RHC Method Based 2D-equal-step Path Generation for UAV Swarm Online Cooperative Path Planning in Dynamic Mission Environment

Yue, Shen
School of Artificial Intelligence, University of Chinese Academy of Sciences; Institute of Automation, Chinese Academy of Sciences

Guoliang, Fan*
Institute of Automation, Chinese Academy of Sciences

ABSTRACT

This paper first mathematically models the UAV swarm online cooperative path planning problem based on the prerequisite assumptions of transparent posture and dynamic mission environment. Then the receding horizon control (RHC) and 2D-equal-step path generation method are briefly introduced and combined with the improved firefly optimization algorithm to solve the UAV swarm online cooperative path planning problem modeled in the previous. Simulations show that the improved firefly algorithm combining the RHC and 2D-equal-step path generation methods can be used to optimally solve the UAV swarm online cooperative path planning problem for moving mission targets in dynamic environments, and the improved firefly algorithm is more powerful and more efficient than the original algorithm in this process of application.

CCS CONCEPTS

• **Theory of computation**; • **Theory and algorithms for application domains**; • **Theory of randomized search heuristics**;

KEYWORDS

FA, Improved FA, RHC, 2D-equal-step Path Generation

ACM Reference Format:

Yue, Shen and Guoliang, Fan. 2023. RHC Method Based 2D-equal-step Path Generation for UAV Swarm Online Cooperative Path Planning in Dynamic Mission Environment. In *2023 3rd International Conference on Artificial Intelligence, Automation and Algorithms (AI2A '23), July 21–23, 2023, Beijing, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3611450.3611451

1 INTRODUCTION

UAV path planning is an important component of UAV mission planning and is usually defined as the planning process that occurs after the UAV mission assignment process to determine a comprehensive assessment of the optimal path for the UAV from its current location to the target location of its assigned mission. At present, it is usually academically reduced to a mathematical planning problem of finding the optimal solution to a specific objective cost function (e.g., distance cost, time cost, threat cost, etc.) under various constraints (e.g., terrain constraints, no-fly zone constraints, weather constraints, obstacle constraints, flight control constraints, UAV performance constraints, etc.), which essentially belongs to the NP-hard problem.

In recent years, the study of UAV path planning has received more and more attention from researchers and gradually accumulated a large number of research results. For the existing mainstream UAV path planning algorithms, academics usually classify them into two categories: traditional algorithms and intelligent optimization algorithms (also known as heuristic algorithms). Among them, traditional algorithms can be further divided into mathematical planning-based algorithms (e.g., integer planning [1], nonlinear planning [2], and dynamic planning [3]), graph search-based algorithms (e.g., Voronoi graph method [4], A* algorithm [5], D* algorithm [6], and Dijkstra algorithm [7]), sampling-based algorithms (e.g., probability road map method (PRM) [8] and rapidly-exploring random tree algorithm (RRT) [9]) and artificial potential field-based algorithms [10]; meanwhile, intelligent optimization algorithms mainly include swarm intelligence algorithms [11]- [13], deep learning algorithms [14], reinforcement learning algorithms [15], etc.

At present, although there are more studies on UAV swarm cooperative path planning, they are studied based on the scenario of fixed mission targets, and there is a lack of studies on cooperative path planning aiming at moving mission targets. In this paper, we propose to conduct an experimental study on UAV swarm online cooperative path planning in the context of air warfare based on transparent posture assumptions and dynamic mission scenario modeling.

2 PROBLEM SETTING AND MODELING

2.1 Problem Setting

In the air warfare scenario of transparent posture, the central battlefield is set as a 20KM×20KM square bounded airspace, our UAV cluster and enemy UAV cluster are both 4 homogeneous UAVs (numbered as UAV0, UAV1, UAV2 and UAV3 respectively). In the previous stage, the UAV cluster has been assigned to carry out the strike mission against the enemy's UAVs with the same number by our UAVs. For this purpose, we need to plan the corresponding path for each UAV in the UAV cluster, and represent them as $\{O_0, P_{01}, P_{02}, ..., P_{0m}, T_0\}$, $\{O_1, P_{11}, P_{12}, ..., P_{1m}, T_1\}$, $\{O_2, P_{21}, P_{22}, ..., P_{2m}, T_2\}$, $\{O_3, P_{31}, P_{32}, ..., P_{3m}, T_3\}$ respectively,so that the coordinated paths implements the strike mission against the enemy target with the

*Corresponding author. Tel.: +86 10 82544727. E-mail address: guoliang.fan@ia.ac.cn

AI2A '23, July 21–23, 2023, Beijing, China

ACM ISBN 979-8-4007-0760-5/23/07.
https://doi.org/10.1145/3611450.3611451

Table 1: The initial position coordinates and heading angles of our UAVs and enemy UAVs

Code Name	Initial Position Coordinates (km,km)	Initial Heading Angles (degree)
Our UAV0	(-9.8,-9.8)	45
Our UAV1	(9.8,-9.8)	135
Our UAV2	(9.8,9.8)	225
Our UAV3	(-9.8,9.8)	315
Enemy UAV0	(4.0,4.0)	180
Enemy UAV1	(-4.0,4.0)	270
Enemy UAV2	(-4.0,-4.0)	0
Enemy UAV3	(4.0,-4.0)	90

minimum threat cost and the shortest distance cost while satisfying all constraints, where O_v is the origin point of the planning path, that is, the location of our UAV v (v = 0, 1, 2, 3) at the initial moment of planning, T_v is the end point of the planning path, i.e., the location of the enemy UAV v at the initial planning moment, $\{P_{v1}, P_{v2}, ..., P_{vm}\}$ is the set of planning path points of our UAV v. To simplify the research discussion, the following assumptions are further made:

1. Assume that all the aircrafts are flying at the same altitude, i.e., the battlefield environment is two-dimensional(2D).

2. Assume that all the aircrafts are flying at a constant speed, without considering the speed's change or adjustment, and set the constant to 200m/s.

3. Assume that the threat range of all threat sources (i.e., the 3 UAVs outside the enemy aircraft of the mission target) becomes circles centered on the coordinates of their position at the current moment and with their mounted missile range as the radius.

4. Assume that the performance of the missiles mounted on our UAVs is superior to that of enemy UAVs, which allows us to ensure priority strikes. It is further specified that the missile range of our UAVs is 3 km and and the missile range of enemy UAVs is 2 km.

5. Assume that the enemy UAVs' maneuvering strategy is fixed area patrol, and they change course every 50s.

6. All aircraft are considered as mass points in the map.

2.2 Environment Modeling

Using a 20KM×20KM bounded airspace (the center as the origin (0,0)) to construct a two-dimensional coordinate system. At the initial moment (t=0), as shown in Table 1, we set the initial positions coordinates of UAVs and their heading angles (counterclockwise rotation angle in the positive direction of the x-axis of the coordinate system) be:

Any time after the t=0 moment, let our UAVs' heading angle be $\varphi_v{}^t$ (v = 0,1,2,3),and their position coordinates are $(x_v,y_v)^t$.

Figure 1 shows the air warfare posture at the initial moment. The symbols and lines are explained by legends on the right side of the figure.

2.3 Constraints

The UAV collaborative path planning constraints usually include three broad categories of self-relative constraints, mission environment-related constraints and collaborative-related constraints.

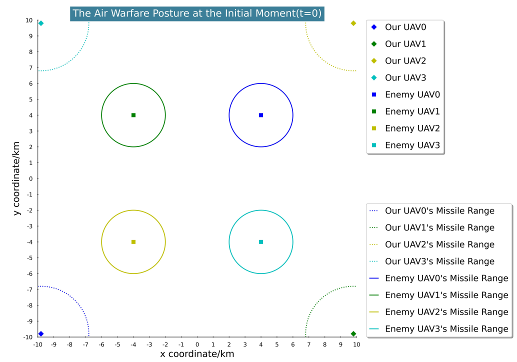

Figure 1: The air warfare posture at the initial moment.

2.3.1 Self-related constraints. Self-related constraints mainly include yaw angle constraint, climb/dive angle constraint, flight speed constraint, flight altitude constraint, minimum path segment constraint, maximum distance constraint(also be expressed as fuel constraint), etc. These constraints are merely related to the UAV's own flight control and platform parameters, and can also be regarded as the hardware constraints. Specifically, the maximum yaw angle constraint and the minimum path segment constraint are the most relevant constraint for the problem set in this paper.

1. Maximum yaw angle constraint

For v= 0, 1, 2, 3 and i= 1, 2, ..., m, define the heading angle of UAV_v on $P_{i-1}P_i$ path segment (P_0 is viewed as the origin point O_v of the path planning) as $\varphi_{v(i)}$ and the heading angle of it on P_iP_{i+1} path segment (P_{m+1} is viewed as the end point T_v of the path planning) as $\varphi_{v(i+1)}$. Then the yaw angle of UAV_v at point P_i could defined as $\triangle\varphi_{v(i)}$, so there are inequality constraint as below:

$$\Delta\varphi_{v(i)} = \left|\varphi_{v(i+1)} - \varphi_{v(i)}\right| \leq \Delta\varphi_{max} \;\; \forall v = 0, 1, 2, 3;\; \forall i = 1, 2, ..., m \tag{1}$$

In this research, $\triangle\varphi_{max}$ will be set as 30 degrees.

2. Minimum path segment constraint

For v= 0, 1, 2, 3 and i= 1, 2, ..., m+1, define the path length of UAV_v on $P_{i-1}P_i$ path segment (as mentioned above, P_0 represents the planning origin point O_v and P_{m+1} represents the planning ending point T_v) as $S_{v(i)}$, then there is the constraint as below:

$$S_{v(i)} \geq S_{min} \;\; \forall v = 0, 1, 2, 3;\; \forall i = 1, 2, ..., m + 1 \tag{2}$$

In this research, S_{min} will be set as 500 meters.

2.3.2 Mission environment-related constraints. The mission environment-related constraints mainly include mission boundary constraints, terrain constraints, no-fly zone constraints, obstacle constraints, electronic interference constraints, threat source constraints, etc. Since the problem context assumed in this paper is free high altitude confrontation, only the mission boundary constraint and threat source constraint are considered.

1. Mission boundary constraint

For v= 0, 1, 2, 3 and i= 1, 2, ..., m, define the planning path points P_i of UAV_v with the position coordinates ($x_{v(i)}$,$y_{v(i)}$), so we get mission boundary constraint:

$$\begin{gathered} -10000 \le x_{v(i)}, y_{v(i)} \le 10000 \\ \forall v = 0, 1, 2, 3;\ \forall i = 1, 2, ..., \mathrm{m} \end{gathered} \tag{3}$$

2. Threat source constraint

For v= 0, 1, 2, 3 and i= 1, 2, ..., m+1, assume P_{vij} is one point on $P_{i-1}P_i$ segment of the UAV_v's planning path which be the closest point to the threat source center T_j (again, P_0 is the planning origin point O_v and P_{m+1} is the planning ending point T_v), then the minimum distance d_{vij} between P_{vij} and T_j can be calculated by the following equation:

$$d_{vij} = \begin{cases} \left|\overrightarrow{P_{v(i-1)}T_j}\right| & if \quad \frac{\overrightarrow{P_{v(i-1)}T_j}\cdot\overrightarrow{P_{v(i-1)}P_{v(i)}}}{\left|\overrightarrow{P_{v(i-1)}P_{v(i)}}\right|^2} \le 0 \\ \left|\overrightarrow{P_{v(i)}T_j}\right| & if \quad \frac{\overrightarrow{P_{v(i-1)}T_j}\cdot\overrightarrow{P_{v(i-1)}P_{v(i)}}}{\left|\overrightarrow{P_{v(i-1)}P_{v(i)}}\right|^2} \ge 1 \\ \left|\overrightarrow{P_{v(i)x}T_j}\right| & if \quad 0 < \frac{\overrightarrow{P_{v(i-1)}T_j}\cdot\overrightarrow{P_{v(i-1)}P_{v(i)}}}{\left|\overrightarrow{P_{v(i-1)}P_{v(i)}}\right|^2} < 1 \end{cases} \tag{4}$$
$$\forall v = 0, 1, 2, 3;\ \forall i = 1, 2, ..., \mathrm{m};\ j \in (0, 1, 2, 3)\ and\ j \neq v$$

Where $P_{v(i)x}$ is the vertical foot point from the threat source center T_j to the path segment $P_{i-1}P_i$ of UAV_v.

Thus, the threat source constraint is obtained as follows:

$$\begin{gathered} d_{vij} > r_j \quad \forall v = 0, 1, 2, 3; \\ \forall i = 1, 2, ..., \mathrm{m};\ j \in (0, 1, 2, 3)\ and\ j \neq v \end{gathered} \tag{5}$$

Further, r_j is the threat radius of threat source T_j, which corresponds to the enemy UAV_j' missile range and is uniformly 3km in this paper.

2.3.3 Collaborative-related constraints. The collaborative-related constraints mainly include spatial collaborative constraints and temporal collaborative constraints, which are used to regulate the consistency of time and space in the process of the UAV swarm's collaborative mission execution. Since the hypothetical problem in this paper does not have the requirement and necessity of synergy for the completion time of each aircraft's strike mission, only the spatial collaborative constraint is considered.

For v,u= 0, 1, 2, 3(v≠u) and i= 1, 2, ..., m, any two of our UAVs UAV_v and UAV_u are required to maintain a minimum safe distance d_{min} at all times during their flight and without collision, then there are constraint:

$$\begin{gathered} d\left(P_{v(i)}, P_{u(i)}\right) \ge d_{min} \quad \forall v = 0, 1, 2, 3; \\ \forall i = 1, 2, ..., \mathrm{m};\ \forall u = 0, 1, 2, 3 \text{ not } v \end{gathered} \tag{6}$$

Where P_{vi} and P_{ui} are the i-th planning pathpoints of UAV_v and UAV_u respectively, and $d(P_{v(i)}, P_{u(i)})$ represents their spacing.

2.4 Objective Cost Function

The design of the objective cost function for the UAV swarm online path planning problem usually depends on the specific mission requirements and the selection preferences of the command decision maker,even though, in academic, the most commonly cost functions are distance cost, time cost, path threat cost and path feasibility cost.

Regardless of the choice of one or more objective costs, and regardless of how exactly the objective cost function is designed, the objective cost function basically boils down to the following general expression:

$$\min F = \sum_{i}^{n} \omega_i J_i \tag{7}$$

Where, F denotes the composite cost, i.e., the objective function of the optimization problem; J_i denotes a specific cost, ω_i is its weight coefficient, and there is:

$$\sum_{i}^{n} \omega_i = 1 \tag{8}$$

If the optimization is not weighted by subcosts but is split separately, it needs to be solved optimally as a multi-objective optimization problem.

In this paper, the objective cost function is designed as following:

$$\begin{aligned} \min F = 0.2 \times \sum_{v=0}^{3}\sum_{i=1}^{m} F_{\varphi_{vi}} + 0.2 \times \sum_{v=0}^{3}\sum_{i=1}^{m+1} F_{S_{vi}} \\ +0.3 \times \sum_{v=0}^{3}\sum_{i=1}^{m+1}\sum_{j=0;j\neq v}^{3} F_{Threat_{vij}} + 0.3 \times \sum_{v=0}^{3}\sum_{u=0,u\neq v}^{3}\sum_{i=1}^{m} F_{d_{vui}} \end{aligned} \tag{9}$$

Of which, $F\varphi_{vi}$ denotes the yaw angle cost of UAV_v at planning path point P_i and satisfies:

$$F_{\varphi_{vi}} = \begin{cases} \Delta\varphi_{vi} & \text{if } \Delta\varphi_{vi} \le \Delta\varphi_{max} \\ \infty & \text{if } \Delta\varphi_{vi} > \Delta\varphi_{max} \end{cases} \tag{10}$$

(read maximum yaw angle constraint for reference)

F_{Svi} represents the path length cost of UAV_v on $P_{i-1}P_i$ path segment and satisfies:

$$F_{S_{vi}} = \begin{cases} S_{vi} & \text{if } S_{vi} \ge S_{min} \\ \infty & \text{if } S_{vi} < S_{min} \end{cases} \tag{11}$$

(read minimum path segment constraint for reference)

$F_{Threatvij}$ means the threat cost of UAV_v on $P_{i-1}P_i$ path segment subject to the threat source T_j and satisfies:

$$F_{Threat_{vij}} = \begin{cases} r_j/d_{vij} & \text{if } d_{vij} > r_j \\ \infty & \text{if } d_{vij} \le r_j \end{cases} \tag{12}$$

(read threat source constraint for reference)

F_{dvui} is the spatial collaborative cost of any two UAVs v and u at the i-th planning path point P_i and satisfies:

$$F_{d_{vui}} = \begin{cases} d_{min}/d\left(\mathrm{P}_{v(i)}, P_{u(i)}\right) & \text{if } d\left(\mathrm{P}_{v(i)}, P_{u(i)}\right) \ge d_{min} \\ \infty & \text{if } d\left(\mathrm{P}_{v(i)}, P_{u(i)}\right) < d_{min} \end{cases} \tag{13}$$

(read spatial collaborative constraint for reference)

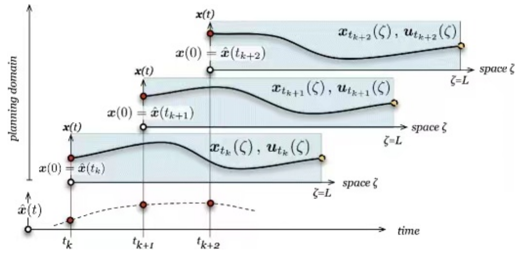

Figure 2: Schematic diagram of receding horizon control. (https://link.springer.com/article/10.1007/s12544-014-0140-6)

3 RECEDING HORIZON CONTROL METHOD

Since the mission targets in the modeled UAV swarm online path planning problem are all dynamic moving platforms, the previous offline path planning algorithms are no longer applicable to it. At the same time, the amount of computation required for each path planning is very large, and how to maintain the real-time planning according to the changing air combat posture during the execution of task becomes the key problem and core difficulty of UAV swarm online path planning in dynamic mission environment.

To this end, this paper intends to make an attempt to solve the problem based on the receding horizon control method.

Receding horizon control (RHC), also known as model predictive control (MPC), proposed by Richalet J and Rault A et al [16], is a modern control theory developed and improved in the late 1970s, mainly aim at the uncertainty caused by model mismatch, distortion, perturbation or other reasons, it is a method to split a large-scale complex global optimization problem into a series of small-scale simple local optimization problems in a time-rolling manner, reflecting the idea of "simplifying the complexity". At present, it has been applied academically to the problem of UAV path planning, reference the job in [17].

Specifically, for the application of RHC in UAV swarm online path planning, as shown in Figure 2 By selecting a fixed prediction domain first, the optimal path is predicted in this domain, meanwhile, set a determined control time domain $\triangle$t, in this control time domain the UAV will not need to re-plan the path but fly according to the predicted optimal path in the prediction domain. Once the control time domain is exceeded at the next moment, the predicted domain and the control time domain for the next phase are redefined, and so on, until the UAV reaches the location of the mission target.

The disassembly process steps are as follows:

- Assume current time is t_k, based on the current UAVs' location $O(t_k)$ and mission tasks' location $T(t_k)$, the optimal $Path(t_k)$ of the UAVs from $O(t_k)$ to $T(t_k)$ is solved optimally in a finite time domain [t,t+x], x is an unknown variable, and we don't care it's value.
- Select the previous part of the trajectory represented by the time domain $[t_k : t_{k+1}]$ in the $Path(t_k)$ as the reference flight path of the UAVs, during the period we need not to re-plan the path.
- At the moment t_{k+1}, the new optimal $Path(t_{k+1})$ of the UAVs from the current location $O(t_{k+1})$ to the mission tasks' current location $T(t_{k+1})$ is solved optimally in a finite time domain $[t_{k+1},t_{k+1}+y]$, also, y is an unknown variable.
- Starting from the time t_{k+2}, repeat the above steps until triggering the termination condition of iteration. Here, is the situation when all the enemy UAVs arrived within the missile range of our corresponding UAVs.

Although the introduction of RHC into online path planning can solve the problem that the original planning path cannot be applied to the new mission environment due to the dynamic change of the mission environment, and ensure the real-time performance of online path planning. However, since RHC is a continuous local optimization process with time-rolling manner, which is similar to the greedy algorithm, the final output trajectory cannot be guaranteed to be globally optimal.

4 IMPROVED FIREFLY OPTIMIZATION ALGORITHM

The firefly optimization algorithm (FA) is a swarm intelligence algorithm proposed by Yang in 2009 and inspired by the courtship behavior of fireflies [18], which is based on the principle of using artificial fireflies to simulate an arbitrary solution vector in the solution space, using the brightness of fireflies to characterize the magnitude of the objective function value of the solution vector, using the attraction and movement behavior of the fireflies due to the difference in brightness to simulate the search optimization process of the algorithm.

4.1 Principle of firefly optimization algorithm

According to the algorithm, fireflies are attracted to each other regardless of their sex, i.e., all fireflies can be attracted to each other. The attraction of a firefly is related to its own brightness (i.e., the magnitude of the objective function of its location vector), and a brighter firefly will attract other fireflies around it that are not as bright. At the same time, the brightness of a firefly decays with the distance it travels. For any firefly, if there is no firefly brighter than it, it will move randomly.

Specifically, the firefly optimization algorithm process is as follows:

4.1.1 Random initialization of artificial firefly swarm. Suppose the dimension of the solution space of the optimization problem Y = f(X) to be solved is D. Further, let the size of the artificial firefly population be n. The initial moment k = 0 generates n fireflies in the solution space using random initialization, which are respectively X_1 to X_n. Then for any artificial firefly X_i, it can be expressed as $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$. For the objective function Y = f(X), let maximization be the optimization direction, the attraction relationship between fireflies will occur according to the brightness which is the magnitude of the objective function value, while assuming that the distance between different artificial fireflies is defined as the Cartesian distance between their location vectors.

4.1.2 Calculation and comparison of the fluorescence brightness and relative fluorescence brightness of fireflies themselves. For any artificial firefly X_i, calculate the objective function value of its

Firefly Algorithm

Objective function $f(\boldsymbol{x}), \quad \boldsymbol{x} = (x_1, ..., x_d)^T$
Generate initial population of fireflies $\boldsymbol{x}_i \ (i = 1, 2, ..., n)$
Light intensity I_i *at* $\boldsymbol{x}_i$ *is determined by* $f(\boldsymbol{x}_i)$
Define light absorption coefficient γ
while *(t <MaxGeneration)*
for $i = 1 : n$ *all n fireflies*
  for $j = 1 : n$ *all n fireflies (inner loop)*
  if $(I_i < I_j)$, *Move firefly i towards j;* **end if**
    Vary attractiveness with distance r via $\exp[-\gamma r]$
    Evaluate new solutions and update light intensity
  end for j
end for i
Rank the fireflies and find the current global best $\boldsymbol{g}_*$
end while
Postprocess results and visualization

Figure 3: Pseudocode of FA.

location vector $f(X_i)$, as its own fluorescence brightness I_{i0}. And calculate the relative fluorescence brightness of another firefly X_j relative to it by the following equation:

$$I_{ji} = I_{j0} \times e^{-\gamma \times r_{ij}} \tag{14}$$

where γ is the light intensity absorption coefficient, which is also the fluorescence attenuation coefficient; r_{ij} is the distance between firefly i and firefly j.

4.1.3 Firefly location update. If for firefly X_i, its own fluorescence brightness I_{i0} is greater than or equal to the relative fluorescence brightness of all the other (n-1) fireflies for it, then firefly X_i moves randomly in the current iteration round (k+1), and its location update equation is as below:

$$x_i^{k+1} = x_i^k + \lambda \times \alpha \tag{15}$$

where λ is a random number uniformly distributed in the interval [-1,1]; α is the step vector.

Otherwise, firefly X_i is attracted and moves closer to one of the (n-1) fireflies, X_j, for which it has the greatest relative fluorescence brightness for X_i, and X_i updates its value by the following equation:

$$x_i^{k+1} = x_i^k + \beta_{ji} \times \left(x_j^k - x_i^k \right) \tag{16}$$

Where, β_{ji} is the relative attraction of firefly X_j to firefly X_i, which determines the amplitude of the movement from X_i to X_j, and is calculated by this equation:

$$\beta_{ji} = \beta_0 \times e^{-\gamma \times r_{ij}^2} \tag{17}$$

β_0 is the maximum attractiveness and is generally set to 1.

4.1.4 Loop iteration. Repeat steps 4.1.2 and 4.1.3 until the algorithm iterations reach the maximum number of iterations T_{max} or the setted search precision, and output the location of the firefly with the largest fluorescence brightness of itself, which is the optimal solution of the algorithm iteration.

In summary, the firefly optimization algorithm pseudocode is as Figure 3

4.2 Improvements to the firefly optimization algorithm

The firefly optimization algorithm is simple in principle and has a low computational complexity of $O(n^2)$. However, there are two drawbacks about it, which are relatively easy to fall into local optimum and insufficient local search capability, for which the following improvements are proposed in this paper.

4.2.1 The introduction of the subgroup metric factor d_{near} and the subgroup maximum proportionality factor C.. After each round of location update, for each firefly X_i, calculate the distance r_{ij} between any other firefly X_j and it and count the times m when r_{ij} is less than the subgroup metric factor d_{near}. If m > n*C, where C is the subgroup maximum proportionality factor, then it indicates the existence of a subpopulation of fireflies with very compact distance and number exceeding the threshold, reflecting the possibility that the algorithm iterates into a local optimum. At which time, firefly X_i will be reinitialized with other m fireflies, forcing the subpopulation to be dissolved and then continue iteration.

4.2.2 Orientation perturbation for location update. In formula (16), since β_{ji} is a constant, thus x_i^{k+1} is updated from x_i to x_j. Although this location update formula can make the firefly swarm quickly gather to the brighter individuals and accelerate the convergence speed of the algorithm, while it is too deterministic and loses the stochastic exploration that the swarm intelligence algorithm should have, and does not match the real movement trajectory of fireflies in nature. For this reason, this paper proposes to introduce a random directional perturbation to the location update formula, so as to enhance the diversity and local search range of the algorithm.

The improved location update formula is as follows:

$$x_i^{k+1} = x_i^k + \beta_{ji} \times S^k \times \left(x_j^k - x_i^k\right) \tag{18}$$

Where S_k is the introduced random perturbation factor, and each of its d-dimensional components $S_d{}^k$ is random number uniformly distributed in the interval [0,1].

With the introduction of the location update random perturbation factor, the enhancement of individual search range allows the firefly swarm to cover more areas in the search space during movement.

5 2D-EQUAL-STEP PATH GENERATION METHOD

The 2D-equal-step path generation method was proposed by Zhu W et al. in 2013 [19] and is used in combination with the chaotic biological predator algorithm to optimally solve the 2D spatial UAV path planning problem presented in that paper, and the final result was satisfactory. Inspired by the method, this paper intends to combine it with the receding horizon control method and the improved FA for an attempted solution of UAV swarm online path planning for moving mission targets in dynamic mission environment.

The following describes the step-by-step ideas of the method:

1. Determine the planning origin point O (UAV's current location) and the end point T(moving mission's current location) for two-dimensional spatial path planning, connect OT and equate OT into (m+1) segments by m equidistant interval points, and the length of each segment, i.e., the planning step length d= |OT| / (m+1).

2. At each equidistant interval point make m lines L_1, L_2, ..., L_k, ..., L_m, perpendicular to OT, respectively, using (x_1,y_1), (x_2,y_2), ...,(x_k,y_k), ..., (x_m,y_m) denote any point on these m lines, the 2D-equal-step path can be expressed as a sequential connection of the following discrete points:

$$Path = \begin{Bmatrix} O \to (x_1, y_1) \to (x_2, y_2) \to \\ \ldots \to (x_k, y_k) \to \ldots \to (x_m, y_m) \to T \end{Bmatrix} \tag{19}$$

3. Coordinate transformation of the 2D space, the transformed coordinate system takes O as the origin and OT as the horizontal axis, as shown in Figure 4. The conversion equation is as follows as:

$$\begin{bmatrix} x'_k \\ y'_k \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_k - x_o \\ y_k - y_o \end{bmatrix} \tag{20}$$

where (x_o,y_o) is the coordinate of the planning origin point O in original coordinate system, (x_k,y_k) is the coordinate of any point on L_k in original coordinate system; (x'_k,y'_k) is the new coordinate after coordinate transformation of (x_k,y_k); θ is the angle between the x-axis of the original coordinate system and the x-axis(OT) of the new coordinate system, which can be calculated by the following equation:

$$\theta = \arctan\left(\frac{y_t - y_o}{x_t - x_o}\right) \tag{21}$$

where (x_t,y_t) is the original coordinate of the task endpoint T.

4. Since the horizontal coordinates of the path points are all known to be fixed in the new coordinate system, the path points' coordinates that need to be solved optimally after coordinate conversion are their vertical coordinates merely, i.e. $\{y'_1, y'_2, ..., y'_m\}$, which greatly reduces the computation complexity.

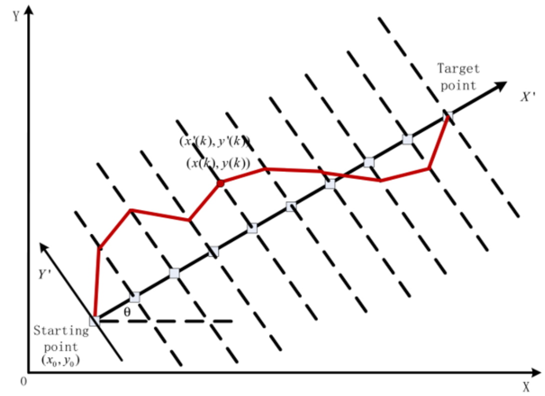

Figure 4: Coordinate system conversion of 2D-equal-step path generation. [19]

At this stage, the selection of traditional path planning algorithms is very dependent on the digital map modeled by the path planning problem. For example: the application of probabilistic sampling algorithms such as Probabilistic Road Marking (PRM) algorithm and Rapid Expansion Random Tree (RRT) algorithm require modeling based on probabilistic maps; the application of direct search algorithms such as Dijkstra algorithm, A* algorithm and D* algorithm requires modeling based on static road network maps (among them the rasterized digital maps is the most widely used); the application of Voronoi diagram algorithm and artificial potential field algorithm are based on their respective specific modeling maps. While as for the intelligent optimization algorithms, especially the swarm intelligence algorithms, the solution to the path planning problem is mainly based on the optimization of the objective cost function with constraints, which is different from the traditional path planning algorithms to a large extent. In layman's terms, this means that the application of swarm intelligence algorithms for UAV path planning has no additional requirements on the modeling of digital maps, which makes it possible to have the advantages of fast real-time and easy computation required for online path planning problems.

Meantime, the 2D-equal-step path generation method introduced in previous can further enhance the computational advantages of the swarm intelligence algorithm applied to the UAV swarm online path planning problem, which is mainly achieved by dimensionality reduction of the mapping from the original 2D solution space to the 1D solution space. Thus, by combining the 2D-equal-step path generation method with the improved FA, the complexity of digital map modeling is reduced on the one hand, and the optimization difficulty of the original intelligent optimization algorithm is reduced on the other hand, thus ensuring the real-time performance and possibility of UAV swarm online path planning under the receding horizon control mechanism.

Table 2: The initial parameters of FA and improved FA

Parameters of FA	The Initial Value	Parameters of Improved FA	The Initial Value
n: Artificial Firefly Swarm Size	100	n: Artificial Firefly Swarm Size	100
k: Initial Iteration Round	0	k: Initial Iteration Round	0
D: The Dimension of Solution Space[1]	4	D: The Dimension of Solution Space[1]	4
γ: Fluorescence Attenuation Coefficient	1	γ: Fluorescence Attenuation Coefficient	1
β_0: Maximum Attractiveness Coefficient	1	β_0: Maximum Attractiveness Coefficient	1
t_{max}: Maximum Iterations	100	t_{max}: Maximum Iterations	100
		C: Subgroup Maximum Proportionality Factor	0.2
		d_{near}: Subgroup Metric Factor	10

[1] It is also the number of path points of planning path.

6 SIMULATION EXPERIMENT

In order to verify the research idea of "the improved firefly optimization algorithm combining with the receding horizon control method and 2D-eaual-step path generation method can solve the UAV swarm online path planning problem for moving mission targets in dynamic mission environment", and to compare the performance difference between the improved firefly optimization algorithm and the basic firefly optimization algorithm when applied to the path planning problem modeled in chapter two of this paper, series of experiments are conducted in a Python programming environment on a Linux Server with four 2.8 GHz CPU(P4 Xeon).

The initial parameters of both FA and improved FA are set as Table 2 display:

Figure 5 shows the planning paths of the basic FA combined with RHC and 2D-equal-step method from the initial posture (t= 0) to the terminal posture (t= 62), Figure 6 shows the planning paths of the improved FA combined with RHC and 2D-equal-step method from the initial posture (t= 0) to the terminal posture (t= 54).

We could find that, the improved FA have well planned the flight paths to avoid all the threat sources for UAVs, and the planning paths satisfy all the constraints proposed in chapter two obviously.While the FA did not plan well, its planning paths are less smoother than former and lack of robustness, also, which visibly violate the maximum yaw angle constraint and spatial collaborative constraint. Leave that aside, just judging by the final result, its planned flight paths also apparently failed, which can be seen in the last subgraph in Figure 5, our UAVs are all within enemy UAVs' missile range(t=62) before they complete their strike mission against their respective targets. By contrast, as shown in Figure 6, throughout the whole planning process, the UAVs' RHC paths planed by improved FA are always out of the enemies' missile range, which ensures priority strike and finally completed the strike mission against the moving UAV targets, and the whole process is highly collaborative, which are not too much to call it beautiful!

We could further find the difference in performance between the two methods from the Figure 7 below, which reveals that the improved FA is much faster than the basic FA in the convergence rate.

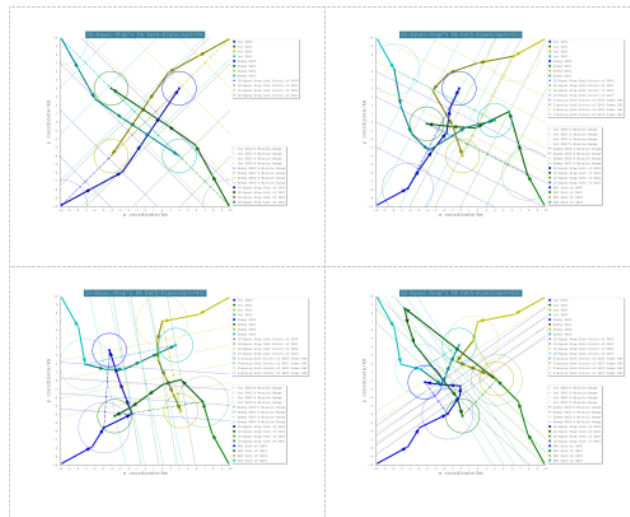

Figure 5: The air warfare posture from the initial moment to teminal moment for the FA.

7 CONCLUSIONS

In this paper,an algorithm based on the improved firefly optimization algorithm combined with receding horizon control method and 2D-equal-step path generation method was proposed for solving the UAV swarm online path planning problem aiming at moving targets in dynamic mission environment. The generated paths can ensure the maximum safety with the minimum fuel cost and steering cost of UAVs within the various constraints, which aslo maintain a high degree of collaboration in real-time throughout the planning process at the same time, as shown in the simulation results. Further, the comparative simulation between FA and improved FA also indicates that the latter is more powerful and efficient than the former in solving previous problem.

Our future work will focus on the extensive application of our proposed method in UAV swarm online path planning of 3D space, which is a challenging issues for next stage.

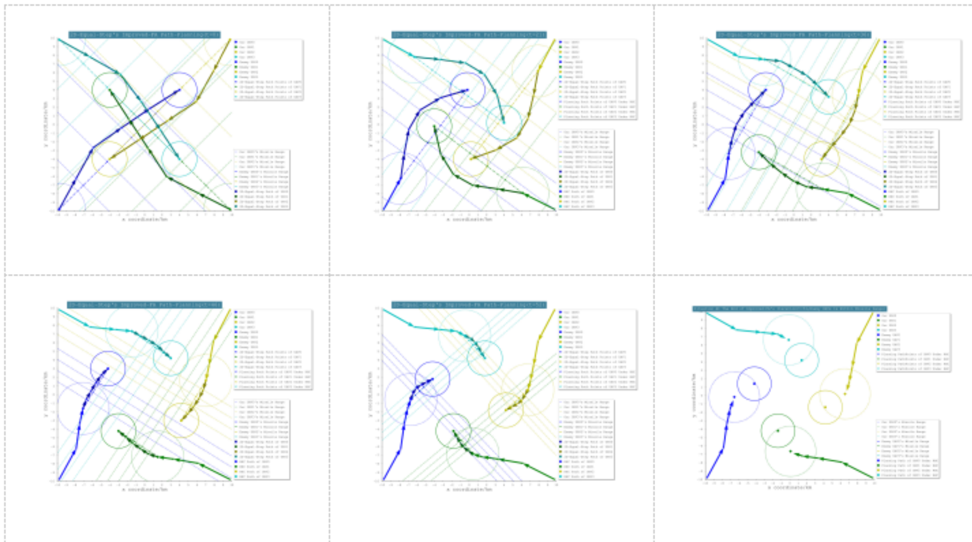

Figure 6: The air warfare posture from the initial moment to teminal moment for the improved FA.

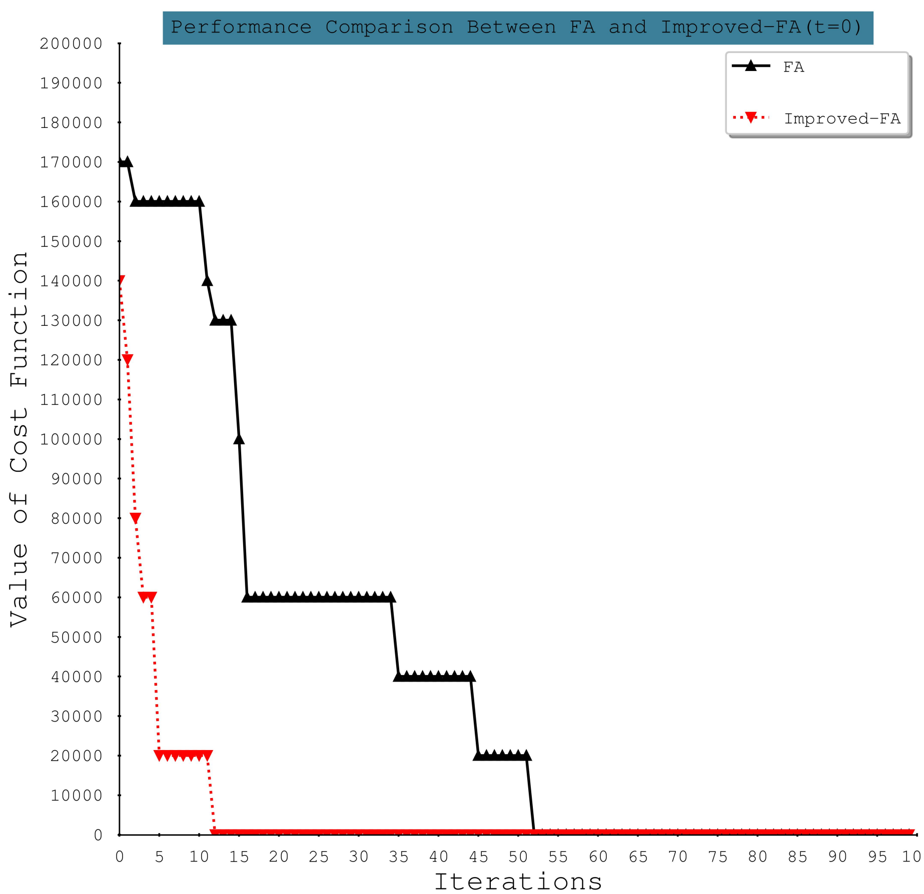

Figure 7: Comparison of optimization ability between FA and improved FA in the same solution space (t = 0).

ACKNOWLEDGMENTS

This work was supported by National Basic Scientific Research Program under grants #JCKY2019203C029 and #JCKY2019207B022, and National Foundation Fund under grant #Y2009-1B-02-353.

REFERENCES

[1] Schouwenaars, T., Moor, B.D., Feron, E., & How, J.P. (2001). Mixed integer programming for multi-vehicle path planning. 2001 European Control Conference (ECC), 2603-2608.
[2] Zhang, Z., Li, J., & Wang, J. (2018). Sequential convex programming for nonlinear optimal control problems in UAV path planning. Aerospace Science and Technology, 76, 280-290.
[3] Mokrane, A., Braham, A.C., & Cherki, B. (2020). UAV Path Planning Based on Dynamic Programming Algorithm On Photogrammetric DEMs. 2020 International Conference on Electrical Engineering (ICEE), 1-5.
[4] Pehlivanoglu, Y.V. (2012). A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV. Aerospace Science and Technology, 16, 47-55.
[5] Wang, H., Qi, X., Lou, S., Jing, J., He, H., & Liu, W. (2021). An Efficient and Robust Improved A* Algorithm for Path Planning. Symmetry, 13, 2213.
[6] Stentz, A. (1994). Optimal and efficient path planning for partially-known environments. Proceedings of the 1994 IEEE International Conference on Robotics and Automation, 3310-3317 vol.4.
[7] Karve, D., & Kapadia, F. (2020). Multi-UAV Path Planning using Modified Dijkstra's Algorithm. International Journal of Computer Applications, 175, 26-33.
[8] Yan, F., Liu, Y., & Xiao, J. (2013). Path Planning in Complex 3D Environments Using a Probabilistic Roadmap Method. International Journal of Automation and Computing, 10, 525-533.
[9] Kothari, M., Postlethwaite, I., & Gu, D. (2009). Multi-UAV path planning in obstacle rich environments using Rapidly-exploring Random Trees. Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, 3069-3074.
[10] Chen, X., & Zhang, J. (2013). The Three-Dimension Path Planning of UAV Based on Improved Artificial Potential Field in Dynamic Environment. 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics, 2, 144-147.
[11] Huang, C., & Fei, J. (2017). UAV Path Planning Based on Particle Swarm Optimization with Global Best Path Competition. Int. J. Pattern Recognit. Artif. Intell., 32, 1859008:1-1859008:23.
[12] Arantes, M.D., Arantes, J.D., Toledo, C.F., & Williams, B.C. (2016). A Hybrid Multi-Population Genetic Algorithm for UAV Path Planning. Proceedings of the Genetic and Evolutionary Computation Conference 2016.
[13] Konatowski, S., & Pawlowski, P. (2018). Ant colony optimization algorithm for UAV path planning. 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET), 177-182.
[14] Zhang, Z., Wang, S., Chen, J., & Han, Y. (2023). A Bionic Dynamic Path Planning Algorithm of the Micro UAV Based on the Fusion of Deep Neural Network Optimization/Filtering and Hawk-Eye Vision. IEEE Transactions on Systems, Man, and Cybernetics: Systems.
[15] Cui, Z., & Wang, Y. (2021). UAV Path Planning Based on Multi-Layer Reinforcement Learning Technique. IEEE Access, 9, 59486-59497.
[16] Richalet, J., Rault, A., Testud, J.L., & Papon, J. (1978). Model predictive heuristic control: Applications to industrial processes. Autom., 14, 413-428.
[17] Koohifar, F., Kumbhar, A., & Guvenc, I. (2017). Receding Horizon Multi-UAV Cooperative Tracking of Moving RF Source. IEEE Communications Letters, 21, 1433-1436.
[18] Yang, X. (2009). Firefly Algorithms for Multimodal Optimization. symposium on Stochastic Algorithms: Foundations and Applications.
[19] Zhu, W., & Duan, H. (2014). Chaotic predator-prey biogeography-based optimization approach for UCAV path planning. Aerospace Science and Technology, 32, 153-161.