., Chang, Y., Li, X., Chang, Z., Feng, Z., 2021. A novel F-SVM based on FOA for improving SVM performance. Expert Systems with Applications 165, 1–11.

Han, W., Shen, X., Hou, E., Xu, J., 2016. Precision time synchronization control method for smart grid based on wolf colony algorithm. International Journal of Electrical Power & Energy Systems 78, 816–822.

He, W., Qi, X., Liu, L., 2021. A novel hybrid particle swarm optimization for multi-UAV cooperate path planning. Applied Intelligence , 1–15.

Iscan, H., Gunduz, M., 2017. An application of fruit fly optimization algorithm for traveling salesman problem. Procedia computer science 111, 58–63.

Khan, P.W., Xu, G., Latif, M.A., Abbas, K., Yasin, A., 2019. UAV's agricultural image segmentation predicated by clifford geometric algebra. Ieee Access 7, 38442–38450.

Kumar, P.B., Rawat, H., Parhi, D.R., 2019. Path planning of humanoids based on artificial potential field method in unknown environments. Expert Systems 36, 1–12.

Lai, X., Li, J., Chambers, J., 2021. Enhanced center constraint weighted A* algorithm for path planning of petrochemical inspection robot. Journal of Intelligent & Robotic Systems 102, 1–15.

Li, B., Gong, L.g., Yang, W.l., 2014. An improved artificial bee colony algorithm based on balance-evolution strategy for unmanned combat aerial vehicle path planning. The Scientific World Journal 2014, 1–10.

Li, J., Xiong, Y., She, J., 2021a. An improved ant colony optimization for path planning with multiple UAVs, in: 2021 IEEE International Conference on Mechatronics (ICM), IEEE. pp. 1–5.

Li, K., Ge, F., Han, Y., Wang, Y., Xu, W., 2020. Path planning of multiple UAVs with online changing tasks by an ORPFOA algorithm. Engineering Applications of Artificial Intelligence 94, 1–15.

Li, M., 2014. Three-dimensional path planning of robots in virtual situations based on an improved fruit fly optimization algorithm. Advances in Mechanical Engineering 6, 1–12.

Li, S., Jiang, Z., 2019. Prediction of COP value of air-conditioning chillers based on improved FOA-GRNN, in: 2019 Chinese Control And Decision Conference (CCDC), IEEE. pp. 1343–1347.

Li, W., Zhen, Z., 2021. Intelligent surveillance and reconnaissance mode of police UAV based on grid, in: 2021 7th International Symposium on Mechatronics and Industrial Informatics (ISMII), IEEE. pp. 292–295.

Li, Y., Ming, Y., Zhang, Z., Yan, W., Wang, K., 2021b. An adaptive ant colony algorithm for autonomous vehicles global path planning, in: 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), IEEE. pp. 1117–1122.

Li, Y., Yang, Z., Li, X., Jiao, D., Jiang, W., 2021c. Multi-group fault positioning model of modular multilevel converter based on FOA-LSSVM, in: 2021 6th Asia Conference on Power and Electrical Engineering (ACPEE), IEEE. pp. 1423–1427.

Liu, C., Yan, X., Liu, C., Wu, H., 2011. The wolf colony algorithm and its application. Chinese Journal of Electronics 20, 212–216.

Luo, G., Yu, J., Mei, Y., Zhang, S., 2015. UAV path planning in mixed-obstacle environment via artificial potential field method improved by additional control force. Asian Journal of Control 17, 1600–1610.

Luo, S., Sarabandi, K., Tong, L., Pierce, L., 2019. A SAR image classification algorithm based on multi-feature polarimetric parameters using FOA and LS-SVM. IEEE Access 7, 175259–175276.

Mandloi, D., Arya, R., Verma, A.K., 2021. Unmanned aerial vehicle path planning based on A* algorithm and its variants in 3D environment. International Journal of System Assurance Engineering and Management 12, 990–1000.

Miao, C., Chen, G., Yan, C., Wu, Y., 2021. Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm. Computers & Industrial Engineering 156, 1–10.

Nazir, H., Bajwa, I.S., Samiullah, M., Anwar, W., Moosa, M., 2021. Robust secure color image watermarking using 4D hyperchaotic system, DWT, HbD, and SVD based on improved FOA algorithm. Security and Communication Networks 2021, 69–74.

Nex, F., Remondino, F., 2014. UAV for 3D mapping applications: A review. Applied Geomatics 6, 1–15.

Niu, H., Savvaris, A., Tsourdos, A., Ji, Z., 2019. Voronoi-visibility roadmap-based path planning algorithm for unmanned surface vehicles. The Journal of Navigation 72, 850–874.

Pan, W.T., 2012. A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowledge-Based Systems 26, 69–74.

Shanmugavel, M., Tsourdos, A., White, B., Żbikowski, R., 2010. Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs. Control engineering practice 18, 1084–1092.

Shao, Z., Yan, F., Zhou, Z., Zhu, X., 2019. Path planning for multi-UAV formation rendezvous based on distributed cooperative particle swarm optimization. Applied Sciences 9, 2621.

Shi, K., Zhang, X., Xia, S., 2020. Multiple swarm fruit fly optimization algorithm based path planning method for multi-UAVs. Applied Sciences 10, 1–21.

Tian, X., Li, J., 2019. A novel improved fruit fly optimization algorithm for aerodynamic shape design optimization. Knowledge-Based Systems 179, 77–91.

Wang, X., Zhou, X., Wang, H., 2017. Optimizing PI controller of the single-phase inverter based on FOA, in: 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), IEEE. pp. 151–155.

Xu, F., Li, H., Pun, C.M., Hu, H., Li, Y., Song, Y., Gao, H., 2020. A new global best guided artificial bee colony algorithm with application in robot path planning. Applied Soft Computing 88, 1–31.

Yao, Y.l., Liang, X.f., Li, M.z., Yu, K., Chen, Z., Ni, C.b., Teng, Y., 2021. Path planning method based on D* lite algorithm for unmanned surface vehicles in complex environments. China Ocean Engineering 35, 372–383.

Yuan, P., Zhou, Y., Wang, D., 2017. An improved fruit fly optimization algorithm incorporating average learning and step changing, in: 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC), IEEE. pp. 1516–1521.

Yuan, X., Dai, X., Zhao, J., He, Q., 2014. On a novel multi-swarm fruit fly optimization algorithm and its application. Applied Mathematics and Computation 233, 260–271.

Zhang, B., Duan, H., 2015. Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment. IEEE/ACM transactions on computational biology and bioinformatics 14, 97–107.

Zhang, D., Duan, H., 2018. Social-class pigeon-inspired optimization and time stamp segmentation for multi-UAV cooperative path planning. Neurocomputing 313, 229–246.

Zhang, X., Lu, X., Jia, S., Li, X., 2018. A novel phase angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism applied to UAV path planning. Applied Soft Computing 70, 371–388.

Zhu, D.D., Sun, J.Q., 2021. A new algorithm based on Dijkstra for vehicle path planning considering intersection attribute. IEEE Access 9, 19761–19775.