*Int. J. Appl. Math. Comput. Sci., 2026, Vol. 36, No. 2, 195–209*
DOI: 10.61822/amcs-2026-0014

# RESEARCH ON UAV DYNAMIC ADVERSARIAL GAME STRATEGIES USING AN IMPROVED MARINE PREDATOR ALGORITHM

NANNAN CHENG [a,*], QING LI [a], HENG WANG [a], SAISAI TONG [b], XINGJIAN FU [c]

[a]Key Laboratory of Knowledge Automation for Industrial Processes of the Ministry of Education
School of Automation and Electrical Engineering
University of Science and Technology Beijing
30 Xueyuan Road, Haidian District, Beijing, China
e-mail: cheng@ustb.edu.cn

[b]School of Automation Science and Electrical Engineering
Beihang University
37 Xueyuan Road, Haidian District, Beijing, China

[c]School of Automation
Beijing Information Science and Technology University
12 Qinghe Xiaoying East Road, Haidian District, Beijing, China

This paper proposes an improved version of the marine predator algorithm (abbreviated as IMPA) to optimize unmanned aerial vehicle (UAV) dynamic adversarial game strategies, effectively addressing critical limitations of the original marine predator algorithm (MPA), including slow convergence speed and a pronounced tendency to fall into local optima. The IMPA incorporates four key innovations: opposition-based learning (OBL) for enhanced initial population quality, an adaptive mechanism for dynamic population resizing, nonlinear step-size control, and an inertia weight strategy. These improvements collectively accelerate convergence, balance global exploration with local exploitation, and significantly improve the ability to escape local optima. To validate the algorithm's performance, a dynamic game model based on situation assessment and utility functions is established for red–blue UAV confrontation scenarios, and the IMPA is successfully applied to solve for Nash equilibrium. Comprehensive simulation results demonstrate that the proposed algorithm converges over 40% faster than the original MPA and other benchmark algorithms, and robustly achieves the Nash solution in all 100 consecutive tests, underscoring its superior effectiveness and reliability in dynamic adversarial environments.

**Keywords:** improved marine predator algorithm, UAV dynamic adversarial game, convergence speed, local optima, Nash equilibrium.

## 1. Introduction

The rapid advancement of UAV technology has expanded its applications in military operations, including not only surveillance and reconnaissance but also offensive operations (e.g., Merheb *et al*., 2015; Salazar *et al*., 2020). This expansion has, in turn, spurred extensive research into UAV cooperative combat, encompassing critical areas such as attack strategy selection, task allocation, and cooperative control (Xu *et al*., 2020; Liu *et al.*, 2022b; 2022c). Xu *et al*. (2020) have proposed a multi-UAV confrontation model considering the uncertainty of war information. To solve its impact, an interval decision method has been employed for game-theoretic solutions. Liu *et al*. (2022b) have established a bilateral combat model and applied an iterative game method to obtain Nash equilibrium strategies in multi-UAV air combat scenarios. Further addressing information uncertainty, Liu *et al*. (2022c) have utilized interval-valued parameters to represent incomplete information and employed quantum particle swarm optimization to solve for mixed-strategy Nash equilibria in adversarial engagements. For maritime

*Corresponding author

strike missions, Kim *et al*. (2021) have formulated a UAV adversarial model and introduced a heuristic social-learning particle swarm optimization (SL-PSO) algorithm to optimize task allocation. Meanwhile, Fu *et al*. (2022) have proposed a pursuit-evasion strategy that integrates the deep deterministic policy gradient (DDPG) algorithm with imitation learning (IL). This integrated approach, termed IL-DDPG, uses a quasi-proportional guidance law to generate expert demonstrations and improves exploration efficiency through experience replay mechanisms.

All aforementioned studies focus on single-step combat models, whereas realistic battlefield engagements are inherently multi-stage and dynamic. To address this, researchers have proposed dynamic game-theoretic frameworks. Duan *et al*. (2015) have developed a novel methodology for dynamic multi-UAV confrontation target allocation. Grounded in dynamic game theory, a multi-stage engagement game model have been formulated and the predatory particle swarm optimization (PPSO) algorithm has been employed to derive its Nash equilibrium strategies. Addressing dynamic confrontation challenges in UAV attacks, Wang *et al*. (2015) have developed an air combat game model incorporating survival probability and weapon expenditure, utilizing a modified particle swarm algorithm with elite reselection to obtain high-accuracy Nash equilibria. Li *et al*. (2019) have proposed an intuitionistic fuzzy game-theoretic framework that integrates game theory with intuitionistic fuzzy sets to address multi-UAV air combat under uncertainty, and have employed an enhanced differential evolution algorithm to solve the model. Liu *et al*. (2021a) have established a multi-UAV combat threat assessment model for adversarial engagements and have employed an enhanced estimation of distribution algorithm to derive optimal game strategies.

Despite these advances, algorithm performance in highly uncertain and dynamic environments remains limited (e.g., Kantue and Pedro, 2022). Recent improvements to meta-heuristic algorithms are quite promising. For example, Gong *et al*. (2024) have proposed an enhanced quantum particle swarm optimization algorithm with diversity migration (DM-QPSO), which improves global exploration but struggles in high-dimensional spaces. Song *et al*. (2023) have developed a dynamically hybridized differential evolution algorithm, but its complex parameterization limits adaptability. The marine predators algorithm (MPA) (e.g., Faramarzi *et al*., 2020; Jiang *et al*., 2022) has emerged as a promising approach, using Brownian and Lévy motion to balance local refinement and global exploration in UAV adversarial decision-making. Enhanced variants further improve its utility, namely, Fu *et al*. (2023) have introduced a phased-improved MPA integrating differential evolution, sine-cosine fluctuations, and Cauchy mutation with opposition-based learning, applicable to fault diagnosis; Oszust (2021) have incorporated a local escape operator to mitigate premature convergence from imbalanced exploration-exploitation. Beyond the MPA, other nature-inspired approaches have been advanced. Liu *et al*. (2021b) have formulated an intuitionistic fuzzy game model for multi-AUV dynamic confrontations and efficiently solved it via fractional-order particle swarm optimization. Separately, Li *et al*. (2022) have employed nonlinear inertia weights in PSO to better balance exploration and exploitation, thereby reducing premature convergence.

Nevertheless, these improved strategies have yet to be systematically integrated into MPA-based optimization for UAV dynamic adversarial games, thereby limiting the algorithm's effectiveness in this domain. Practical multi-UAV dynamic engagement scenarios present several persistent challenges:

- high environmental uncertainty, including unpredictable enemy maneuver strategies and sensor inaccuracies due to battlefield disturbances, demands exceptional algorithmic robustness;
- multi-UAV cooperative decision-making requires frameworks that reconcile individual tactical actions with group coordination and overall payoff optimization (e.g., Tizhoosh, 2005; Ramezani *et al*., 2021; Fan *et al*., 2022).

Furthermore, inherent limitations of the MPA such as inefficient and non-diverse initial population generation and a tendency toward local optima can lead to suboptimal convergence behavior and hinder global optimization. These issues currently restrict the MPA's applicability in dynamic UAV adversarial games.

Therefore, this paper proposes an IMPA to systematically enhance the original MPA's performance in multi-stage dynamic games by increasing global search efficiency, avoiding local optima, and improving environmental adaptability. The improvements incorporate opposition-based learning (OBL) during population initialization to accelerate convergence through simultaneous evaluation of the original and opposite solutions. An adaptive population mechanism dynamically adjusts the population size per iteration, thus promoting diversity. Moreover, nonlinear step-size control dynamically adapts to the search state, enabling an adaptive balance between exploration and exploitation and improving both global search capacity and local convergence precision. Finally, inertia weight adjustment is integrated to further suppress convergence to local optima.

The article is structured as follows. The scenarios of drone dynamic confrontation are analyzed in Section 2, constructing a payoff function based on situation

assessment and a multi-stage dynamic confrontation game model. Section 3 elaborates on the principles of the original MPA, conducts an in-depth analysis of its limitations in dynamic games, and proposes the IMPA, including key improvement strategies such as opposition-based learning initialization, adaptive population regulation, and inertia weight adjustment. Section 4 verifies the effectiveness of the algorithm from multiple dimensions through simulation experiments by comparing the IMPA with the genetic algorithm (GA), the PSO algorithm, and the original MPA. Finally, Section 5 summarizes the paper.

## 2. UAV dynamic adversarial model

**2.1. UAV confrontation posture.** The tactical dominance of UAVs in combat is determined by the relative engagement posture, including angle, altitude, speed, and distance. These factors collectively dictate the outcome of an engagement, underscoring the critical role of confrontation posture in air combat scenarios. The typical confrontation posture between two UAVs is illustrated in Fig. 1.

In the figure, the speed and altitude of Red UAV $i$ are denoted as $v_i$ and $h_i$, respectively, while those of Blue UAV $j$ are $v_j$ and $h_j$. Further, $\theta_i$ and $\theta_j$ are the direction angles of Red UAV $i$ and Blue UAV $j$, respectively, with respect to each other. Lastly, $d_{i,j}$ denotes the relative distances between the two-sides UAVs. The specific confrontation posture functions are defined as follows.

1. Angular posture $P_\theta$ is denoted by

$$P_\theta = \frac{1}{2}\Big(1 + \frac{|(\pi - \theta_j)| - |\theta_i|}{2\pi}\Big), \tag{1}$$

where $|\theta_i|$ and $|\theta_j|$ stand for the absolute values of the direction angles of the relative positions of the two UAVs.

2. Height posture $P_h$ is (Zhang *et al.*, 2022) written as

$$P_h = \begin{cases} e^{-\frac{h_i - h_{ibest}}{h_{ibest}}}, & h_{ibest} \le h_i, \\ e^{\frac{h_i - h_{ibest}}{h_j}}, & h_j < h_i \le h_{ibest}, \\ -0.5 + \frac{h_i}{h_j}, & h_i \le h_j, \end{cases} \tag{2}$$

where $h_i$ and $h_j$ are the altitudes of UAV $i$ on the red side and UAV $j$ on the blue side, respectively, and $h_{ibest}$ is the optimal air combat altitude for UAV $i$ on the red side.

3. Velocity posture $P_v$ is denoted by

$$P_v = \begin{cases} 0.1, & v_i \le 0.67 v_j, \\ -0.5 + \dfrac{v_i}{v_j}, & 0.67 v_j < v_i \le 1.5 v_j, \\ 1, & v_i > 1.5 v_j, \end{cases} \tag{3}$$

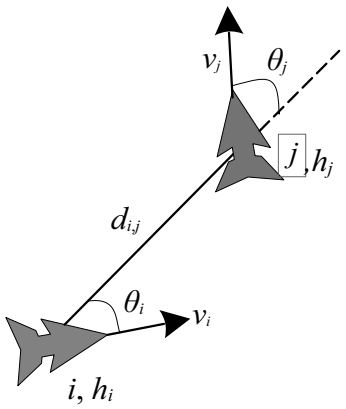



Fig. 1. Confrontation posture of the two-sides UAVs.

where $v_i$ and $v_j$ are the speeds of UAV $i$ on the red side and UAV $j$ on the blue side.

4. Distance posture $P_d$ is written as

$$P_d = e^{-\frac{|d_{i,j} - d_i^m|}{2d_{i,j}}}, \tag{4}$$

where $d_{i,j}$ is the relative distance between the two UAVs, while

$$d_i^m = \frac{1}{2}(d_i^{\max} + d_i^{\min}),$$

where $d_i^{\max}$ and $d_i^{\min}$ denote respectively the maximum and minimum ranges of the missiles carried by UAV $i$.

To summarize, the overall posture of Red UAV $i$ against Blue UAV $j$ is

$$P_{i,j} = \omega_1 P_\theta + \omega_2 P_h + \omega_3 P_v + \omega_4 P_d, \tag{5}$$

where $\omega_1$, $\omega_2$, $\omega_3$ and $\omega_4$ are weighting factors while $\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$.

**2.2. Payoff functions for UAV adversarial games.** The destruction probability $Q_{i,j}^r$ of Red UAV_$i$ against Blue UAV_$j$ is

$$Q_{i,j}^r = \alpha Q_r P_{i,j} + \omega_3 P_v + \omega_4 P_d, \tag{6}$$

where $\alpha$ is the environmental impact factor, $Q_r$ is the ideal kill probability of Red's missile, and $P_{i,j}$ denotes the engagement level of Red UAV against Blue UAV.

The survival probability $G_j^b$ of UAV $j$ of the blue side is

$$G_j^b = \prod_{i=1}^{n}(1 - Q_{i,j}^r), \quad j = 1, 2, \ldots, m. \tag{7}$$

Similarly, the destruction probability of Blue UAV$_j$ against Red UAV$_i$ is

$$Q_{j,i}^b = \alpha Q_b P_{j,i}, \tag{8}$$

where $Q_b$ is the ideal kill probability of the blue side's missile while $P_{j,i}$ denotes the engagement level of the blue side's UAV against the red side's UAV.

The survival probability $G_i^r$ of UAV $i$ of the red side is

$$G_i^r = \prod_{j=1}^{m}(1 - Q_{j,i}^b), \quad i = 1, 2, \ldots, n. \tag{9}$$

Let the red side have $N_r$ UAVs and the blue side have $M_b$ UAVs. Each drone of both sides can only choose one UAV to attack at a time, and the air combat confrontation is divided into $T_{total}$ steps. The destruction probability of Red and Blue UAVs is

$$Q_{i,j}^r(T) = \alpha_1 Q_{i,j}^r P_{i,j}^r(T-1), \tag{10}$$

$$Q_{j,i}^b(T) = \alpha_2 Q_{j,i}^b P_{j,i}^b(T-1), \tag{11}$$

where the number of combat steps of both UAVs is $T = 1, 2, \ldots, T_{total}$, $Q_{i,j}^r(T)$ is the destruction probability of Red UAV $i$ against Blue UAV $j$ at step $T$ of the combat, $Q_{j,i}^b(T)$ is the destruction probability of Blue UAV $j$ against Red UAV $i$ when the battle reaches step $T$, $\alpha_1$ and $\alpha_2$ are the influence factors of the dynamic confrontation environment, $Q_{i,j}^r$ and $Q_{j,i}^b$ are the destruction probabilities of both UAVs against each other in an ideal dynamic combat environment, and $P_{i,j}^r(T-1)$ and $P_{j,i}^b(T-1)$ are the overall adversarial posture degrees of both UAVs after step $T-1$ of the battle.

The survival probability of both Red and Blue UAVs is expressed as (Ma *et al*., 2022)

$$G_i^r(T) = G_i^r(T-1)\prod_{j=1}^{M_b}(1 - Q_{j,i}^b(T))^{C_{j,i}^b(T)}, \tag{12}$$

$$G_j^b(T) = G_j^b(T-1)\prod_{i=1}^{N_r}(1 - Q_{i,j}^r(T))^{C_{i,j}^r(T)}, \tag{13}$$

where $G_i^r(T)$ and $G_j^b(T)$ are the survival probabilities of Red UAV $i$ and Blue UAV $j$, respectively, after $T$ steps of the battle, $C_{j,i}^b(T)$ is the number of missiles used by Blue UAV $j$ against Red UAV $i$ at step $T$ of the battle, and $C_{i,j}^r(T)$ is the number of missiles used by Red UAV $i$ against Blue UAV $j$ at step $T$ of the battle.

Red and Blue UAVs missile intories must satisfy the following:

$$\sum_{j=1}^{M_b} C_{i,j}^r(T) \le w_i^r(T), \quad \forall i = 1, 2, \ldots, N_r, \tag{14}$$

$$\sum_{i=1}^{N_r} C_{j,i}^b(T) \le w_j^b(T), \quad \forall j = 1, 2, \ldots, M_b, \tag{15}$$

$$\sum_{T=1}^{T_{total}}\sum_{j=1}^{M_b} C_{i,j}^r(T) \le W_i^r, \quad \forall i = 1, 2, \ldots, N_r, \tag{16}$$

$$\sum_{T=1}^{T_{total}}\sum_{i=1}^{N_r} C_{j,i}^b(T) \le W_j^b, \quad \forall j = 1, 2, \ldots, M_b, \tag{17}$$

where $w_i^r(T)$ and $w_j^b(T)$ are the maximum number of missiles that can be launched by Red UAV $i$ and Blue UAV $j$, respectively, in step $T$ of the battle, and $W_i^r$ and $W_j^b$ denote the maximum number of missiles that can be fired by Red UAV $i$ and Blue UAV $j$, respectively, during the entire battle.

The gain function for both UAVs is described as

$$f^r = \sum_{i=1}^{N_r}\frac{u_i^r G_i^r(T)}{C_i^r} - \sum_{j=1}^{M_b}\frac{u_j^r G_j^b(T)}{C_j^b}, \tag{18}$$

$$f^b = \sum_{j=1}^{M_b}\frac{u_j^b G_j^b(T)}{C_j^b} - \sum_{i=1}^{N_r}\frac{u_i^b G_i^r(T)}{C_i^r}, \tag{19}$$

where $f^r$ and $f^b$ represent the revenue functions for the red and blue sides, respectively, the terms $u_i^r$ and $u_j^r$ denote the values of Red UAV $i$ and Blue UAV $j$ relative to the red side while $u_i^b$ and $u_j^b$ indicate their values relative to the blue side, and $C_i^r$ and $C_j^b$ are the sum of the number of missiles fired by Red UAV $i$ and Blue UAV $j$, respectively.

Based on the analysis, both sides' strategies depend on the number of combat steps $T$. Each UAV aims to maximize its own gains while minimizing the enemy's benefits. The goal is to maximize their own side's advantage. From the adversarial strategies, the gain matrices for Red and Blue UAVs are derived following $T_{total}$ time steps:

$$F_r = \begin{bmatrix} f_{1,1}^r & f_{1,2}^r & \cdots & f_{1,mb}^r \\ f_{2,1}^r & f_{2,2}^r & \cdots & f_{2,mb}^r \\ \vdots & \vdots & \ddots & \vdots \\ f_{nr,1}^r & f_{nr,2}^r & \cdots & f_{nr,mb}^r \end{bmatrix}, \tag{20}$$

$$F_b = \begin{bmatrix} f_{1,1}^b & f_{1,2}^b & \cdots & f_{1,mb}^b \\ f_{2,1}^b & f_{2,2}^b & \cdots & f_{2,mb}^b \\ \vdots & \vdots & \ddots & \vdots \\ f_{nr,1}^b & f_{nr,2}^b & \cdots & f_{nr,mb}^b \end{bmatrix}, \tag{21}$$

where $F_r$ and $F_b$ are the payoff matrices of the red and blue sides, respectively, and $nr$ and $mb$ are their respective numbers of strategies.

## 3. Nash equilibrium in UAV confrontation

**Definition 1.** Assume a two-player finite noncooperative game: $\Omega = \{S_1, S_2; f_1, f_2\}$, where $S_1 = \{\eta_1, \eta_2, \ldots, \eta_m\}$ and $S_2 = \{\mu_1, \mu_2, \ldots, \mu_n\}$ are the sets of optional strategies for players 1 and 2, respectively, and $\{f_1, f_2\}$ is the payoff function of players 1 and 2. If game players 1 and 2 choose strategies $\eta^* \in S_1$ and $\mu^* \in S_2$, respectively, to form a situation $(\eta^*, \mu^*)$ satisfying

$$f_1(\eta^*, \mu^*) =\geq f_1(\eta, \mu^*), \forall \eta \in S_1, \tag{22}$$

$$f_2(\eta^*, \mu^*) =\geq f_2(\eta^*, \mu), \forall \mu \in S_2, \tag{23}$$

then the situation $(\eta^*, \mu^*)$ is said to be a pure-strategy Nash equilibrium of the game. If strategy $\eta_i, i = 1, 2, \cdots, m$, is adopted by player 1 with probability $x_i$ and strategy $\mu_j$, $j = 1, 2, \cdots, n$, is adopted by player 2 with probability $y_j$, then

$$X = \{x = (x_1, x_2, \ldots, x_m) | \sum_{i=1}^{m} x_i = 1, x_i \geq 0\}, \tag{24}$$

$$Y = \{y = (y_1, y_2, \ldots, y_n) | \sum_{j=1}^{n} y_j = 1, y_j \geq 0\} \tag{25}$$

form the set of mixed strategies of players 1 and 2, respectively, $\forall x \in X, \forall y \in Y, (x, y)$ is a mixed situation, and $x$, $y$ are the mixed strategies of players 1 and 2, respectively.

**Definition 2.** In the game $\Omega = \{S_1, S_2; F_1, F_2\}$, where $\{F_1, F_2\}$ is the payoff matrix of players 1 and 2, if there exists a mixed situation $(x^*, y^*)$ satisfying $x^* \in X$, $y^* \in Y$, and for any other mixed strategy $x \in X$, $y \in Y$ there are

$$x^* F_1 y^{*T} \geq x F_1 y^{*T}, \tag{26}$$

$$x^* F_2 y^{*T} \geq x^* F_2 y^T, \tag{27}$$

then the mixed situation $(x^*, y^*)$ is said to be a mixed-strategy Nash equilibrium for game $\Omega$.

From Definitions 1 and 2, the Nash equilibrium strategy is a steady state in the game. In this state, the optimal strategy is chosen by each participant. If any of the parties change their strategies, their payoffs are not increased. This is because changing strategies destroys the original combination of optimal strategies, which results in their own final payoff being reduced. In a Nash equilibrium, the optimal strategy has already been used and there is no better strategy that can be chosen by each participant.

**Lemma 1.** (Zhao *et al.*, 2019) *The sufficient condition for a mixed strategy $(x^*, y^*)$ to be a Nash equilibrium strategy for game $\Omega = \{S_1, S_2; F_1, F_2\}$ is*

$$F_1^{i,} y^{*T} \leq x^* F_1 y^{*T}, \qquad \forall i = 1, 2, \ldots, m, \tag{28}$$

$$x^* F_2^{,j} \leq x^* F_2 y^{*T}, \qquad \forall j = 1, 2, \ldots, n, \tag{29}$$

*where $F_1^{i,}$ is the $i$ row of the player 1 payoff matrix $F_1$ and $F_2^{,j}$ is the $j$ column of the player 2 payoff matrix $F_2$.*

The IMPA is employed in this paper to solve the Nash equilibrium strategy of the UAV dynamic adversarial game. The fitness function is defined as follows:

$$\begin{aligned} f(x, y) = &\max\left\{\max_i\left\{F_1^{i,} y^{*T} - x^* F_1 y^* T\right\}, 0\right\} \\ &+ \max\left\{\max_j\left\{x^* F_2^{,j} - x^* F_2 y^{*T}\right\}, 0\right\}. \end{aligned} \tag{30}$$

By Lemma 1, the fitness function $f(x, y)$ is minimized if and only if $(x, y)$ is a Nash equilibrium strategy for the game and the minimum value is 0.

### 3.1. Marine predator algorithm.

The MPA is structured into three phases: exploration, transition, and exploitation, governed by rules that mimic natural predator-prey dynamics. Each phase employs distinct stochastic movement strategies: Brownian motion for localized search in prey-rich regions and Lévy flight for long-jump exploration in resource-sparse environments. The implementation proceeds as follows.

*1. Initialization phase.*

During initialization, a population of candidate solutions (prey) is randomly distributed across the search space while determining the initial population size. The initial population of size $n$ is defined as

$$X_i = X_{\min} + \text{rand}(X_{\max} - X_{\min}), \quad i = 1, 2, \ldots, n, \tag{31}$$

where $X_i$ is the $i$ member of the initial population, $X_{\max}$ and $X_{\min}$ denote the upper and lower limits, respectively, and rand is a random vector in the range $[0, 1]$.

The fitness value of each member of the initial population is evaluated by means of a fitness function. According to the theory of survival of the fittest, the highest-fitness individuals constitute the predators and are referred to as elite members. The vector of elite members is copied $n$ times to form the elite matrix,

$$E = \begin{bmatrix} X_{1,1}^B & X_{1,2}^B & \cdots & X_{1,d}^B \\ X_{2,1}^B & X_{2,2}^B & \cdots & X_{2,d}^B \\ \vdots & \vdots & \ddots & \vdots \\ X_{n,1}^B & X_{n,2}^B & \cdots & X_{n,d}^B \end{bmatrix}, \tag{32}$$

where $X^B$ denotes the vector of elite members of the predator in the population, $n$ is the population size, and $d$ stands for the search space dimension, equal to the number of variables in the problem.

After initializing the created population, excluding the elite members, the remaining individual members are constituted as the prey matrix. The predator will update its position according to that of the individuals in the prey matrix, and the best adapted solution is searched. The performance and efficiency of the whole population is improved. The prey matrix is represented as follows:

$$P = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,d} \end{bmatrix}, \tag{33}$$

where $X_{i,j}$ denotes the $j$ dimension of the $i$ prey.

*2. Optimizing phase.*

The MPA divides the optimization process into three main stages according to the speed ratio between the predator and the prey. In the first stage, also known as the exploratory one, the prey moves faster than the predator. In the second stage, from exploration to exploitation, the predator and the prey move at about the same speed. In the last stage, the exploitation one, the predator moves faster than the prey. The search efficiency is improved by assigning a fixed iteration period to each stage based on the movement patterns of the different stages.

The exploration phase occupies the first one-third of total iterations. Here the prey's velocity significantly exceeds the elite predator's, and the predator remains stationary while the prey takes a random Brownian motion. The mathematical model is expressed as follows:

while

$$T < \frac{1}{3}T_{\max}, i = 1, 2, \ldots, n,$$

we have

$$s_i = R_B \otimes (E_i - R_B \otimes P_i), \tag{34}$$

$$P_{i+1} = P_i + p \cdot R \otimes s_i, \tag{35}$$

where $T$ is the current and $T_{\max}$ the maximum number of iterations, $s_i$ denotes the individual motion steps of the top predator and prey, $R_B$ is a random vector of normally distributed Brownian motion, $E_i$ and $P_i$ denote the individual positions of the $i$ top predator and prey for the current iteration number, respectively, $\otimes$ is the sign of the term-by-term multiplication operation of the Kronecker product, $p$ denotes a constant taking the value 0.5, $R$ is a uniform random vector taking the value in the range [0,1], and $R_B \otimes P_i$ simulates the Brownian motion of the prey.

Besides, the trajectory of Brownian motion has the characteristics of discontinuity, non-smoothness and irregularity. In that motion, the step size of a particle is sampled from a normal distribution with a mean $(\mu)$ of zero and a variance $(\sigma^2)$ of 1. That is, the distance traveled by the particle at each time step is randomly generated via the probability density function of a normal distribution. The density function for this motion at point $x$ is defined as follows:

$$g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{36}$$

The transition phase occurs between one-third and two-thirds of total iterations. During this phase, prey and predator velocities are nearly identical. Consequently, the population divides equally: one half explores known space while the other half exploits new space. The prey performs Lévy motion for exploration, and the predator adopts Brownian motion for exploitation. The mathematical model is expressed as follows:

while

$$\frac{1}{3}T_{\max} < T < \frac{2}{3}T_{\max}, i = 1, 2, \ldots, \frac{n}{2},$$

we have

$$s_i = R_L \otimes (E_i - R_L \otimes P_i), \tag{37}$$

$$P_{i+1} = P_i + p \cdot R \otimes s_i, \tag{38}$$

and while

$$\frac{1}{3}T_{\max} < T < \frac{2}{3}T_{\max}, i = \frac{n}{2}, \ldots, n,$$

we have

$$s_i = R_B \otimes (R_B \otimes E_i - P_i), \tag{39}$$

$$P_{i+1} = P_i + p \cdot CF \otimes s_i, \tag{40}$$

$$CF = \left(1 - \frac{T}{T_{\max}}\right)^{\frac{2T}{T_{\max}}}, \tag{41}$$

where $R_L$ denotes the vector of random numbers generated by Lévy motion, $R_L \otimes P_i$ simulates the Lévy motion of the prey, and $CF$ is the adaptive parameter. The step size of the predator's motion is controlled by the adaptive parameters.

Unlike Brownian motion, the step size of Lévy motion is determined by a number of independent random variables, and a long-tailed distribution is possessed by these random variables. That is, a large number of extreme values are being present in the random variables. As a result, very large step sizes are generated in a given time period, and thus long-distance precision jumps are realized. This property allows the Lévy motion to be more efficient than the Brownian one when a globally optimal solution is sought. The method of generating random numbers based on Lévy motion is as follows:

$$R_L = 0.05 \times \frac{a}{|b|^{1/\beta}}, \tag{42}$$

where $R_L$ is a random number generated according to Lévy's motion, $\beta = 1.5$ is a constant, and $a$ and $b$ are two normally distributed variables. The standard deviations are $\sigma_a$ and $\sigma_b$ are denoted as

$$a = \text{Normal}\left(0, \sigma_a^2\right), \tag{43}$$

$$b = \text{Normal}\left(0, \sigma_b^2\right), \tag{44}$$

where $\sigma_b = 1$, and $\sigma_a$ is calculated as follows:

$$\sigma_a = \left[\frac{\Gamma\left(1+\beta\right) \cdot \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{(1+\beta)}{2}\right) \cdot \beta \cdot 2^{\frac{(\beta-1)}{2}}}\right]^{1/\beta}, \tag{45}$$

with $\Gamma$ being a Gamma function for integers $\beta$, $\Gamma\left(1+\beta\right) = \beta!$.

The exploitation phase, also known as the low velocity ratio one, usually occurs in the last one-third of the total iteration. In this phase, the prey's speed is much less than the predator's and at this point the strategy of Lévy movement is adopted by the predator. The mathematical model of this phase is represented as follows:

while

$$T > \frac{2}{3}T_{\max}, i = 1, 2, \ldots, n,$$

we have

$$s_i = R_L \otimes \left(R_L \otimes E_i - P_i\right), \tag{46}$$

$$P_{i+1} = E_i + p \cdot CF \otimes s_i, \tag{47}$$

where $R_L \otimes E_i$ simulates the Lévy movement of a predator. By increasing the step length of movement at the elite predator position, prey is sought more efficiently.

Specifically, when prey is concentrated locally, the predator employs Brownian motion for targeted search. When prey is dispersed, the predator utilizes Lévy motion to perform jump-based exploration. Additionally, predator behavior is influenced by environmental factors like eddy currents and fish aggregating devices (FADs), etc. This adaptive strategy tends to reduce local optima entrapment. The mathematical model is expressed as

$$P_{i+1} = \begin{cases} P_i + CF \cdot X_i \otimes U, & r < p_f, \\ P_i + \left[p_f\left(1-r\right) + r\right]\left(P_{r1} - P_{r2}\right), & r \geq p_f, \end{cases} \tag{48}$$

where $p_f = 0.2$ denotes the influence factor, $r$ is a uniform random number in the range [0,1], $U$ denotes an array of binary vectors containing 0 and 1, and $P_{r1}$ and $P_{r2}$ are two randomly selected prey members.

In the whole optimization process, the elite matrix and the prey matrix are known as crucial components. Both predators and prey are considered as individuals in the search space. The predator's own fitness is improved by searching for better adapted prey. The prey's own fitness is improved also by searching for its own food. At the end of each iteration, if a predator is replaced by a better adapted one, the replaced predator becomes the prey in the next generation of the population, and the new elite members will be updated with better adapted predators.

However, the following performance drawbacks are also possessed by the MPA:

- efficient and diverse initial populations cannot be generated;
- local optimal solutions cannot be jumped out quickly and may converge to unreasonable results, while the global optimal solution is difficult to be found quickly.

These drawbacks result in the Nash equilibrium solution of the UAV dynamic adversarial game being obtained with excessive time consumption and a huge number of iterations by the traditional marine predator algorithm, and such a solution may even fail to be identified entirely.

**3.2. Improved marine predator algorithm.** In order to address the shortcomings of the MPA described above, four improvements are proposed in this paper. As a result, the diversity of the population is increased, the algorithm can quickly get out of the local optimum solution, and its convergence speed is improved.

*1. Opposition-based learning.*

Opposition-based learning (OBL) is known as a new approach to machine intelligence. The method uses adversarial algorithms to optimize models (e.g., Özdemir *et al*., 2024; Meng *et al*., 2021; Liu *et al*., 2022a; Yu *et al*., 2022). This concept was originally introduced by Tizhoosh (2005). In the MPA, the search for the optimal solution is initiated by a randomly generated initial population. If the initial population is close to the optimal solution, the correct one can be found quickly. On the other hand, if the initial population is far from the optimal solution, the search may take longer, and even the optimal one may not be found. For a feasible initial population, adversarial learning can be used to find the corresponding solution at the opposite location. Its expression is denoted as

$$\acute{X}_i = X_{\max} + X_{\min} - X_i, \tag{49}$$

where $\acute{X}_i$ is the opposite position of $X_i$ in the spatially feasible solution, and $X_{\max}$ and $X_{\min}$ represent respectively the upper and lower limits of the variables in the problem. By the calculation of the fitness function,

if the opposite position is better, the previous population member is replaced.

To summarize, dyadic learning does not only rely on the initialization of a random population, but also uses known information and prior experience. The search situation is improved and accelerated. Therefore, opposition-based learning can circumvent the uncertainty caused by random populations to some extent, and the efficiency and accuracy of the optimization process are improved.

High-quality initial populations are rapidly screened out via simultaneous evaluation of the original solutions and their opposite counterparts by opposition-based learning initialization, which cuts down the number of iterations associated with the blind search in the early stage of the algorithm. Meanwhile, the population size is reduced by the adaptive population mechanism in the later iteration stage, with the search scope concentrated near the optimal solution. This decreases computational complexity while ensuring solution accuracy, thus satisfying the computing power requirements for decision-making in UAV dynamic adversarial games.

*2. Adaptive population regulation.*

Like many algorithms, the MPA begins by randomly generating an initial population. The population size significantly affects optimization efficiency and convergence speed. To balance computational efficiency with the avoidance of overfitting or slow convergence, an appropriate population size must be selected. This paper introduces an adaptive population method that dynamically adjusts the population size during iteration based on optimization progress, yielding improved performance. The initial population size is set as follows:

$$Q_{\text{size}} = 10 \times d, \tag{50}$$

where $d$ denotes the dimension of the problem.

Then the new population size is defined as

$$\bar{Q}_{\text{size}} = \max\left\{\text{round}\left(Q_{\text{size}} + \eta \cdot Q_{\text{size}}\right), d\right\}, \tag{51}$$

where 'round' is a rounding function, and $\eta$ denotes a random number in the range $[-0.5, 0.5]$. It can increase or decrease the current population size by half.

To handle varying population sizes across iterations, the algorithm employs an elite-based retention strategy. When the new population size increases, all individuals from the previous iteration are retained, and the additional positions are filled using elite members. This preserves high-quality solutions while enhancing diversity and global search capability. Conversely, if the new population size decreases, only the best-performing individuals are retained, eliminating poorly adapted members to maintain evolutionary efficiency. Furthermore, if the new population size is smaller than the problem dimension, it is automatically adjusted to match the dimension to ensure optimization stability and validity.

The population size can be dynamically adjusted by the adaptive population regulation mechanism according to the game stage. In the early stage of combat, the population is expanded to cover the full set of strategy combinations for multi-UAV systems (e.g., 16 pure strategies for both red and blue teams). This avoids the omission of high-payoff strategies that is associated with the traditional fixed population, and is particularly suitable for the characteristic that the strategy space expands dynamically with combat steps in multi-stage games.

*3. Improvement of step control parameters.*

In the iterative process of the MPA, the effectiveness of the search is governed by the step-size control parameter, which balances exploration and exploitation. A larger step size enhances global exploration, facilitating the discovery of optimal regions, while a smaller one improves local exploitation and accelerates convergence. To better coordinate these phases, this paper introduces a novel nonlinear step-size control parameter that dynamically adjusts based on the current search state. This allows more adaptive and flexible regulation, thereby enhancing both global exploration capability and local convergence performance. We have

$$CF = \left(1 - \sin\left(\frac{\pi}{2} \cdot \frac{T}{T_{\max}}\right)\right)^{\left(\frac{2T}{T_{\max}}\right)}, \tag{52}$$

where $CF$ is the new step factor control parameter, $T$ is the current iteration number, and $T_{\max}$ is the maximum iteration number.

With the nonlinear step factor control parameter, the relationship between exploration and development phases is better balanced by the improved MPA. At the same time, global search and local convergence capabilities are available, thus the optimization effect is improved.

The nonlinear step-size control enables real-time responses to changes in battlefield environments (e.g., sudden changes in enemy maneuver strategies). It dynamically balances the proportion between global exploration of new strategies and local optimization of selected strategies, thereby addressing the problem of the algorithm's search direction being prone to deviation in high-uncertainty scenarios.

*4. Inertia weighting coefficient.*

The MPA has fast search capability by predators and prey in the early and middle stages of optimization. Due to too fast an exploration, local optimal solutions are easily trapped in the algorithm. In order to improve the performance of the algorithm and increase the probability of successfully searching for the optimal solution, an

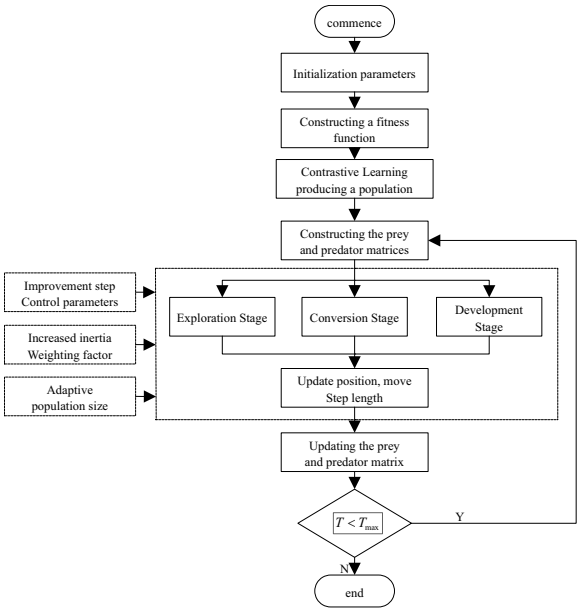



Fig. 2. Flowchart of the IMPA.

inertia weight coefficient is introduced by

$$BF = \left(\sin\left(\frac{\pi}{2}\cdot\frac{T}{T_{\max}}\right)\right)^{e^{\left(-\frac{T}{T_{\max}}\right)}}, \tag{53}$$

$$P_{i+1} = BF\cdot P_i + p\cdot R\otimes s_i\cdot\cdot, \tag{54}$$

$$P_{i+1} = BF\cdot E_i + p\cdot CF\otimes s_i, \tag{55}$$

where $BF$ is the inertia weight coefficient, $E_i$ and $P_i$ denote the $i$ top predator and prey individual position for the current iteration number, respectively, $s_i$ stands for the individual motion steps of the predator and the prey, $p$ denotes a constant taking the value 0.5, and $R$ is a uniform random vector taking the value in the range $[0, 1]$.

The inertia weight coefficient plays a key role in the early and middle stages of the MPA. By increasing the memory of previous search states, it helps the algorithm better track the location of the optimal solution and avoids over-exploration. By balancing the ratio of global and local search, the coefficient reduces the number of times the algorithm falls into local optimal solutions.

Building on this mechanism, inertial weight adjustment further preserves effective search experiences accumulated in the algorithm's initial phases and suppresses convergence oscillations caused by sudden battlefield disturbances. As a result, the computed Nash equilibrium solution achieves greater stability in multi-stage combat scenarios. Even when confronted with contingencies such as temporary tactical adjustments by the opponent, the stability of strategy payoffs can be maintained, effectively averting equilibrium solution failure.

Through four key enhancements including OBL initialization, adaptive population adjustment, nonlinear step-size control, and inertia weight adjustment, the IMPA significantly improves global search efficiency, avoids local optima, and strengthens both environmental adaptability and algorithmic robustness. Furthermore, the IMPA demonstrates superior applicability to multi-UAV dynamic confrontation games characterized by high environmental uncertainty, enabling faster and more accurate identification of the game's Nash equilibrium solution.

The synergistic operation of these four improved modules facilitates better coordination between individual UAV tactics and group objectives. When solving for Nash equilibrium, the algorithm simultaneously considers the survival probability of individual UAVs and optimizes the overall formation payoff. This integrated approach prevents the emergence of individually optimal but collectively inefficient strategy combinations, thereby fulfilling the core requirements of multi-UAV dynamic confrontation games.

**Algorithm 1.** IMPA solution.

**Step 1.** Initialize parameters. Set the maximum number of iterations $T_{\max}$.

**Step 2.** Construct the fitness function of the UAV confrontation game between the two sides and evaluate the fitness of each population individual via Eqn. (30).

**Step 3.** Population initialization process. Equation (49) is adopted to determine the initial population position.

**Step 4.** Construct the initial prey matrix, compute fitness values, and designate the prey with the best fitness value as the elite predator matrix.

**Step 5.** Equation (51) updates the population size, Eqn. (52) adjusts the movement step size, and Eqns. (54)–(55) update the prey position.

**Step 6.** The fitness values of each prey are recalculated, and the prey and elite predator matrices are updated.

**Step 7.** Determine if the maximum number of iterations has been reached. If so, the algorithm ends. Otherwise, proceed to Step 2.

**3.3. Steps and flowchart of the IMPA solution.** The flowchart of the IMPA for solving the Nash equilibrium solution of the game is shown in Fig. 2. The specific steps are expressed in Algorithm 1.

## 4. Simulation results and analysis

To verify the effectiveness of the proposed strategy, the following simulation experiments will be conducted. Let the red side have $N_r = 2$ UAVs as $U_1^r, U_2^r$, and the blue side have $M_b = 2$ UAVs as $U_1^b, U_2^b$. Each UAV on both sides carries two missiles, namely, $W_i^r = W_j^b = 2$,

Table 1. Angle, speed, altitude, and distance situation of Red and Blue UAVs.

| Red square | Blue square | $\theta$ [rad] | $v$ [km/h] | $H$ [km] | $d$ [km] |
|---|---|---|---|---|---|
| $U_1^r$ | $U_1^b$ | $(5\pi/9, 5\pi/18)$ | (300;200) | (4.55;5.28) | 2.6890 |
| $U_1^r$ | $U_2^b$ | $(5\pi/6, 5\pi/12)$ | (300;330) | (4.55;4.82) | 3.9595 |
| $U_2^r$ | $U_1^b$ | $(2\pi/3, 4\pi/9)$ | (240;200) | (5.46;5.28) | 4.0860 |
| $U_2^r$ | $U_2^b$ | $(19\pi/36, 2\pi/9)$ | (240;330) | (5.46;4.82) | 5.4121 |

Table 2. Red and Blue UAV combat strategies.

| Combat steps | Red UAV | Blue UAV |
|---|---|---|
| 1 | $U_1^r$ attack $U_1^b$, $U_2^r$ attack $U_1^b$ | $U_1^b$ attack $U_1^r$, $U_2^b$ attack $U_1^r$ |
| | $U_1^r$ attack $U_1^b$, $U_2^r$ attack $U_2^b$ | $U_1^b$ attack $U_1^r$, $U_2^b$ attack $U_2^r$ |
| | $U_1^r$ attack $U_2^b$, $U_2^r$ attack $U_1^b$ | $U_1^b$ attack $U_2^r$, $U_2^b$ attack $U_1^r$ |
| | $U_1^r$ attack $U_2^b$, $U_2^r$ attack $U_2^b$ | $U_1^b$ attack $U_2^r$, $U_2^b$ attack $U_2^r$ |
| 2 | The second stage strategy is consistent with the first stage. | |

$(i, j = 1, 2)$. The total number of combat steps for both sides is $T_{total} = 2$. In a single combat step, the number of missiles allocated to both UAVs is $w_i^r(T) = w_j^b(T) = 1$, $(T = 1, 2)$. The value of Red and Blue UAVs is $u_i^r = u_j^r = u_i^b = u_j^b = 20$. The probability of missile damage in an ideal combat environment for both sides is $Q_{1,1}^r = 0.75$, $Q_{1,2}^r = 0.4$, $Q_{2,1}^r = 0.6$, $Q_{2,2}^r = 0.65$, $Q_{1,1}^b = 0.25$, $Q_{1,2}^b = 0.75$, $Q_{2,1}^b = 0.5$, $Q_{2,2}^b = 0.35$. The impact factor is $\alpha_1 = \alpha_2 = 1$ in dynamic adversarial environments. The maximum range of both UAV missiles is $d^{\max} = 10$ km, and the minimum range is $d^{\min} = 1$ km. The situational weight coefficients are $\omega_1 = \omega_3 = 0.2$ and $\omega_2 = \omega_4 = 0.3$.

The initial situation of Red and Blue UAVs is shown in Table 1. Among them, $\theta$ [rad], $v$ [km/h], $H$ [km] and $d$ [km] represent respectively the azimuth, velocity, altitude, and distance of both drones, while $(5\pi/9, 5\pi/18)$ is the azimuth of Red $U_1^r$ relative to Blue $U_1^b$ as $5\pi/9$ and the azimuth of Blue $U_1^b$ relative to Red $U_1^r$ as $5\pi/18$. The other data in the table have similar meaning and will not be repeated. The combat strategies of the red and blue drones are shown in Table 2. The battle is divided into two steps, with 16 combat strategies for each side, and the payoff matrix is formed as $16 \times 16$. Equations (18)–(19) are used to calculate the specific profit function values for both parties. Because the red and blue sides are in a zero-sum game, the sum of their earnings is 0. Therefore, this article only lists the red side return matrix, as shown in Tables 3 and 4. The meanings of the numbers in the tables are as follows: taking the red strategy (1,2; 1,1) as an example, in the first battle round, red side $U_1^r$ chooses to attack blue side $U_1^b$ and red side $U_2^r$ chooses to attack blue side $U_2^b$; in the second battle round, red side $U_1^r$ attacks blue side $U_1^b$ and red side $U_2^r$ attacks blue side $U_1^b$. The value corresponding to the formed game situation is the profit value of the red side, when the respective combat strategies of both sides are selected.

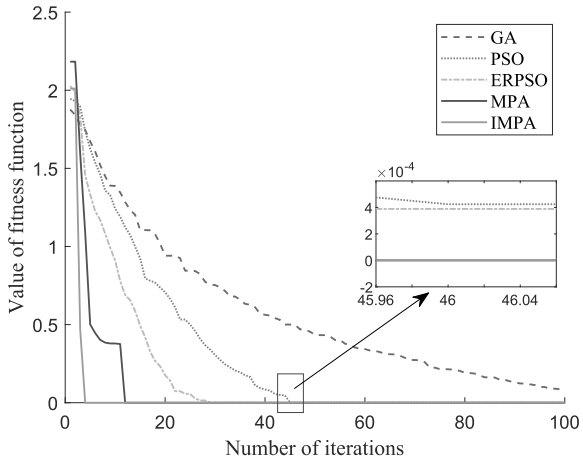



Fig. 3. Comparison of fitness function values.

**4.1. Algorithm performance analysis.** The genetic algorithm, particle swarm optimization, elite particle swarm optimization (ERPSO) (e.g., Wang *et al*., 2015), the MPA (e.g., Liu *et al*., 2021a) and the IMPA are used for comparative simulation experiments in this paper. To reduce the impact of random factors on simulation results, each algorithm is subjected to 100 optimization tests in the same simulation environment. The initial population size is set to 100 and the maximum number of iterations is 100. In the GA, the crossover probability is taken as 0.5 and the variance probability is taken as 0.1. In PSO, the particle inertia weights are taken as 0.6, the individual learning factor is taken as 0.5 and the population learning factor is taken as 0.5. In ERPSO, the number of individuals with clonal mutations is 20, and the elite particle re-selection threshold is 0.1. In the MPA, the probability of being affected by factors such as FADs is $p_f = 0.2$.

A comparison of the fitness function values for the five algorithms is presented in Fig. 3. From the figure,

Table 3. Revenue matrix for the red square UAV (Part 1).

| Blue strategy | | Red strategy | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| | | 1,1;1,1 | 1,1;1,2 | 1,1;2,1 | 1,1;2,2 | 1,2;1,1 | 1,2;1,2 | 1,2;2,1 | 1,2;2,2 |
| 1 | 1,1;1,1 | 3.6577 | 2.7264 | 1.5237 | 1.0671 | 3.1634 | 2.4553 | 1.4986 | 1.1627 |
| 2 | 1,1;1,2 | 5.2841 | 4.9126 | 4.1095 | 3.9036 | 4.3078 | 4.0012 | 3.2548 | 3.1210 |
| 3 | 1,1;2,1 | 4.4502 | 3.8911 | 2.9009 | 2.6142 | 3.6605 | 3.2359 | 2.3761 | 2.1970 |
| 4 | 1,1;2,2 | 5.2198 | 5.1796 | 4.4243 | 4.4925 | 4.0511 | 4.0745 | 3.3014 | 3.4535 |
| 5 | 1,2;1,1 | 4.2481 | 3.9232 | 3.0894 | 2.9344 | 4.8961 | 4.6281 | 3.8902 | 3.7685 |
| 6 | 1,2;1,2 | 4.6348 | 4.7340 | 4.0645 | 4.2434 | 4.9931 | 5.1506 | 4.4744 | 4.7118 |
| 7 | 1,2;2,1 | 4.1687 | 4.1559 | 3.3800 | 3.4823 | 4.5991 | 4.6662 | 3.9116 | 4.0871 |
| 8 | 1,2;2,2 | 3.8098 | 4.2308 | 3.4933 | 4.0071 | 3.7059 | 4.3112 | 3.4554 | 4.1738 |
| 9 | 2,1;1,1 | 3.4645 | 2.9495 | 1.9303 | 1.6918 | 3.8972 | 3.4920 | 2.6229 | 2.4478 |
| 10 | 2,1;1,2 | 4.2215 | 4.2063 | 3.4321 | 3.5316 | 4.2684 | 4.3224 | 3.5566 | 3.7276 |
| 11 | 2,1;2,1 | 3.6100 | 3.4515 | 2.5386 | 2.5547 | 3.7702 | 3.7187 | 2.8587 | 2.9692 |
| 12 | 2,1;2,2 | 3.5338 | 3.8664 | 3.0526 | 3.4961 | 3.1198 | 3.6326 | 2.7086 | 3.3559 |
| 13 | 2,2;1,1 | 1.7126 | 1.7657 | 0.9335 | 1.1292 | 4.0383 | 4.1662 | 3.3733 | 3.6151 |
| 14 | 2,2;1,2 | 1.4621 | 1.8619 | 1.1184 | 1.6420 | 3.1061 | 3.7535 | 2.8737 | 3.6484 |
| 15 | 2,2;2,1 | 1.1247 | 1.4403 | 0.6175 | 1.0759 | 2.8350 | 3.4039 | 2.4657 | 3.1765 |
| 16 | 2,2;2,2 | 0.2904 | 0.9341 | 0.1127 | 0.9366 | 0.5728 | 1.7976 | 0.5591 | 2.0775 |

Table 4. Revenue matrix for the red square UAV (Part 2).

| Blue strategy | | Red strategy | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| | | 2,1;1,1 | 2,1;1,2 | 2,1;2,1 | 2,1;2,2 | 2,2;1,1 | 2,2;1,2 | 2,2;2,1 | 2,2;2,2 |
| 1 | 1,1;1,1 | 1.6833 | 1.2215 | 0.5084 | 0.3140 | 1.6862 | 1.3439 | 0.7726 | 0.6400 |
| 2 | 1,1;1,2 | 2.3784 | 2.1728 | 1.5361 | 1.4918 | 2.0999 | 1.9747 | 1.4136 | 1.4324 |
| 3 | 1,1;2,1 | 1.9188 | 1.6509 | 0.9571 | 0.8915 | 1.7466 | 1.5802 | 0.9857 | 0.9898 |
| 4 | 1,1;2,2 | 2.0804 | 2.1305 | 1.4376 | 1.6218 | 1.5812 | 1.7371 | 1.0663 | 1.3535 |
| 5 | 1,2;1,1 | 3.4130 | 3.2434 | 2.5770 | 2.5355 | 4.3320 | 4.1857 | 3.6030 | 3.5663 |
| 6 | 1,2;1,2 | 3.4371 | 3.5970 | 2.9607 | 3.2024 | 4.1168 | 4.3314 | 3.7035 | 3.9955 |
| 7 | 1,2;2,1 | 3.1001 | 3.1943 | 2.5012 | 2.6990 | 3.8375 | 3.9963 | 3.3284 | 3.5817 |
| 8 | 1,2;2,2 | 2.4167 | 2.9119 | 2.1453 | 2.7458 | 2.5647 | 3.2716 | 2.3873 | 3.2188 |
| 9 | 2,1;1,1 | 2.2470 | 1.9942 | 1.2531 | 1.1880 | 3.0524 | 2.8537 | 2.2259 | 2.1824 |
| 10 | 2,1;1,2 | 2.4352 | 2.5168 | 1.8146 | 2.0130 | 2.9499 | 3.1027 | 2.4288 | 2.6907 |
| 11 | 2,1;2,1 | 2.0373 | 2.0504 | 1.2885 | 1.4494 | 2.6326 | 2.7315 | 2.0193 | 2.2498 |
| 12 | 2,1;2,2 | 1.5142 | 1.9349 | 1.1101 | 1.6604 | 1.4964 | 2.1339 | 1.2107 | 1.9953 |
| 13 | 2,2;1,1 | 1.5387 | 1.7133 | 0.9571 | 1.2530 | 4.4925 | 4.6488 | 3.9783 | 4.2220 |
| 14 | 2,2;1,2 | 0.9223 | 1.4300 | 0.6539 | 1.2916 | 3.2247 | 3.9414 | 3.0673 | 3.9016 |
| 15 | 2,2;2,1 | 0.6638 | 1.1131 | 0.2877 | 0.8797 | 3.0351 | 3.6902 | 2.7851 | 3.5665 |
| 16 | 2,2;2,2 | -0.625 | 0.1855 | -0.748 | 0.2770 | -0.078 | 1.4307 | 0.0043 | 1.8619 |

it is evident that the GA only reaches its result when the maximum iteration limit (100 generations) is attained. The curve is still found to have a decreasing trend and does not converge completely. A localized zoom of Fig. 3 reveals that the PSO algorithm approximately converges to the global optimal solution at generation 46, achieving a fitness value of 4.38E-04. The ERPSO algorithm reaches the lower bound of convergence at 30 generations with an adaptation value of 3.92E-04. The convergence speed and computational accuracy are better than those of PSO and the GA. In this simulation experiment, the fitness function value sought by both the MPA and IMPA is 0 and the computational accuracy is optimal. However, the MPA converges completely only in the 13th generation, while the IMPA converges to 0 when the 5th generation is reached and does so faster.

The mixed-strategy error that defines the adversarial game between the two UAVs is defined as follows:

$$e_p(T) = \|Z_i(T) - E_S\| = \sqrt{\sum_{j=1}^{m+n} (Z_{ij}(T) - E_{Sj})^2}, \quad (56)$$

where $Z_i(T) = (x_{i1}, \cdots, x_{im}, y_{i1}, \cdots, y_{in})$ is the mixed strategy of Red and Blue corresponding to the optimal fitness value at iteration $T$, $m = 16$ is the number of pure strategies on the red side and $n = 16$ is the number of pure strategies on the blue side, while $E_S$ denotes the two-party mixed strategy Nash equilibrium solution. The error comparison graphs of the five algorithms are shown in Fig. 4. In relation to the other algorithms, the IMPA has the highest accuracy with an error of 0 at generation 6, and the fastest convergence rate is found.

The fitness values and errors of the five algorithms are mainly analyzed by the above. The improved ocean predator algorithm proposed in this paper is embodied to have the highest solving accuracy and the lowest error. In addition, the simulation time and the ability of the algorithms to solve for the Nash equilibrium strategy are also analyzed in this paper. The performance comparison results of the five algorithms in 100 simulation tests are given in Table 5. The comparison parameters are denoted as the average fitness function value, the best fitness function value, the average error, the average simulation time, and the number of times when the error is greater than 0.0001.

As can be seen from Table 5, the GA has the longest average simulation time with a solution probability of 65%. The remaining four algorithms have similar simulation times, so only the solution probabilities are compared: 83% for PSO, 95% for ERPSO, 87% for the MPA, and 100% for the IMPA. In summary, the overall performance of the IMPA is better than that of the other four algorithms. The UAV air combat game Nash equilibrium problem is solved more applicable.

**4.2. Verification of the Nash equilibrium condition.** The IMPA is employed to solve for the Nash equilibrium mixed strategies of both UAVs. The blue strategy is $y^* = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.9576, 0.0424, 0, 0, 0, 0]$, and the red strategy is $x^* = [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$. Then Red's 6th strategy (1,2;1,2) is chosen with probability 1, Blue's 11th strategy (2,1;2,1) is chosen with probability 0.9576, and the 12th strategy (2,1;2,2) is chosen with probability 0.0424. According to the Nash equilibrium strategy, the red-side payoff is $x^* F_r y^{*T} = 2.9709$ and the blue-side payoff is computed as $x^* F_b y^{*T} = -2.9709$.

In order for the correctness of the required Nash equilibrium mixed strategies to be verified, this section randomly generates 200 sets of mixed strategies $([x]_{200\times16}, [y]_{200\times16})$ for Red and Blue. The returns $xF_r y^{*T}$ and $x^* F_b y^T$ are computed when Red and Blue, respectively, randomly change their strategies. The results of the returns for both sides are shown in Figs. 5 and 6. Those figures demonstrate that, when the blue

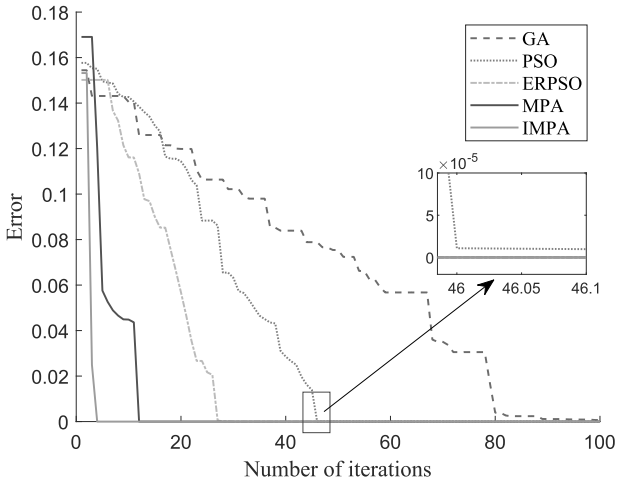



Fig. 4. Comparison of fitness function values.

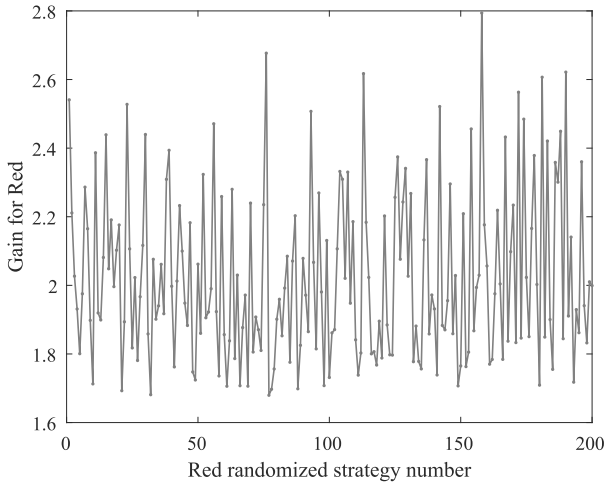



Fig. 5. Benefits of Red's randomized strategy.

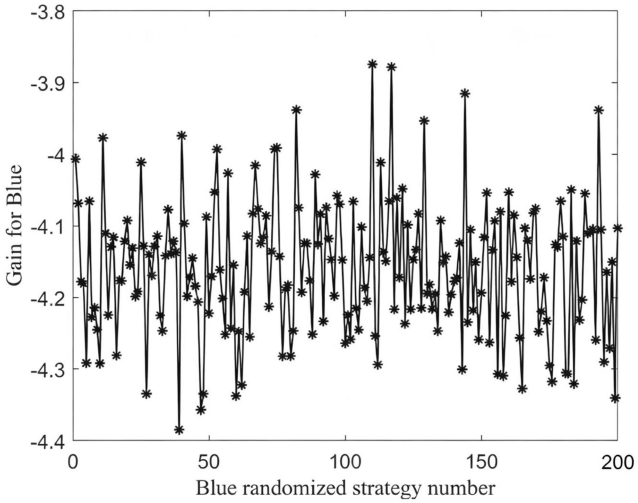



Fig. 6. Benefits of the Blue's randomized strategy.

Table 5. Algorithm performance comparison.

| Algorithm | Average fitness | Optimal fitness | Average error | Simulation time | Unsolved count |
|---|---|---|---|---|---|
| GA | 0.0955 | 0.0086 | 0.0687 | 1.6392 | 35 |
| PSO | 5.78E-04 | 4.24E-05 | 1.09E-04 | 0.4936 | 17 |
| ERPSO | 4.51E-04 | 3.88E-05 | 1.38E-04 | 0.5623 | 5 |
| MPA | 2.69E-05 | 2.16E-09 | 4.49E-06 | 0.4801 | 13 |
| IMPA | 1.59E-11 | 0 | 6.76E-12 | 0.5431 | 0 |

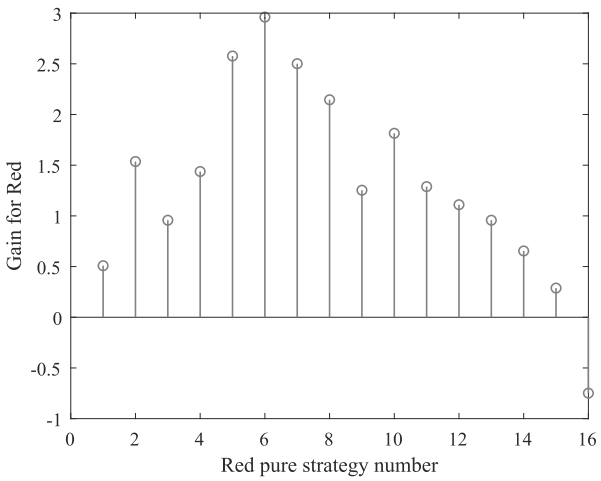



Fig. 7. Red pure strategy returns.

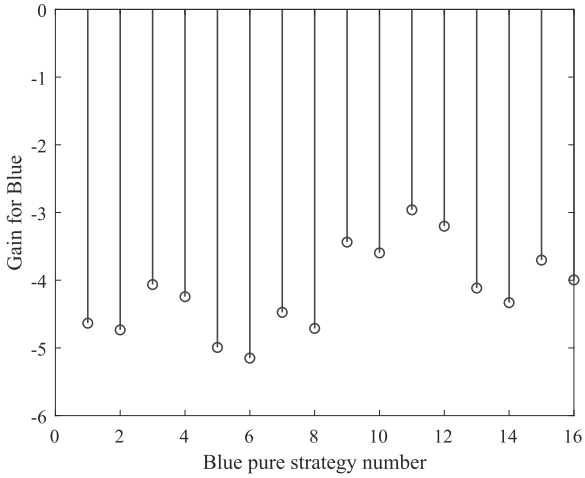



Fig. 8. Blue pure strategy returns.

side adopts the Nash equilibrium mixed strategy, any randomly chosen alternative strategy by Red yields a payoff below 2.9709. Conversely, when Red employs the Nash equilibrium strategy, all deviations by blue result in payoffs lower than $-2.9709$. These experimental results confirm that neither payer can unilaterally improve their payoff by changing strategies. The data thus satisfies the Nash equilibrium condition for this dynamic adversarial game.

**4.3. Approximate pure strategy verification.** To further validate the effectiveness of the proposed Nash equilibrium strategies, the approximate pure strategy method is employed (Fu *et al.*, 2022). Using the IMPA, the red side was found to select its 6th strategy with a probability of 1, while the blue side selects its 11th strategy with a probability of 0.9576. These results were approximated as near-pure strategies. The payoffs under these strategies were then compared with those of other pure strategy scenarios to assess performance.

The gains for Red and Blue under pure strategies are shown in Figs. 7 and 8, respectively. When Red adopts pure strategy 6, Blue's gain is maximized only with pure strategy 11. Conversely, when Blue uses pure strategy 11, Red achieves maximum gain with pure strategy 6. Simulation results confirm that the strategy profile Red (1,2;1,2) and Blue (2,1;2,1) forms the optimal Nash equilibrium, further verifying the effectiveness of the proposed strategy.

## 5. Conclusion

This paper investigated the multi-UAV dynamic adversarial game problem. A game-theoretic model of UAV confrontation was established by incorporating real-time air combat situation awareness and survival probability assessments for both red and blue agents. To solve for the Nash equilibrium of the formulated model, the IMPA was proposed. The algorithm integrates several key enhancements: opposition-based learning for population initialization, adaptive population adjustment, nonlinear step-size control, and an inertia weight mechanism. These innovations collectively improve the initial solution quality, increase population diversity, accelerate convergence, and enhance the algorithm's ability to avoid local optima. Simulation results demonstrate the effectiveness and superiority of the proposed IMPA over existing algorithms, confirming its suitability for solving UAV air combat decision-making problems.

## Acknowledgment

The authors gratefully acknowledge all those who contributed to this work through their guidance, support, and insightful discussions, and declare no conflict of interest.

## References

Duan, H., Li, P. and Yu, Y. (2015). A predator-prey particle swarm optimization approach to multiple UCAV air combat modeled by dynamic game theory, *IEEE/CAA Journal of Automatica Sinica* **2**(1): 11–18.

Fan, Q., Huang, H., Chen, Q., Yao, L., Yang, K. and Huang, D. (2022). A modified self-adaptive marine predators algorithm: Framework and engineering applications, *Engineering with Computers* **38**(4): 3269–3294.

Faramarzi, A., Heidarinejad, M., Mirjalili, S., and Gandomi, A.H. (2020). Marine predators algorithm: A nature-inspired metaheuristic, *Expert Systems with Applications* **152**: 113377, DOI: 10.1016/j.eswa.2020.113377.

Fu, X., Liu, S.L., Guan, Z.F. and Liu, H. (2023). Phased-improvement marine predators algorithm and its application, *Control and Decision* **38**(4): 902–910.

Fu, X., Zhu, J., Wei, Z., Wang, H. and Li, S. (2022). A UAV pursuit-evasion strategy based on DDPG and imitation learning, *International Journal of Aerospace Engineering* **2022**(1): 3139610.

Gong, C., Zhou, N., Xia, S. and Huang, S. (2024). Quantum particle swarm optimization algorithm based on diversity migration strategy, *Future Generation Computer Systems* **157**: 445–458.

Jiang, G.-s., Shi, X.-m., Chen, J., Zhao, M. and Liu, H.-b. (2022). Dynamic weapon target assignment based on incomplete information game, *Command Control and Simulation* **44**(3): 20–24.

Kantue, P. and Pedro, J.O. (2022). Integrated fault-tolerant control of a quadcopter UAV with incipient actuator faults, *International Journal of Applied Mathematics and Computer Science* **32**(4): 607–617, DOI: 10.34768/amcs-2022-0042.

Kim, J., Oh, H., Yu, B. and Kim, S. (2021). Optimal task assignment for UAV swarm operations in hostile environments, *International Journal of Aeronautical and Space Sciences* **22**(2): 456–467.

Li, S., Ding, Y. and Gao, Z. (2019). UAV air combat maneuvering decision based on intuitionistic fuzzy game theory, *Systems Engineering and Electronics* **41**(5): 1063–1070.

Li, Y., Li, J., Liu, C., Li, J., Xin, Z. and Chen, Z. (2022). An auction-based attack-defense decision-making method for UAV air combat, *2022 IEEE International Conference on Unmanned Systems (ICUS), Guangzhou, China*, pp. 902–909.

Liu, C., Sun, S., Tao, C., Shou, Y. and Xu, B. (2021a). Sliding mode control of multi-agent system with application to UAV air combat, *Computers & Electrical Engineering* **96**(A): 107491, DOI: 10.1016/j.compeleceng.2021.107491.

Liu, L., Zhang, S., Zhang, L., Pan, G. and Bai, C. (2021b). Multi-AUV dynamic maneuver decision-making based on intuitionistic fuzzy counter-game and fractional-order particle swarm optimization, *Fractals* **29**(08): 2140039.

Liu, F., Dong, X., Yu, J., Hua, Y., Li, Q. and Ren, Z. (2022a). Distributed Nash equilibrium seeking of $n$-coalition noncooperative games with application to UAV swarms, *IEEE Transactions on Network Science and Engineering* **9**(4): 2392–2405.

Liu, H., Wang, Y., Chen, M. and Zhang, Y. (2022b). UAV air combat game based on iteration method, *Electronics Optics & Control* **29**(2): 1–6.

Liu, J., Wu, Q., Wang, Y. and Zhou, D. (2022c). UAV game decision based on quantum particle swarm optimization, *Fire Control and Command Control* **47**(09): 73–78+84.

Ma, C., Zeng, G.H., Huang, B. and Liu, J. (2022). Marine predator algorithm based on chaotic opposition learning and group learning, *Computer Engineering and Applications* **58**: 271–283.

Meng, Y., He, J., Luo, S., Tao, S. and Xu, J. (2021). An improved predator-prey particle swarm optimization algorithm for nash equilibrium solution, *PLOS One* **16**(11): e0260231.

Merheb, A.-R., Noura, H. and Bateman, F. (2015). Design of passive fault-tolerant controllers of a quadrotor based on sliding mode theory, *International Journal of Applied Mathematics and Computer Science* **25**(3): 561–576, DOI: 10.1515/amcs-2015-0042.

Oszust, M. (2021). Enhanced marine predators algorithm with local escaping operator for global optimization, *Knowledge-Based Systems* **232**: 107467, DOI: 10.1016/j.knosys.2021.107467.

Özdemir, R., Taşyürek, M. and Aslantaş, V. (2024). Improved marine predators algorithm and extreme gradient boosting (XGBoost) for shipment status time prediction, *Knowledge-Based Systems* **294**: 111775, DOI: 10.1016/j.knosys.2024.111775.

Ramezani, M., Bahmanyar, D. and Razmjooy, N. (2021). A new improved model of marine predator algorithm for optimization problems, *Arabian Journal for Science and Engineering* **46**(9): 8803–8826.

Salazar, J.C., Sanjuan Gómez, A., Nejjari Akhi-Elarab, F. and Sarrate Estruch, R. (2020). Health-aware and fault-tolerant control of an octorotor UAV system based on actuator reliability, *International Journal of Applied Mathematics and Computer Science* **30**(1): 47–59, DOI: 10.34768/amcs-2020-0004.

Song, Y., Cai, X., Zhou, X. Zhang, Bin, C., Huiling, L., Yuangang, D., Wuquan, D., Wu (2023). Dynamic hybrid mechanism-based differential evolution algorithm and its application, *Expert Systems with Applications* **213**: 118834, Part: A, DOI: 10.1016/j.eswa.2022.118834.

Tizhoosh, H.R. (2005). Opposition-based learning: A new scheme for machine intelligence, *International Conference on Computational Intelligence for Modelling, Control and Automation/International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA–IAWTIC'06), Austria,* Vol. 1, Vienna, pp. 695–701.

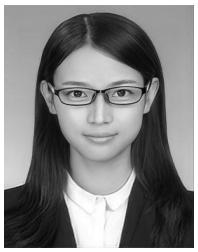
Wang, Y., Zhang, W., Fu, L., Huang, D. and Li, Y. (2015). Nash equilibrium strategies approach for aerial combat based on elite reelection particle swarm optimization, *Control Theory and Applications* **32**(7): 857–865.

Xu, J., Deng, Z., Song, Q., Chi, Q., Wu, T., Huang, Y., Liu, D. and Gao, M. (2020). Multi-UAV counter-game model based on uncertain information, *Applied Mathematics and Computation* **366**: 124684, DOI: 10.1016/j.amc.2019.124684.

Yu, Y., Liu, J. and Wei, C. (2022). Hawk and pigeon's intelligence for UAV swarm dynamic combat game via competitive learning pigeon-inspired optimization, *Science China Technological Sciences* **65**(5): 1072–1086.

Zhang, Q., Zeng, Q., Zhang, Z. and Ye, U.X.Y. (2022). Research on weapon-target assignment problem for marine predator algorithm, *Journal of Ordnance Equipment Engineering* **43**(8): 158–163.

Zhao, Y.L., Song, Y.X., Zhang, J.J. and Kang, L.W. (2019). Fuzzy game decision-making of unmanned aerial vehicles air-to-ground attack based on the particle swarm optimization integrating multiply strategies, *Control Theory and Applications* **36**(10): 1644–1652.



**Nannan Cheng** holds a Bachelor's degree from Beijing Information Science & Technology University and a Master's degree from Beihang University. She also holds a PhD degree in automatic science and electrical engineering from Beihang University. She has long been engaged in research in deep learning, industrial intelligence, multi-modal information processing, and data modeling. Through this work, she has accumulated extensive experience and achieved significant research results.

**Qing Li** is a professor at the University of Science and Technology Beijing. He holds several esteemed positions, including that of an expert for the Engineering Education Professional Certification Association of China in Electronics, Information, and Electrical Engineering, and a standing committee member of the Artificial Intelligence and Robotics Education Professional Committee of the China Association of Education Development Strategy. He has long been engaged in research on intelligent information processing and intelligent optimization theory, and their applications.

**Heng Wang** received his Bachelor's degree from the School of Information Science and Engineering at Northeastern University in 2003 and his PhD from the same school in 2008. He is currently a professor at the School of Automation, University of Science and Technology Beijing. His main research interests include simultaneous localization and mapping (SLAM), as well as navigation and control for unmanned vehicles.

**Saisai Tong** received his Master's degree in mechanical engineering from Beihang University, Beijing, China, in 2020. His research interests include compressors, hydraulic servo control, smart material actuators, and nonlinear control.

**Xingjian Fu** received his PhD degree from the School of Automation, University of Science and Technology Beijing, China, in 2005. Currently, he is a professor at Beijing Information Science and Technology University. He is a committee member of the Mechanical Industry Automation Branch of the China Mechanical Engineering Society, as well as a member of the China Automation Society and of the Chinese Association for Artificial Intelligence. His research interests include intelligent control, robust fault-tolerant control and motion body control.

Received: 13 October 2025
Revised: 10 February 2026
Accepted: 25 March 2026