An Improved WPA Based on RHC Method and 2D-equal-step Path Generation Method is Proposed for Solving the Single UAV Online Path Planning Problem in Dynamic Mission Environment

YUE, Shen
School of Artificial Intelligence, University of Chinese Academy of Sciences; Institute of Automation, Chinese Academy of Sciences; Email Address: shenyue2020@ia.ac.cn

GUOLIANG, FAN*
Institute of Automation, Chinese Academy of Sciences; Email Address: guoliang.fan@ia.ac.cn

ABSTRACT

Based on the premises of transparent posture and dynamic mission environment, the mathematical modeling of the single Unmanned Aerial Vehicle (UAV) online path planning problem is carried out. Then, Receding Horizon Control method (RHC) and 2D-equal-step path generation methods are briefly introduced, which are combined with the Improved Wolf Pack Algorithm (IWPA) to solve the single UAV online path planning problem modeled in the previous. The simulation results show that the new algorithm can be used to solve the single UAV online path planning problem aiming at moving targets in dynamic mission environment, and the performance of the improved wolf pack algorithm is more powerful than the original wolf pack algorithm in this process of application.

CCS CONCEPTS

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

KEYWORDS

WPA, Improved WPA, RHC, 2D-equal-step path generation

ACM Reference Format:
YUE, Shen and GUOLIANG, FAN. 2023. An Improved WPA Based on RHC Method and 2D-equal-step Path Generation Method is Proposed for Solving the Single UAV Online Path Planning Problem in Dynamic Mission Environment. In *2023 International Conference on Frontiers of Artificial Intelligence and Machine Learning (FAIML 2023), April 14–16, 2023, Beijing, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3616901.3616908

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 position to its assigned mission position. 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 optimization algorithms and intelligent optimization algorithms (also known as heuristic algorithms). Among them, traditional optimization 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 single UAV path planning, they are studied based on the scenario of fixed mission targets, and there is a lack of studies on path planning aiming at moving mission targets. In this paper, we propose to conduct an experimental study on single UAV online 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 envisioned environment of transparent posture air warfare, the central battlefield is set as a 20KM × 20KM square bounded airspace, our aircraft swarm are 4 homogeneous UAVs, and the enemy aircraft swarm are 1 manned-aircraft and 3 homogeneous UAVs (named as UAV_1, UAV_2 and UAV_3). In the previous stage, after the UAV swarm cooperative mission assignment process, it has been determined that one of our UAVs (named as UAV_0) will carry out the strike mission against the enemy manned-aircraft, for which it is necessary to plan the corresponding path, denoted as {S, P_1, P_2, ..., P_m, T}, so that it can complete the mission with the minimum threat cost and distance cost while satisfying the constraints at

*Corresponding author. Tel.: +86 10 82544727.

FAIML 2023, April 14–16, 2023, Beijing, China

ACM ISBN 979-8-4007-0754-4/23/04.
https://doi.org/10.1145/3616901.3616908

the same time. Where S is the starting point of the path planning, which refers to the position of UAV_0 at the initial planning moment, T is the end point of the path planning, which refers to the position of manned-aircraft at the initial planning moment, $\{P_1, P_2, ..., P_m\}$ is the set of planned path points. To simplify the research discussion, the following assumptions are further made:

- Assume that all the aircraft are flying at the same altitude, i.e., the battlefield environment is two-dimensional(2D).
- Assume that all the aircraft are flying at constant speed, not considering the speed's change or adjustment, we set the flight speed of the unmanned aircraft to 200m/s and the flight speed of manned-aircraft to 400m/s in this paper.
- Assume that the threat range of all threat sources (i.e., three enemy unmanned aircraft UAV_1, UAV_2 and UAV_3) becomes a circle centered on their position coordinates at the current moment and with a radius of their mounted missile range.
- Assume that the performance of the missiles mounted on UAV_0 is superior to that of enemy aircraft, which allows us to ensure priority strikes. It is further stipulated that the missile range of UAV_0 is 5 km, the missile range of enemy manned-aircraft is 4 km, and the missile range of enemy UAV is 3 km.
- Assume that the enemy aircraft's maneuvering strategy is irregular patrol, irregular change of course, and adjustment of course when encountering the border.
- All the 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), set UAV_0's initial track angle be 45 degrees (counterclockwise in the positive direction of the x-axis of the coordinate system) and all enemy aircraft's initial track angle be 225 degrees, set UAV_0's, manned-aircraft's, UAV_1's, UAV_2's and UAV_3's position coordinates be (-9000.0, -9000.0), (9000.0, 9000.0), (-7000.0, 7000.0), (0.0, 0.0) and (7000.0, -7000.0) respectively. Any second after initial moment (t > 0), define UAV_0's track angle and position coordinate as φ_t and $(x_0, y_0)_t$, define manned-aircraft's, UAV_1's, UAV_2's and UAV_3's position coordinates as $(x_1,y_1)_t$, $(x_2,y_2)_t$, $(x_3,y_3)_t$ and $(x_4,y_4)_t$. The enemy aircraft random changes its track angle every 10s and the change satisfies the maximum yaw angle constraint.

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.

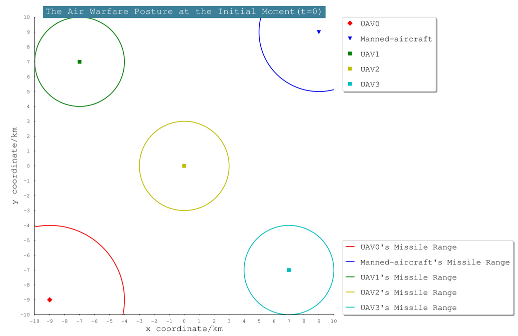

Figure 1: The air warfare posture at the initial moment.

2.3 Constraints

Single UAV online path planning constraints usually include two categories of self-related constraints and mission environment-related constraints, compared with UAV swarm cooperative online path planning, the multi-aircraft cooperation-related constraints are no need to consider.

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.

- Maximum yaw angle constraint

For i= 1, 2, ..., m, define the track angle of UAV_0 on $P_{i-1}P_i$ path segment (P_0 is viewed as the starting point S of the path planning) as φ_i and the track angle of UAV_0 on P_iP_{i+1} path segment (P_{m+1} is viewed as the end point T of the path planning) as φ_{i+1}. Then the yaw angle of UAV_0 at point P_i could defined as $\Delta\varphi_i$, so there are inequality constraint as below:

$$\Delta\varphi_i = |\varphi_{i+1} - \varphi_i| \leq \Delta\varphi_{\max} \quad \forall i = 1, 2, ..., m \tag{1}$$

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

- Minimum path segment constraint

For i= 1, 2, ..., m+1, define the length of $P_{i-1}P_i$ segment of the path (as mentioned above, P_0 represents the planning starting point S and P_{m+1} represents the planning ending point T) as S_i, then there is the constraint as below:

$$S_i \geq S_{\min} \quad \forall i = 1, 2, ..., m+1 \tag{2}$$

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 the threat source constraint are considered.

- Mission boundary constraint

For i= 1, 2, ..., m, define the position coordinate of the planning path points P_i as (x_i,y_i), so we get mission boundary constraint:

$$-10000 \leq x_i, y_i \leq 10000 \quad \forall i = 1, 2, ..., m \tag{3}$$

- Threat source constraint

For i= 1, 2, ..., m+1, assume P_{ij} is one point on $P_{i-1}P_i$ segment of the path which be the closest point to the threat source center $Threat_j$ (again, P_0 is the planning starting point S and P_{m+1} is the planning ending point T), then the minimum distance d_{ij} between P_{ij} and T_j can be calculated by the following equation:

$$d_{ij} = \begin{cases} \left|\overrightarrow{P_{i-1}T_j}\right| & if \quad \frac{\overrightarrow{P_{i-1}T_j}\cdot\overrightarrow{P_{i-1}P_i}}{\left|\overrightarrow{P_{i-1}P_i}\right|^2} \le 0 \\ \left|\overrightarrow{P_iT_j}\right| & if \quad \frac{\overrightarrow{P_{i-1}T_j}\cdot\overrightarrow{P_{i-1}P_i}}{\left|\overrightarrow{P_{i-1}P_i}\right|^2} \ge 1 \\ \left|\overrightarrow{P_{ix}T_j}\right| & if \quad 0 < \frac{\overrightarrow{P_{i-1}T_j}\cdot\overrightarrow{P_{i-1}P_i}}{\left|\overrightarrow{P_{i-1}P_i}\right|^2} < 1 \end{cases} \tag{4}$$

Where P_{ix} is the vertical foot point from T_j to the path segment $P_{i-1}P_i$.

Thus, the threat source constraint is obtained as follows:

$$d_{ij} > r_j \quad \forall i = 1, 2, ..., m + 1; \forall j \tag{5}$$

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

2.4 Objective Cost Function

The design of the objective cost function for the UAV 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, 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:

$$\min F = \sum_{i}^{n} \omega_i J_i \tag{6}$$

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:

$$\sum_{i}^{n} \omega_i = 1 \tag{7}$$

If the optimization is not weighted by subcost 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:

$$\min F = 0.2 \times \sum_{i=1}^{m} F_{\varphi_i} + 0.3 \times \sum_{i=1}^{m+1} F_{S_i} + 0.5 \times \sum_{j} \sum_{i=1}^{m+1} F_{Threat_{ij}} \tag{8}$$

Of which, $F_{\varphi i}$ denotes the yaw angle cost of path segment $P_{i-1}P_i$ and satisfies:

$$F_{\varphi_i} = \begin{cases} \Delta\varphi_i & \text{if} \Delta\varphi_i \le \Delta\varphi_{\max} \\ \infty & \text{if} \Delta\varphi_i > \Delta\varphi_{\max} \end{cases} \tag{9}$$

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

$$F_{S_i} = \begin{cases} S_i & \text{if} S_i \ge S_{\min} \\ \infty & \text{if} S_i < S_{\min} \end{cases} \tag{10}$$

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

$$F_{Threat_{ij}} = \begin{cases} r_j/d_{ij} & \text{if} d_{ij} > r_j \\ \infty & \text{if} d_{ij} \le r_j \end{cases} \tag{11}$$

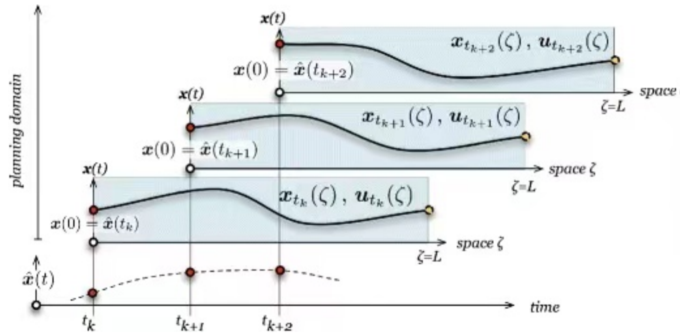

Figure 2: Schematic diagram of receding horizon control. (https://link.springer.com/article/10.1007/s12544-014-0140-6)

3 RECEDING HORIZON CONTROL METHOD

Since the mission targets in the modeled single UAV 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 path planning according to the changing air combat posture during the execution of task becomes the key problem and core difficulty of 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, 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 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 Δt, 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 position of the mission target.

The disassembly process steps are as follows:

- Assume current time is t_k, based on the current UAV_0's position $S(t_k)$ and manned-aircraft's position $T(t_k)$, the optimal $Path(t_k)$ of the UAV_0 from $S(t_k)$ 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 UAV_0, 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 UAV_0 from the current position S(t_{k+1}) to the manned-aircraft's current position 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 manned-aircraft arrived within UAV_0's missile range.

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 WOLF PACK ALGORITHM

Wolf Pack Algorithm (WPA) is a swarm intelligence algorithm emerging in recent years, firstly proposed by Yang et al in 2007 [18], and later improved by Wu Husheng et al [19]. It is a bionic intelligence optimization algorithm inspired by the hunting process of wolves and abstracts three intelligent behaviors of wandering, summoning and besieging, as well as the "winner is the king" head wolf generation rule and "survival of the strongest" wolf pack renewal mechanism.

4.1 Principle of wolf pack algorithm

The wolf pack algorithm draws on the collaborative hunting process of searching prey, summoning companions and besieging prey of wolves and the role differences within the pack based on different division of responsibilities. The artificial wolves are designed with three kinds of responsibilities: head wolf, scout wolf and fierce wolf, which can be transformed into each other under certain conditions, among which: head wolf is the leader of the pack and is responsible for commanding and deciding; scout wolf is the elite of the pack and is responsible for searching prey scent; fierce wolf is the main force of the pack and is responsible for specific hunting activities. Through further abstraction of the intelligent collaborative behavior within the wolf pack and simulation of the survival law of the wolf pack, the final inspiration is applied to the whole process of solving the optimization problem and searching for the optimized solution.

Specifically, the algorithmic process of the WPA consists of the following components:

4.1.1 Random Initialization of Artificial Wolf Packs. Suppose the dimension of the solution space of the optimization problem Y= f(X) to be solved is D. Further, let the artificial wolf pack size be n and the initial iteration round k= 0, generates n artificial wolves in the solution space using random initialization, which are X_1 to X_n. Then for any artificial wolf X_i, it can be expressed as X_i= (x_{i1}, x_{i2}, ..., x_{id}, ..., x_{iD}). For the objective function Y= f(X), let maximization be the optimization direction, and the artificial wolves search the magnitude of the objective function value as the prey odor concentration, while assuming that the distance between different artificial wolves is defined as the Manhattan distance between their state vectors.

4.1.2 The "Winner Is The King" Rule For Head Wolf Generation. In the initial solution space, the prey odor concentration at the location of all randomly initialized artificial wolves is calculated, and the artificial wolf with the highest odor concentration is pushed to be the head wolf; in the process of subsequent iterations, the wolf pack will repeat the pushing process, and if the artificial wolf with the highest prey odor concentration in this round is higher than the odor concentration at the location of the head wolf in the previous round, the artificial wolf will be pushed to be the new head wolf. The head wolf does not perform the following three intelligent behaviors and goes directly to the next iteration of head wolf pushing until it is replaced by other stronger new head wolf.

4.1.3 Wandering Behavior. The n_s artificial wolves in this iteration round except the head wolf are taken as scout wolves for searching prey in the solution space, n_s are randomly taken as integers between [n/(α+1), n/α], and α is the scaling factor of scout wolves. The prey odor concentration Y_i at the location of each scout wolf i is calculated and compared with the prey odor concentration Y_{leader} at the location of the head wolf. If $Y_i > Y_{leader}$, the wolf will take the position of the head wolf and update Y_{leader}= Y_i, and the new head wolf will initiate the summoning behavior; if $Y_i <= Y_{leader}$, the wolf will try to take one step in the surrounding h directions with a fixed wandering $step^a$ and record the prey odor concentration at each new location and then return to the original location, then the wolf will follow p (p= 1,2,...,h) direction and the new location after wandering is:

$$x_{id}^{p} = x_{id} + \sin(2\pi \times \frac{p}{h}) \times step_{d}^{a} \tag{12}$$

Let the prey odor concentration of the new location at this time be Y_{ip}, if the maximum prey odor concentration after wandering one step in h directions is greater than the prey odor concentration of the original location, the scout wolf goes to the new location and updates until the prey odor concentration Y_i perceived by the scout wolf i satisfies $Y_i > Y_{leader}$, or its wandering times W_T reach the upper limit W_T_{max}.

4.1.4 Summoning Behavior. Later in each iteration, the head wolf initiates a summons to all fierce wolves. The number of fierce wolves is n_m= n - n_s -1. The fierce wolves approach the head wolf's location with a fixed running $step^b$ after receiving the head wolf's summons, and the d-dimensional location update is performed according to the following equation for any fierce wolf j in k+1 iterations:

$$x_{jd}^{k+1} = x_{jd}^{k} + step_{d}^{b} \times (g_{d}^{k} - x_{jd}^{k}) / \left| g_{d}^{k} - x_{jd}^{k} \right| \tag{13}$$

Where $g^k{}_d$ denotes the dth dimensional component of the head wolf's kth round coordinate vector. Partly similar to the wandering behavior of scout wolves, when the prey odor concentration in the new location of fierce wolf j after running is greater than the prey odor concentration in the location of head wolf, i.e., $Y_j > Y_{leader}$, then fierce wolf j becomes the new head wolf; conversely, if $Y_j <= Yl_{eader}$, calculate whether the distance d_{jl} from the location of fierce wolf j to the location of head wolf after this round of running is smaller than the judging distance d_{near}, if it is smaller, then the fierce wolf will change from running behavior to besieging behavior. The judging distance d_{near} will be calculated by the following formula:

$$d_{near} = \frac{1}{D \times \omega} \times \sum_{d=1}^{D} |Max_d - Min_d| \tag{14}$$

The value of ω, which is the distance determination factor, largely determines the convergence speed of the wolf pack algorithm. In general, the larger the value of ω, the faster the convergence speed of the wolf pack algorithm, however, too large a value of ω can also make the algorithm lack local search ability and thus cannot enter the besieging behavior. Max_d and Min_d denote the upper and lower bounds of the dth dimension of the solution space, respectively.

4.1.5 Besieging Behavior. The besieging behavior represents the hunting activity of the wolf pack for the prey. At this time, the location of the head wolf (with the strongest prey odor concentration) is regarded as the prey location, and if the number of iterative rounds is k, the location can be written as g^k, and $g^k{}_d$ denotes the dth dimensional component of the head wolf's kth round coordinate vector. Then, after entering the besieging behavior, the location update formula of all artificial wolves except the head wolf in (k+1) th round is as follows:

$$x_{id}^{k+1} = x_{id}^{k} + \lambda \times step_d^c \times \left| g_d^k - x_{id}^k \right| \tag{15}$$

Where λ is a random number uniformly distributed in the interval of [-1,1]; $step^c$ is the siege step; if the prey odor concentration at the location of the artificial wolf in the current round after the implementation of besieging behavior is greater than the prey odor concentration at the location of the previous round, the location of this artificial wolf is updated.

The wandering $step^a$, running $step^b$ and besieging $step^c$ involved in the above three intelligent behaviors have the following relationships:

$$step_d^a = \frac{step_d^b}{2} = 2step_d^c = \frac{|Max_d - Min_d|}{C} \tag{16}$$

where C denotes the step size factor, and its size determines the search accuracy of the wolf pack algorithm.

4.1.6 The Wolf Pack Renewal Mechanism of "Survival of The Strongest". At the end of each iteration, the wolf pack is updated according to the principle of "survival of the strongest", when the R artificial wolves with the lowest prey odor concentration (objective function value) are selected for elimination, while R new artificial wolves are randomly generated. The choice of the value of R is the trade-off between the global search ability and local search ability of the wolf pack algorithm, and is usually taken as random integers within $[n/(2\beta), n/\beta]$, and β is the wolf update proportionality factor.

4.2 Improvements to the wolf pack algorithm

4.2.1 Synchronous Adaptive step Adjustment Mechanism Based on Step Phase Factor. In the wandering, summoning and besieging behaviors of the wolf pack algorithm, the associated wandering $step^a$, running $step^b$ and besieging $step^c$ are all fixed vectors. The setting of these steps affects the search ability and convergence speed of the algorithm itself to a large extent. If the step size is set too large, it will easily affect the convergence of the algorithm in the later stage; if the step size is set too small, it will lead to slower search in the early stage of the algorithm and inefficiency of the algorithm. It can be seen that it is difficult to balance the performance requirements of the early stage of the algorithm and the late stage of the algorithm by using a fixed step size factor C in the whole iterative process of the wolf pack algorithm, which seems too rigid. However, if the adaptive improvement design is made for each of the three step size parameters, it is easy to lose the coordination among the parameters and violate the proportional relationship shown in formula (16).

Therefore, we purposes to adjust the step size factor C adaptively, and complete the synchronous adaptive adjustment for the three step lengths of wandering $step^a$, running $step^b$ and besieging $step^c$ by introducing the phase factor $e^{(k/kmax)}$. The improved synchronous adaptive step adjustment formula is as follows:

$$step_d^a = \frac{step_d^b}{2} = 2step_d^c = \frac{|Max_d - Min_d|}{(e^{\frac{k}{k_{max}}})C} \tag{17}$$

It can be seen that as the number of iterations increases, the three step sizes will be reduced adaptively in the same proportion, which ensures the "coarse before and fine after" iteration of the algorithm and compensates for the imbalance between the global search capability and the convergence speed of the original algorithm.

4.2.2 Improvement of Scout Wolf's Wandering Behavior. In WPA, the setting of the search direction number h of wandering behavior has a great influence on the local search ability of the algorithm. The larger the value of wandering direction h is set, the higher the search accuracy of the wolf pack algorithm will be, but due to the increase of computation, it in turn limits the efficiency of the algorithm and also tends to increase the possibility of falling into local optimum. In general, the optimization algorithm should focus more on the search ability in the early stage and more on the convergence ability in the later stage of the iteration. To this end, a phase factor $\lceil \ln(k_{max}/k+1) \rceil$ can be similarly introduced for h so that it can be gradually reduced with the increase in the number of algorithm iterations. The improved wandering behavior is formulated as follows:

$$x_{id}^{p} = x_{id} + \sin(2\pi \times \frac{p}{h'}) \times step_d^a \tag{18}$$

At last, actually, we could find out that the improved WPA is consistent with the original algorithm in terms of algorithm steps, and the difference only lies in the adaptive change of parameters.

5 2D-EQUAL-STEP PATH GENERATION METHOD

The 2D-equal-step path generation method was proposed by Zhu W et al in 2013 [20] 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 WPA for an attempted solution of single UAV online path planning for moving mission targets in dynamic mission environment.

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

- Determine the planning starting point S (UAV_0's current position) and the end point T (moving mission's current

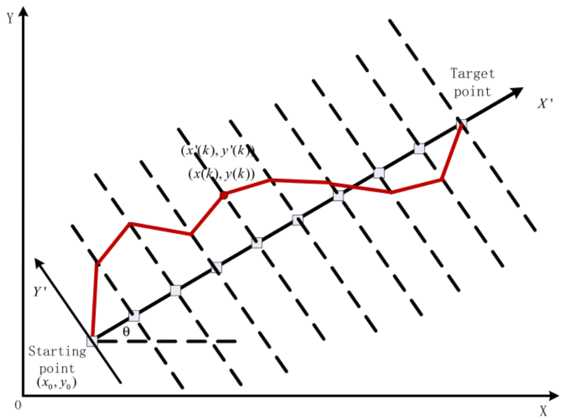

Figure 3: Coordinate transformation of 2D-equal-step path generation. [20]

position) for two-dimensional spatial path planning, connect ST and equate ST into (m+1) segments by m equidistant interval points, and the length of each segment, i.e., the planning step length d=|ST|/(m+1).

- At each equidistant interval point make m lines L_1, L_2, ..., L_k, ..., L_m perpendicular to ST, 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:

$$Path = \{S \to (x_1, y_1) \to (x_2, y_2) \to \dots \to (x_k, y_k) \to \dots \to (x_m, y_m) \to T\} \tag{19}$$

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

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

where (x_s,y_s) is the coordinate of the planning starting point S 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(ST) of the new coordinate system, which can be calculated by the following equation:

$$\theta = \arctan\left(\frac{y_t - y_s}{x_t - x_s}\right) \tag{20}$$

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

- 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 complexity of the solution.

At the present 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 PRM algorithm and 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 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 WPA, 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 online path planning under the receding horizon control mechanism.

6 SIMULATION EXPERIMENT

In order to verify the research idea of "the improved wolf pack algorithm combining with the receding horizon control method and 2D-eaual-step path generation method can solve the single UAV online path planning problem for moving mission targets in dynamic mission environment", and to compare the performance difference between the improved wolf pack algorithm and the basic wolf pack 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 WPA and improved WPA are set as Table 1 display:

Figure 4 shows the planning path of the basic WPA combined with RHC and 2D-equal-step method from the initial posture (t= 0) to the terminal posture (t= 47), Figure 5 shows the planning path of the improved WPA combined with RHC and 2D-equal-step method from the initial posture (t= 0) to the terminal posture (t= 38).

We could find that, on the whole, both basic WPA and improved WPA have well planned the flight path to avoid all the threat sources for UAV_0. However, the planned path of the latter is smoother and more robust than that of the former (though it is not obviously between the comparison of first subgraph or second subgraph, yet no more hide in the comparison of third subgraph). Actually, the fifth segments of the planed path by WPA in the third subgraph of Figure 4 visibly violate the maximum yaw angle constraint introduced in chapter 2, hence this path should not be regarded as valid

Table 1: The initial parameters of WPA and improved WPA

Parameters of WPA	The Initial Value	Parameters of Improved WPA	The Initial Value
n: Artificial Wolf Pack Size	100	n: Artificial Wolf Pack Size	100
k: Initial Iteration Round	0	k: Initial Iteration Round	0
D: The Dimension of Solution Space[1]	9	D: The Dimension of Solution Space[1]	9
α: Scaling Factor of Scout Wolves	4	α: Scaling Factor of Scout Wolves	4
h: The Number of Wandering Direction	10	h: The Number of Wandering Direction	10
W_T_{max}: Maximum Wandering Times	30	W_T_{max}: Maximum Wandering Times	30
ω: Distance Determination Factor	0.01	ω: Distance Determination Factor	0.01
C: Step Size Factor	100	C: Step Size Factor	100
β: Wolf Update Proportionality Factor	6	β: Wolf Update Proportionality Factor	6
k_{max}: Maximum Iterations	200	k_{max}: Maximum Iterations	200

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

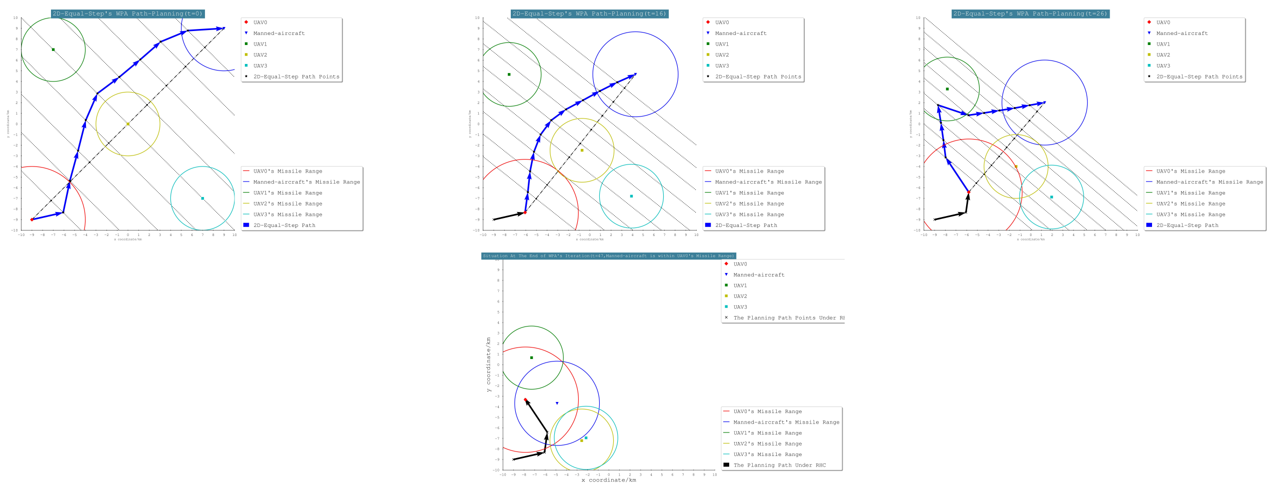

Figure 4: The air warfare posture from the initial moment to the teminal moment for the basic WPA.

result. And throughout the whole planning process, the UAV_0's RHC path planed by improved WPA is always out of the enemies' missile range, which ensures priority strikes and finally completed the strike mission against the target of moving manned aircraft, while, the UAV_0's RHC path planed by basic WPA is not.

We could further find the difference in performance between the two algorithms from the Figure 6 below, which reveals that the improved WPA is much faster than the basic WPA in the convergence rate.

7 CONCLUSIONS

In this paper,an algorithm based on the improved wolf pack algorithm combined with receding horizon control method and 2D-equal-step path generation method was proposed for solving the single UAV online path planning problem aiming at moving targets in dynamic mission environment. The generated path can ensure the maximum safety with the minimum fuel cost and steering cost of UAV within the various constraints, as shown in the simulation results. Meanwhile, this algorithm ensures the real-time performance and possibility of UAV online path planning through combination of various methods. Further, the comparative simulation between basic WPA and improved WPA 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, which is a challenging issues for next stage.

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] Hong, Y., Jung, S., Kim, S., & Cha, J. (2020). Multi-UAV Routing with Priority using Mixed Integer Linear Programming. 2020 20th International Conference on Control, Automation and Systems (ICCAS), 699-702
[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

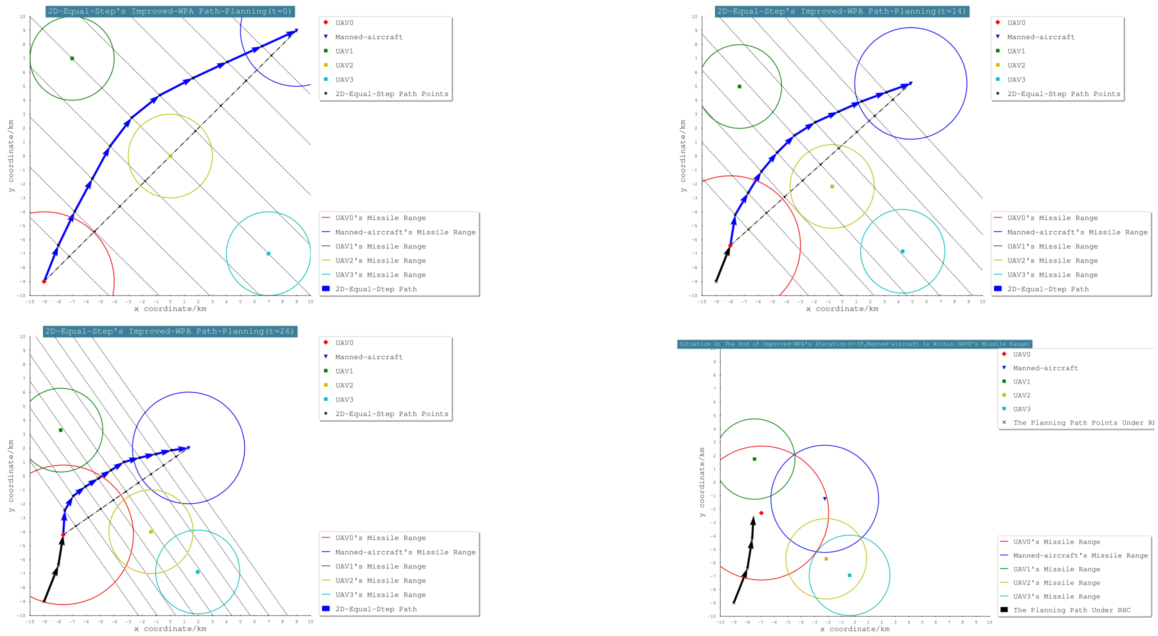

Figure 5: The air warfare posture from the initial moment to the teminal moment for the improved WPA.

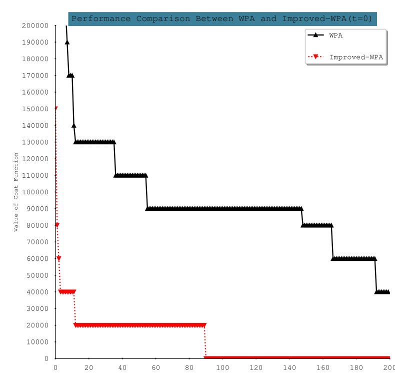

Figure 6: Comparison of optimization ability between basic WPA and improved WPA in the same solution space (t= 0).

[4] Tan, W., Hu, Y., Zhao, Y., Li, W., Zhang, X., & Li, Y. (2021). Mission Planning for Unmanned Aerial Vehicles Based on Voronoi Diagram-Tabu Genetic Algorithm. Wireless Communications and Mobile Computing
[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] Junshu, H., Liu, Y., Tian, L., & Zheng, Y. (2021). Dynamic A* Algorithm to Improve Dynamic Path Planning of Unmanned Epidemic Prevention and Killing Vehicles. 2021 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 1-4
[7] Karve, D., & Kapadia, F. (2020). Multi-UAV Path Planning using Modified Dijkstra's Algorithm. International Journal of Computer Applications, 175, 26-33
[8] Noh, G., Park, J., Min, C., & Lee, D. (2022). Parallelization of Probabilistic RoadMap for Generating UAV Path on a DTED Map. Journal of the Korean Society for Aeronautical & Space Sciences
[9] Castro, G.G., Berger, G.S., Cantieri, Á.R., Teixeira, M., Lima, J., Pereira, A.I., & Pinto, M.F. (2023). Adaptive Path Planning for Fusing Rapidly Exploring Random Trees and Deep Reinforcement Learning in an Agriculture Dynamic Environment UAVs. Agriculture
[10] Pan, Z., Zhang, C., Xia, Y., Xiong, H., & Shao, X. (2022). An Improved Artificial Potential Field Method for Path Planning and Formation Control of the Multi-UAV Systems. IEEE Transactions on Circuits and Systems II: Express Briefs, 69, 1129-1133
[11] Chu, H., Yi, J., & Yang, F. (2022). Chaos Particle Swarm Optimization Enhancement Algorithm for UAV Safe Path Planning. Applied Sciences.
[12] Wei, T.C., Hu, Y., Zhao, Y., Li, W., He, J., & Zhen, Z. (2022). An optimized genetic algorithm for single UAV mission planning. Other Conferences
[13] Jia, Y., Zhou, S., Zeng, Q., Li, C., Chen, D., Zhang, K., Liu, L., & Chen, Z. (2022). The UAV Path Coverage Algorithm Based on the Greedy Strategy and Ant Colony Optimization. Electronics
[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] Ma, X., Zang, S., Li, X., & Ma, J. (2022). Receding Horizon Control with Extended Solution for UAV Path Planning. International Journal of Aerospace Engineering
[18] Yang, C., Tu, X., & Chen, J. (2007). Algorithm of Marriage in Honey Bees Optimization Based on the Wolf Pack Search. The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007), 462-467
[19] Wu, H., & Zhang, F. (2014). Wolf Pack Algorithm for Unconstrained Global Optimization. Mathematical Problems in Engineering, 2014, 1-17
[20] Zhu, W., & Duan, H. (2014). Chaotic predator–prey biogeography-based optimization approach for UCAV path planning. Aerospace Science and Technology, 32, 153-161