scientific reports

OPEN

Observation optimization for marine target tracking by airborne and maritime unmanned platforms cooperation using adaptive enhanced dung beetle optimization

Qiuyang Dai[1]✉, Faxing Lu[1], Haoran Shi[2], Junfei Xu[1] & Jianjing Qu[1]

Cooperative observation optimization for maritime targets is crucial for improving marine monitoring precision. Existing research predominantly focuses on homogeneous platform cooperative observation optimization under random error influences, while neglecting the collaborative optimization challenges of heterogeneous platforms affected by systematic errors. To address these limitations, this paper proposes a heterogeneous unmanned platform cooperative observation optimization method based on Adaptive Enhanced Dung Beetle Optimization (AEDBO) algorithm. First, we derive optimal observation configurations for the aerial unmanned platform, maritime cooperative unmanned platform, and targets under azimuth systematic error impacts using an attitude correction algorithm. Subsequently, we design AEDBO by integrating an improved Tent chaotic mapping and centroid opposition-based learning strategy to enhance population diversity. An adaptive convergence factor and nonlinear ball-rolling dung beetle population decline model are introduced to balance global exploration and local exploitation capabilities, while Cauchy-Gaussian mutation strategies are employed to prevent premature convergence. Finally, AEDBO is applied to aerial unmanned platform trajectory optimization. Experimental results demonstrate that compared with DBO, AEDBO achieves improved optimization accuracy across 15 benchmark functions. In both safe and hazardous zone scenarios, optimized trajectories reduce target tracking errors to below 10 m, with optimal observation configurations validated through practical experiments. This study establishes a novel theoretical framework and optimization toolkit for heterogeneous unmanned platform cooperative observation.

Keywords Unmanned platform, Cooperative tracking, Trajectory optimization, Error reduction, Dung beetle optimization, Systematic error

Unmanned platforms offer advantages of low cost, high convenience, and strong concealment, enabling efficient work and reduced personnel risk. They have wide applications across aerospace, maritime exploration and environmental monitoring[1]. Among them, path tracking[2], trajectory tracking[3,4] and target tracking[5] are the core technologies for autonomous navigation and mission planning of unmanned platforms, and there are significant differences in the technical connotations of the three. Path tracking emphasizes time-constraint-free tracking of predefined geometric paths, trajectory tracking requires synergistic control of paths and temporal parameters, and target tracking focuses on real-time target state estimation. In this paper, we take target tracking as an entry point and focus on the trajectory optimization problem of unmanned platforms.

Methods to improve target tracking accuracy are divided into two main categories: the first is data processing optimization. Error in airborne unmanned platform marine target observation stems from sensor detection and attitude errors, which are generally composed of random error and systematic error[6]. Random error can be filtered out using algorithms like Kalman Filter (KF), Extended Kalman Filter (EKF) and Cubature Kalman Filter (CKF)[7–9]. Systematic error requires error reduction algorithms including real-time quality control, least squares[10] , maximum likelihood[11] , exact maximum likelihood estimation[12] and other algorithms. In addition,

[1]School of Ordnance Engineering, Naval University of Engineering, Wuhan 430033, China. [2]National Key Laboratory of Electromagnetic Energy, Naval University of Engineering, Wuhan 430033, China. ✉email: z21182605@nue.edu.cn

some researchers[13] proposed a target localization method using auxiliary beacons. The unmanned platform observes the auxiliary beacons and the target simultaneously to reduce systematic error.

The second is observation geometry optimization. The positional relationship between the platforms and the target affects tracking accuracy. This is referred to as the optimal observation configuration for an unmanned platform, determined by target observation. Fisher Information Matrix (FIM) is commonly used to analyze optimal observation configurations, which can be divided into two categories: A-optimal and D-optimal. A-optimal aims to minimize the inverse trace of the FIM, while D-optimal seeks to maximize the determinant of the FIM.SHI et al.[14] extended the optimal observation configuration for UAVs based on distance and angle measurements to 3D space. Hung et al.[15] proposed a multi-UAV system using heterogeneous sensors for tracking, with a cost function based on the D-optimal criterion for the problem of unknown target tracking. Chen et al.[16] focused on sensor deployment optimization for arrival time direct localization in shallow water multipath environments, considering multipath signal loss and angle and distance constraints. They proposed an optimization problem based on the A-optimal criterion to determine the optimal sensor location. Wu et al.[17] proposed a sensor deployment optimization method for the simultaneous time-of-arrival based multi-objective localization problem using A-optimal criterion, combining alternating minimization, the multiplier alternating direction method and maximizing minimization. However, most existing optimal observation configuration analysis based on FIM only considers the effect of observation random error on target tracking, with few considering the effect of observation systematic error.

Cooperative observation optimization is essentially a trajectory optimization problem, belonging to the control methods of airborne unmanned platforms. Common trajectory optimization methods for unmanned airborne platforms include optimal control[18], guidance[19] and intelligent algorithms[20]. Optimal control is a control strategy that optimizes the system performance index, and commonly uses dynamic programming, model predictive control and reinforcement learning. Literature[21]. proposed an end-to-end artificial intelligence framework to deal with the trajectory optimization problem for airborne unmanned platforms, significantly improving performance and efficiency. Literature[22] proposed a dual-deep Q-network based trajectory optimization method for airborne unmanned platforms considering reconfigurable intelligent surfaces. Literature[23] used a distributed model predictive control method to optimize the trajectory of an airborne unmanned platform in a dynamic environment. Although optimal control can achieve optimality and high accuracy, it has high computational complexity, strict model constraints, sensitivity to initial conditions, poor real-time performance and difficulty in dealing with complex constraints. Lyapunov potential field guidance method controls an airborne unmanned platform by forming a potential field around the guidance path, which has low computational complexity and real-time potential field changes, for rapid control and guidance. Literature[24] controlled the airborne unmanned platform heading to complete tracking of straight lines, arcs and arbitrary paths based on the vector field. Literature[25] designed the outer loop controller of the airborne unmanned platform to generate the required heading rate command via vector field . Although the guidance method has good real-time performance, it requires designing the guidance law based on the known optimal observation configuration. It cannot solve the trajectory optimization problem under complex unknown conditions. Compared to the guidance method, intelligent algorithms do not require an exact mathematical model. They can be solved by evaluating the objective function value, suitable for problems lacking accurate models. Intelligent algorithms have strong versatility and adaptability, enabling effective solutions to complex practical problems. Intelligent algorithms are widely used in unmanned platforms. Literature[26] applied swarm intelligence algorithms to search and rescue scenarios of airborne unmanned platforms. Literature[27] utilized a distributed control framework based on homing pigeon hierarchical strategy to address airborne unmanned platform swarming problems. Swarm intelligence algorithms are also widely used in trajectory optimization problems. Literature[28] combined adaptive differential evolution with Nash optimization for trajectory optimization, improving target tracking accuracy and reducing complexity. The results showed that the new method performed better in real target trajectory tracking, airborne unmanned platform safety, and stability. Literature[29] proposed an improved bat algorithm to optimize tracking trajectory of an airborne unmanned platform for dynamic intrusion target tracking in oilfield inspection. Literature[30] proposed a new multi-objective trajectory optimization algorithm using a cut-and-fill coding strategy to optimize populations where individuals may have different lengths.

DBO[31] is a new swarm intelligence algorithm proposed by XUE et al. in 2022. Compared to PSO[32] , GWO[33], SSA[34], SCA[35], MVO[36] and HHO[37], results proved the superiority of its performance. Once proposed, DBO had wide applications across various fields. Literature[38] used DBO to determine optimal feature color combination for olfactory sensors to help determine fatty acids in stored wheat. Literature[39] combined DBO with KELM (Extreme Learning Machine) to improve accuracy of recognizing internal leaks in unilateral and bilateral ball valves. Like other swarm intelligence algorithms. DBO also easily falls into local optima and has weak global exploration ability. To fully develop its performance, Literature[40] improved IDBO to select hyperparameters for BiLSTM[41–43] DBO algorithm is optimized in terms of population initialization, search strategy, and population diversity, effectively improving DBO performance. Existing studies have not used DBO for trajectory optimization of airborne unmanned platforms.

In summary, current research on optimizing unmanned platform cooperative observation of marine targets has two deficiencies: one is that the current observation optimization problem focuses on homogeneous platforms; the other is that previous research on trajectory optimization mainly considers the influence of random error, and rarely considers the influence of systematic error on tracking accuracy. In this paper, we address these problems, introduce the orientation system error of sensor observation, collaborate airborne unmanned platforms with sea unmanned platforms, and study the optimization method of multi-unmanned platforms collaborative observation of maritime targets.

The main contributions of this paper compared to previous studies are as follows:

(1) Based on the error reduction algorithm, first-order Taylor expansion of error is used to analyze azimuthal systematic error's impact on tracking error. Optimal observation configurations for the airborne unmanned platform, maritime cooperative unmanned platform, and target are derived;
(2) AEDBO is proposed using improved Tent chaos mapping and center-of-mass contrastive inverse learning for population initialization, a new adaptive convergence factor, an adaptive nonlinear ball-rolling dung beetle tree-volume decreasing model, and an adaptive Cauchy-Gaussian mutation strategy.
(3) Taking airborne and maritime unmanned platforms as cooperative objects, AEDBO is used to optimize the flight trajectory of airborne unmanned platforms in safe and hazardous area environments to obtain the optimal observation configurations of the airborne unmanned platform, maritime cooperative unmanned platform and target, thereby improving target tracking accuracy under a receding horizon optimization framework.

The paper is organized as follows. Section II describes and models the problem. Section III analyzes the trajectory optimization of the airborne unmanned platform, based on the error reduction algorithm to analyze the optimal observation configuration under azimuthal system error. Section IV proposes an optimal control method for airborne unmanned platforms based on AEDBO, which consists of four improved methods: population initialization incorporating improved Tent chaos mapping and center-of-mass dyadic inverse learning, adaptive convergence factor, adaptive nonlinear rolling dung beetle decreasing number model, and adaptive Cauchy-Gaussian mutation strategy with complexity analysis. AEDBO solving steps for the optimal control method of airborne unmanned platforms are given. Section V conducts simulation experiment verification. The performance analysis of AEDBO is carried out firstly, and then the optimal trajectory optimization of the airborne unmanned platform in different environments is carried out using AEDBO. Section VI concludes with future outlook.

Problem description and models

Problem description

The marine target localization and tracking scenario for an airborne unmanned platform based on error reduction of a maritime cooperative unmanned platform is shown in Fig. 1. The airborne unmanned platform observes the maritime cooperative unmanned platform and target simultaneously via its radar, obtaining position and velocity information. It then eliminates systematic error effects on target localization and tracking through an error reduction algorithm. Finally, the platform obtains relatively accurate target state information via filtering estimation. In this case, observation configuration affects the performance of the error reduction algorithm. Due to the unmanned platform's random initial position, ensuring an optimal observation configuration initially is challenging. The key of accurate target tracking via the maritime cooperative unmanned platform for airborne unmanned platform is determining the optimal observation geometry of the airborne unmanned platform, maritime cooperative unmanned platform, and marine target. The goal is to control the airborne unmanned platform to the optimal observation position.

Unmanned platform motion model

Airborne unmanned platforms are typically controlled using a mathematical model that incorporates an autopilot, a practical approach for engineering applications. Each platform is assumed to have a low-order flight controller that adjusts speed and heading angular velocity to regulate the platform's flight state. The fixed-height model of an airborne unmanned platform in three-dimensional space is formulated as:

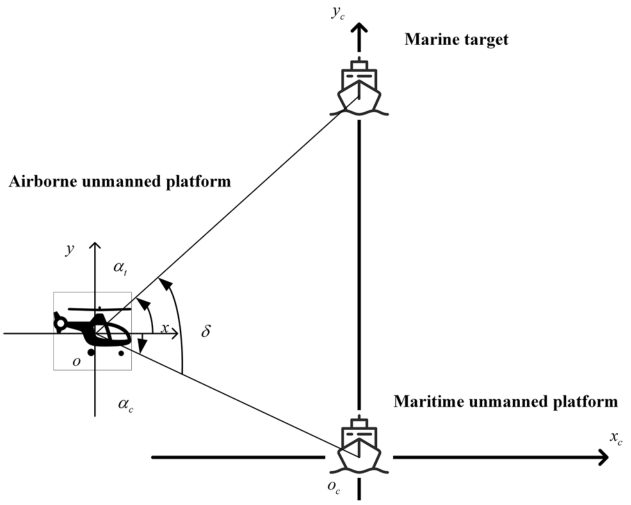


Fig. 1. Marine target tracking scenario by airborne and maritime unmanned platforms cooperation.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{h} \\ \dot{\psi} \\ \dot{v} \\ \dot{\omega} \end{pmatrix} = f(\boldsymbol{x},\boldsymbol{u}) = \begin{pmatrix} v\cos\psi \\ v\sin\psi \\ 0 \\ \omega \\ -\frac{1}{\tau_v}v + \frac{1}{\tau_v}u_v \\ -\frac{1}{\tau_\omega}\omega + \frac{1}{\tau_\omega}u_\omega \end{pmatrix} \tag{1}$$

where,$\boldsymbol{x} = [x, y, h, \psi, v, \omega]^T$ denotes the state of the airborne unmanned platform, including its position, heading angle, velocity, and heading angular velocity of the airborne unmanned platform. τ_v and τ_ω are velocity and angular velocity time delay constants related to the platform and its flight state. $\boldsymbol{u} = [u_v, u_\omega]^T$ constitutes the control input, indicating the autopilot speed control command and heading angular velocity control command. These commands adhere to the following constraints:

$$|u_v - v_0| \leq v_{\max} \tag{2}$$

$$|u_\omega| \leq \omega_{\max} \tag{3}$$

Here,v_0 denotes the cruise speed of the platform.$v_{\max}$ and $\omega_{\max}$ correspond to the maximum range of the cruise speed variation and the maximum heading angular velocity respectively.

In order to facilitate receding horizon optimization control of the airborne unmanned platform, the continuous-time model in Eq. (1) is converted into the following discrete-time model using Euler integration:

$$\boldsymbol{x}_{k+1} = f_d(\boldsymbol{x}_k,\boldsymbol{u}_k) = \boldsymbol{x}_k + T_s f(\boldsymbol{x}_k,\boldsymbol{u}_k) \tag{4}$$

where $\boldsymbol{x}_k = [x_k, y_k, \psi_k, v_k, \omega_k]^T$, $\boldsymbol{u}_k = [u_{vk}, u_{\omega k}]^T$ and T_s is the sampling time.

The motion of a maritime unmanned platform can be approximated as two-dimensional space motion, mainly involving speed and heading control. This motion model is similar to that of the airborne unmanned platform.

Airborne unmanned platform observation model

In the three-dimensional space, the state of the marine target at time k is denoted as $\mathbf{X}_k = [x_t, y_t, z_t]_k^T$, and the state of the airborne unmanned platform is $\mathbf{U}_k = [x_u, y_u, z_u]_k^T$. Relative vectors of the airborne unmanned platform and the target are $\mathbf{r}_k = [r_x, r_y, r_z]_k^T$, $r_x = x_u - x_t$, $r_y = y_u - y_t$, $r_z = z_u - z_t$. The target information observed by the airborne unmanned platform is:

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{X}_k, \mathbf{U}_k) + \mathbf{v}(k) = \begin{bmatrix} \sqrt{r_x^2 + r_y^2 + r_z^2} \\ \tan^{-1}\left(\frac{r_x}{r_y}\right) \\ \tan^{-1}\left(\frac{r_z}{\sqrt{r_x^2 + r_y^2}}\right) \end{bmatrix} + \mathbf{v}(k) \tag{5}$$

where $\mathbf{v}(k)$ denotes the observation noise at time k, obeying a zero-mean Gaussian distribution. The error covariance matrix is $R = \operatorname{diag}\left(\sigma_r^2, \sigma_\alpha^2, \sigma_\beta^2\right)$, where σ_r, σ_α, and σ_β denote the standard deviation of the sensor measurement of distance, bearing, and pitch respectively.

Hazardous area model

Airborne unmanned platforms operating in complex environments at sea may face threats from other airborne platforms and incoming targets. Therefore, it is necessary to consider the impact of hazardous areas. Based on the literature[13], hazardous areas are divided into collision avoidance and collision zones, and the hazardous area function is given by

$$O_i = \begin{cases} \left(\dfrac{R_a^2 - \|\boldsymbol{\chi} - \mathbf{P}\|_2^2}{\|\boldsymbol{\chi} - \mathbf{P}\|_2^2 - R_c^2}\right)^2, & R_c < \|\boldsymbol{\chi} - \mathbf{P}\| \leq R_a \\ 0, & \|\boldsymbol{\chi} - \mathbf{P}\| \geq R_a \end{cases} \tag{6}$$

where $\boldsymbol{\chi}$ denotes the position of the airborne unmanned platform and $\mathbf{P}$ denotes the position of the obstacle. The collision radius $\mathbf{\Phi}_c = \left\{\boldsymbol{\chi} \in R^2 | \, \|\boldsymbol{\chi} - \mathbf{P}\| < R_c\right\}$ and obstacle avoidance radius $\mathbf{\Phi}_a = \left\{\boldsymbol{\chi} \in R^2 | R_c < \|\boldsymbol{\chi} - \mathbf{P}\| \leq R_a\right\}$ are defined as shown in Fig. 2.

Error reduction model

This paper utilizes an error reduction algorithm based on a maritime cooperative unmanned platform for target tracking. Assuming the observation value of the airborne unmanned platform on the maritime cooperative unmanned platform is $(r + \Delta r, \alpha + \Delta\alpha, \beta + \Delta\beta)$, where r, α and β represent the true values of distance, azimuth and elevation angle observed by the airborne unmanned platform respectively, and Δr, $\Delta\alpha$ and $\Delta\beta$ are the corresponding observation errors. The position of the maritime cooperative unmanned platform in the airborne unmanned platform's unstable carrier coordinate system is obtained

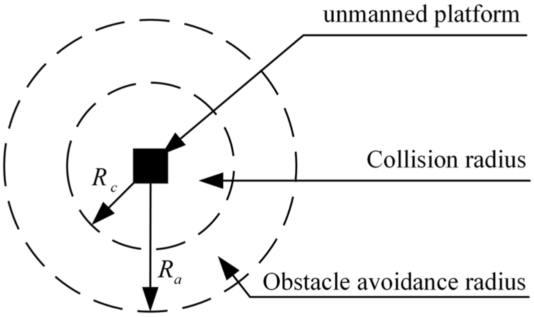


Fig. 2. Hazardous area for the unmanned platform.

$$\mathbf{X}_{co,u}=\begin{pmatrix}(r+\Delta r)\cos(\alpha+\Delta\alpha)\cos(\beta+\Delta\beta)\\(r+\Delta r)\sin(\alpha+\Delta\alpha)\cos(\beta+\Delta\beta)\\(r+\Delta r)\sin(\beta+\Delta\beta)\end{pmatrix} \tag{7}$$

In an ideal environment, there exists

$$\mathbf{X}_c=\mathbf{X}_f+\mathbf{T}_{uts}(\mathbf{P})\mathbf{X}_{co,u} \tag{8}$$

where: $\mathbf{X}_f$ and $\mathbf{X}_c$ are the positions of the maritime cooperative and airborne unmanned platform in the Earth coordinate system,$\mathbf{T}_{uts}$ is the transformation from unstable to stable coordinate system, $\mathbf{P}$ is the attitude of the airborne unmanned platform.

However, in the real environment, Eq. (8) is difficult to establish due to attitude error, observation error and airborne unmanned platform position error. Therefore, Eq. (9) is constructed for error reduction

$$\mathrm{f}(\mathbf{P})=|\mathbf{X}_f+\mathbf{T}_{uts}(\mathbf{P})\mathbf{X}_{co,u}-\mathbf{X}_c| \tag{9}$$

where:f $(\mathbf{P})$ denotes the error between the position of the airborne unmanned platform observing the maritime cooperative unmanned platform and the true position of the maritime cooperative unmanned platform.

We solve $\mathbf{P}_{\min}$ to make $\min \mathrm{f}(\mathbf{P})=\mathrm{f}(\mathbf{P}_{\min})$. Then $\mathbf{P}_{\min}$ is used for attitude value for marine target tracking to improve tracking accuracy.

Trajectory optimization

Tracking error analysis based on error reduction

Airborne unmanned platform observation model approximation

Airborne unmanned platforms observing maritime cooperative unmanned platform without accounting for random error can be described as $\mathbf{Z}_{c,m}$

$$\mathbf{Z}_{c,m}{=}\mathbf{Z}_{c,t}+\mathbf{E}_{c,se} \tag{10}$$

where: $\mathbf{Z}_{c,t}$ and $\mathbf{E}_{c,se}$ are the truth value and systematic error of observing the maritime cooperative unmanned platform.

Converted to Cartesian Coordinates

$$\mathbf{X}_{c,um}{=}\mathbf{h}_{str}(\mathbf{Z}_{c,t}+\mathbf{E}_{c,se}) \tag{11}$$

where: $\mathbf{h}_{str}$ denotes spherical coordinates in rectangular coordinates.

First-order Taylor expansion is at the truth value

$$\mathbf{X}_{c,um}\approx\mathbf{h}_{str}(\mathbf{Z}_{c,t})+\mathbf{H}_{str}(\mathbf{Z}_{c,t})\mathbf{E}_{c,se} \tag{12}$$

where: $\mathbf{H}_{str}$ is the Jacobian matrix of $\mathbf{h}_{str}$.

Without considering random error, the corrected attitude of the airborne unmanned platform is

$$\mathbf{P}_r{=}\mathbf{P}_t+\mathbf{P}_{rs} \tag{13}$$

where: $\mathbf{P}_t$ and $\mathbf{P}_{rs}$ are the truth and correction value of the attitude.

Unstable coordinate system transformed to stable coordinate system

$$\mathbf{T}_{uts}(\mathbf{P}_r){=}\mathbf{T}_{uts}(\mathbf{P}_t+\mathbf{P}_{rs}) \tag{14}$$

First-order Taylor expansion is at the truth value

$$\mathbf{T}_{uts}(\mathbf{P}_r)\approx\mathbf{T}_{uts}(\mathbf{P}_t)+\mathbf{dT}(\varphi)\mathbf{T}(\theta)\mathbf{T}(\gamma)\varphi_{rs}+\mathbf{T}(\varphi)\mathbf{dT}(\theta)\mathbf{T}(\gamma)\theta_{rs}+\mathbf{T}(\varphi)\mathbf{T}(\theta)\mathbf{dT}(\gamma)\gamma_{rs} \tag{15}$$

where: φ,θ,γ are yaw ,pitch and roll,$\varphi_{rs}\theta_{rs}\gamma_{rs}$ are the systematic error of yaw ,pitch and roll.

$$\mathbf{T}(\varphi)=\begin{bmatrix} \cos(\varphi) & \text{-}\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{T}(\theta)=\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \text{-}\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \mathbf{T}(\gamma)=\begin{bmatrix} \cos(\gamma) & 0 & \sin(\gamma) \\ 0 & 1 & 0 \\ \sin(\gamma) & 0 & \cos(\gamma) \end{bmatrix}$$

$$\mathbf{dT}(\varphi)=\begin{bmatrix} \text{-}\sin(\varphi) & \text{-}\cos(\varphi) & 0 \\ \cos(\varphi) & \text{-}\sin(\varphi) & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{dT}(\theta)=\begin{bmatrix} 0 & 0 & 0 \\ 0 & \text{-}\sin(\theta) & \text{-}\cos(\theta) \\ 0 & \cos(\theta) & \text{-}\sin(\theta) \end{bmatrix} \mathbf{dT}(\gamma)=\begin{bmatrix} \text{-}\sin(\gamma) & 0 & \cos(\gamma) \\ 0 & 0 & 0 \\ \cos(\gamma) & 0 & \text{-}\sin(\gamma) \end{bmatrix}$$

Let

$$\mathbf{dT}(\mathbf{P}_r) = \mathbf{dT}(\varphi)\mathbf{T}(\theta)\mathbf{T}(\gamma)\varphi_{rs} + \mathbf{T}(\varphi)\mathbf{dT}(\theta)\mathbf{T}(\gamma)\theta_{rs} + \mathbf{T}(\varphi)\mathbf{T}(\theta)\mathbf{dT}(\gamma)\gamma_{rs} \tag{16}$$

Therefore:

$$\mathbf{T}_{uts}(\mathbf{P}_r) \approx \mathbf{T}_{uts}(\mathbf{P}_t) + \mathbf{dT}(\mathbf{P}_r) \tag{17}$$

True positions of the maritime cooperative unmanned platform and target in the stable coordinate system with the airborne unmanned platform as the origin and considering the airborne unmanned platform's sway, are respectively

$$\mathbf{X}_{c,t} = \mathbf{T}_{uts}(\mathbf{P}_t)\mathbf{X}_{c,ut} \tag{18}$$

$$\mathbf{X}_{t,t} = \mathbf{T}_{uts}(\mathbf{P}_t)\mathbf{X}_{t,ut} \tag{19}$$

where $\mathbf{X}_{c,ut}$, $\mathbf{X}_{t,ut}$ represent true positions of the maritime cooperative unmanned platform and target in the unstable coordinate system with the airborne unmanned platform as the origin.

Error calculation based on error reduction
Errors present after obtaining the corrected attitude $\mathbf{P}_r$, theoretically

$$\mathbf{X}_{c,t} = \mathbf{T}_{uts}(\mathbf{P}_r)\mathbf{X}_{c,um} \tag{20}$$

$$\mathbf{X}_{t,t} = \mathbf{T}_{uts}(\mathbf{P}_r)\mathbf{X}_{t,um} \tag{21}$$

where $\mathbf{X}_{c,um}$, $\mathbf{X}_{t,um}$ represent observation value of the maritime cooperative unmanned platform and target in the unstable coordinate system with the airborne unmanned platform as the origin.

Integration of (18), (19), (20), (21)

$$\mathbf{T}_{uts}(\mathbf{P}_t)\mathbf{X}_{c,ut} - \mathbf{T}_{uts}(\mathbf{P}_t)\mathbf{X}_{t,ut} = \mathbf{T}_{uts}(\mathbf{P}_r)\mathbf{X}_{c,um} - \mathbf{T}_{uts}(\mathbf{P}_r)\mathbf{X}_{t,um} \tag{22}$$

Simplified, it can be obtained that:

$$\mathbf{T}_{uts}(\mathbf{P}_t)(\mathbf{X}_{c,ut} - \mathbf{X}_{t,ut}) = \mathbf{T}_{uts}(\mathbf{P}_r)(\mathbf{X}_{c,um} - \mathbf{X}_{t,um}) \tag{23}$$

First order Taylor expansion

$$\begin{aligned} &\mathbf{T}_{uts}(\mathbf{P}_t)(\mathbf{X}_{c,ut} - \mathbf{X}_{t,ut}) \\ &\approx (\mathbf{T}_{uts}(\mathbf{P}_t) + \mathbf{dT}_{uts}(\mathbf{P}_r))(\mathbf{X}_{c,ut} + \mathbf{H}_{str}(\mathbf{Z}_{c,t})\mathbf{E}_{c,se} - \mathbf{X}_{t,ut} - \mathbf{H}_{str}(\mathbf{Z}_{t,t})\mathbf{E}_{t,se}) \\ &= \mathbf{T}_{uts}(\mathbf{P}_t)(\mathbf{X}_{c,ut} - \mathbf{X}_{t,ut}) + \mathbf{T}_{uts}(\mathbf{P}_t)(\mathbf{H}_{str}(\mathbf{Z}_{c,t})\mathbf{E}_{c,se} - \mathbf{H}_{str}(\mathbf{Z}_{t,t})\mathbf{E}_{t,se}) + \\ &\mathbf{dT}_{uts}(\mathbf{P}_r)(\mathbf{X}_{c,ut} - \mathbf{X}_{t,ut}) + \mathbf{dT}_{uts}(\mathbf{P}_r)(\mathbf{H}_{str}(\mathbf{Z}_{c,t})\mathbf{E}_{c,se} - \mathbf{H}_{str}(\mathbf{Z}_{t,t})\mathbf{E}_{t,se}) \end{aligned} \tag{24}$$

Let

$$\mathbf{e}1 = \mathbf{T}_{uts}(\mathbf{P}_t)(\mathbf{H}_{str}(\mathbf{Z}_{c,t})\mathbf{E}_{c,se} - \mathbf{H}_{str}(\mathbf{Z}_{t,t})\mathbf{E}_{t,se}) \tag{25}$$

$$\mathbf{e}2 = \mathbf{dT}_{uts}(\mathbf{P}_r)(\mathbf{X}_{c,ut} - \mathbf{X}_{t,ut}) \tag{26}$$

$$\mathbf{e}3 = \mathbf{dT}_{uts}(\mathbf{P}_r)(\mathbf{H}_{str}(\mathbf{Z}_{c,t})\mathbf{E}_{c,se} - \mathbf{H}_{str}(\mathbf{Z}_{t,t})\mathbf{E}_{t,se}) \tag{27}$$

Among them,$\mathbf{e}1$ is the main error caused by the sensor, $\mathbf{e}2$ is the main error caused by the attitude and $\mathbf{e}3$ is the coupling error between the sensor and the attitude, which is small and negligible compared to the former.

The final error mainly consists of

$$\mathbf{e} = \mathbf{e}1 + \mathbf{e}2 \tag{28}$$

$$\mathbf{e}1 = \begin{bmatrix} (\cos\varphi_t\cos\gamma_t + \sin\varphi_t\sin\theta_t\sin\gamma_t)\sigma_1^{e1} + (\cos\varphi_t\sin\gamma_t + \sin\varphi_t\sin\theta_t\cos\gamma_t)\sigma_1^{e1} - \sin\varphi_t\cos\theta_t\sigma_3^{e1} \\ (\sin\varphi_t\cos\gamma_t - \cos\varphi_t\sin\theta_t\sin\gamma_t)\sigma_1^{e1} + (\sin\varphi_t\sin\gamma_t - \cos\varphi_t\sin\theta_t\cos\gamma_t)\sigma_2^{e1} + \cos\varphi_t\cos\theta_t\sigma_3^{e1} \\ \sin\theta_t\sigma_1^{e1} + \cos\theta_t\sin\gamma_t\sigma_2^{e1} + \cos\theta_t\cos\gamma_t\sigma_3^{e1} \end{bmatrix} \tag{29}$$

Among them:

$$\begin{aligned}\sigma_1^{e1} &= r_{c,se}\cos\alpha_{c,t}\cos\beta_{c,t} - r_{t,se}\cos\alpha_{t,t}\cos\beta_{t,t} - \alpha_{c,se}r_{c,t}\sin\alpha_{c,t}\cos\beta_{c,t}+\\ &\alpha_{t,se}r_{t,t}\sin\alpha_{t,t}\cos\beta_{t,t} - \beta_{c,se}r_{c,t}\cos\alpha_{c,t}\sin\beta_{c,t} + \beta_{t,se}r_{t,t}\cos\alpha_{t,t}\sin\beta_{t,t}\end{aligned} \tag{30}$$

$$\begin{aligned}\sigma_2^{e1} &= r_{c,se}\sin\alpha_{c,t}\cos\beta_{c,t} - r_{t,se}\sin\alpha_{t,t}\cos\beta_{t,t} + \alpha_{c,se}r_{c,t}\cos\alpha_{c,t}\cos\beta_{c,t}-\\ &\alpha_{t,se}r_{t,t}\cos\alpha_{t,t}\cos\beta_{t,t} - \beta_{c,se}r_{c,t}\sin\alpha_{c,t}\sin\beta_{c,t} + \beta_{t,se}r_{t,t}\sin\alpha_{t,t}\sin\beta_{t,t}\end{aligned} \tag{31}$$

$$\sigma_3^{e1} = r_{c,se}\sin\beta_{c,t} - r_{t,se}\sin\beta_{t,t} + \beta_{c,se}r_{c,t}\cos\beta_{c,t} - \beta_{t,se}r_{t,t}\cos\beta_{t,t} \tag{32}$$

$$\mathbf{e}2 = \begin{bmatrix} e_{21} \\ e_{22} \\ e_{23} \end{bmatrix} \tag{33}$$

Among them:

$$\begin{aligned}e_{21} &= \sigma_3^{e2}\left(\gamma_{rs}\left(\sigma_{11}^{e2} - \sigma_4^{e2}\right) - \varphi_{rs}\left(\sigma_8^{e2} - \sigma_7^{e2}\right) + \theta_{rs}\text{cos}\theta_t\text{cos}\gamma_t\text{sin}\varphi_t\right) -\\ &\left(\varphi_{rs}\text{cos}\varphi_t\text{cos}\theta_t - \theta_{rs}\text{sin}\varphi_t\text{sin}\theta_t\right)\sigma_1^{e2} - \sigma_2^{e2}\left(\varphi_{rs}\left(\sigma_9^{e2} - \sigma_6^{e2}\right) + \gamma_{rs}\left(\sigma_{10}^{e2} - \sigma_5^{e2}\right) - \theta_{rs}\text{cos}\theta_t\text{sin}\varphi_t\text{sin}\gamma_t\right)\end{aligned} \tag{34}$$

$$\begin{aligned}e_{22} &= \sigma_3^{e2}\left(\varphi_{rs}\left(\sigma_{10}^{e2} + \sigma_5^{e2}\right) + \gamma_{rs}\left(\sigma_9^{e2} + \sigma_6^{e2}\right) - \theta_{rs}\text{cos}\varphi_t\text{cos}\theta_t\text{cos}\gamma_t\right) -\\ &\left(\varphi_{rs}\text{cos}\theta_t\text{sin}\varphi_t + \theta_{rs}\text{cos}\varphi_t\text{sin}\theta_t\right)\sigma_1^{e2} - \sigma_2^{e2}\left(\gamma_{rs}\left(\sigma_8^{e2} + \sigma_8^{e2}\right) - \varphi_{rs}\left(\sigma_{11}^{e2} + \sigma_4^{e2}\right) + \theta_{rs}\text{cos}\varphi_t\text{cos}\theta_t\text{sin}\gamma_t\right)\end{aligned} \tag{35}$$

$$e_{23} = \sigma_2^{e2}\left(\gamma_{rs}\text{cos}\theta_t\text{cos}\gamma_t - \theta_{rs}\text{sin}\theta_t\text{sin}\gamma_t\right) - \left(\theta_{rs}\text{cos}\gamma_t\text{sin}\theta_t + \gamma_{rs}\text{cos}\theta_t\text{sin}\gamma_t\right)\sigma_3^{e2} + \theta_{rs}\text{cos}\theta_t\sigma_1^{e2} \tag{36}$$

$$\sigma_1^{e2}{=}r_{c,t}\text{cos}\beta_{c,t}\text{sin}\alpha_{c,t} - r_{t,t}\text{cos}\beta_{t,t}\text{sin}\alpha_{t,t} \tag{37}$$

$$\sigma_2^{e2} = r_{c,t}\text{cos}\beta_{c,t}\text{cos}\alpha_{c,t} - r_{t,t}\text{cos}\beta_{t,t}\text{cos}\alpha_{t,t} \tag{38}$$

$$\sigma_3^{e2} = r_{c,t}\text{sin}\beta_{c,t} - r_{t,t}\text{sin}\beta_{t,t} \tag{39}$$

$$\sigma_4^{e2} = \text{sin}\varphi_t\text{sin}\theta_t\text{sin}\gamma_t \tag{40}$$

$$\sigma_5^{e2} = \text{sin}\varphi_t\text{sin}\theta_t\text{cos}\gamma_t \tag{41}$$

$$\sigma_6^{e2} = \text{cos}\varphi_t\text{sin}\theta_t\text{sin}\gamma_t \tag{42}$$

$$\sigma_7^{e2} = \text{cos}\varphi_t\text{sin}\theta_t\text{cos}\gamma_t \tag{43}$$

$$\sigma_8^{e2} = \text{sin}\varphi_t\text{sin}\gamma_t \tag{44}$$

$$\sigma_9^{e2} = \sin\varphi_t\cos\gamma_t \tag{45}$$

$$\sigma_{10}^{e2} = \text{cos}\varphi_t\text{sin}\gamma_t \tag{46}$$

$$\sigma_{11}^{e2} = \text{cos}\varphi_t\text{cos}\gamma_t \tag{47}$$

where φ_t,θ_t,γ_t are true value of yaw ,pitch and roll; $r_{c,t}$,$\alpha_{c,t}$,$\beta_{c,t}$ represent the true values of distance, azimuth and elevation angle of the airborne unmanned platform to the cooperative platform respectively; $r_{c,se}$,$\alpha_{c,se}$,$\beta_{c,se}$ represent corresponding systematic errors of the airborne unmanned platform to the cooperative platform; $r_{t,t}$,$\alpha_{t,t}$,$\beta_{t,t}$ represent the true values of distance, azimuth and elevation angle of the airborne unmanned platform to the maritime target respectively; $r_{t,se}$,$\alpha_{t,se}$,$\beta_{t,se}$ represent corresponding systematic errors of the airborne unmanned platform to the maritime target.

Analysis of optimal observation configuration

Azimuth error is the main source of sensor error for long-range marine target detection. Therefore, this subsection focuses on azimuth systematic error to discuss optimal observation configurations for unmanned platforms.

Assuming the airborne unmanned platform flies at a fixed height, it can be regarded as moving in a two-dimensional plane. The azimuths of the marine target observed by both the airborne and maritime cooperative unmanned platforms serve as reference quantities for the relative positions of these platforms, and the target.

For convenience in discussion, we set the distance, elevation and attitude systematic errors in Eq. (28) to 0. Equation (28) can be simplified to

$$\mathbf{e}_\alpha = \begin{bmatrix} r_{c,t}\alpha_{c,se}\cos\beta_{c,t}\text{sin}\alpha_{c,t} - r_{t,t}\alpha_{t,se}\cos\beta_{t,t}\text{sin}\alpha_{t,t} \\ r_{t,t}\alpha_{t,se}\cos\beta_{t,t}\text{cos}\alpha_{t,t} - r_{c,t}\alpha_{c,se}\cos\beta_{c,t}\text{cos}\alpha_{c,t} \\ 0 \end{bmatrix} \tag{48}$$

The systematic error in the azimuths of the airborne unmanned platform observing the target and the maritime cooperative unmanned platform during a cycle is so small that it can be approximated as the same value for both. This value can be denoted as α_{se}.

Solving the optimal observation configuration problem can be translated into minimizing the marine target position error. An objective function G_α is constructed as follows

$$G_\alpha = \sqrt{\left(r_{c,t}\alpha_{c,se}\cos\beta_{c,t}\text{sin}\alpha_{c,t} - r_{t,t}\alpha_{t,se}\cos\beta_{t,t}\text{sin}\alpha_{t,t}\right)^2 + \left(r_{t,t}\alpha_{t,se}\cos\beta_{t,t}\text{cos}\alpha_{t,t} - r_{c,t}\alpha_{c,se}\cos\beta_{c,t}\text{cos}\alpha_{c,t}\right)^2} \quad (49)$$

It can be simplified

$$G_\alpha = \sqrt{\alpha_{se}^2\left(r_{c,t}^2\cos^2\beta_{c,t} + r_{t,t}^2\cos^2\beta_{t,t}\right) - 2r_{t,t}r_{c,t}\alpha_{se}^2\cos\beta_{c,t}\cos\beta_{t,t}\cos\left(\alpha_{c,t} - \alpha_{t,t}\right)} \quad (50)$$

Let $r_{c,xyt} = r_{c,t}\cos\beta_{c,t}$, $r_{t,xyt} = r_{t,t}\cos\beta_{t,t}$, $\delta = \alpha_{c,t} - \alpha_{t,t}$, assume $r_{c,xyt} = |c_r|\, r_{t,xyt}$, where c_r is the scale factor, and $|c_r| < 1$(for the safety of the detection platforms, it is assumed that the airborne unmanned platform is always closer to the maritime cooperative unmanned platform. Equation (50) is simplified

$$\begin{aligned} G_\alpha &= \sqrt{\alpha_{se}^2\left(c_r^2 r_{t,xyt}^2 + r_{t,xyt}^2\right) - 2\alpha_{se}^2\,|c_r|\, r_{t,xyt}^2\cos\delta} \\ &= \sqrt{\alpha_{se}^2 r_{t,xyt}^2\left(c_r^2 - 2\cos\delta\,|c_r| + 1\right)} \end{aligned} \quad (51)$$

From Eq. (51), we can conclude that there are two factors affecting the observation position c_r and δ. c_r denotes the distance position relationship between the airborne unmanned platform, maritime cooperative unmanned platform and target.δ denotes the angle position relationship between the airborne unmanned platform and maritime cooperative unmanned platform.

A two-stage analysis is used for the above two factors, with c_r considered in the first stage and δ in the second stage.

The phase1 begins with the analysis of c_r.

Equation (51) is rewritten as

$$G_\alpha = \begin{cases} \sqrt{\alpha_{se}^2 r_{t,xyt}^2\left(c_r^2 + 2\cos\delta c_r + 1\right)} & c_r < 0 \\ \sqrt{\alpha_{se}^2 r_{t,xyt}^2\left(c_r^2 - 2\cos\delta c_r + 1\right)} & c_r \geq 0 \end{cases} \quad (52)$$

It is easy to get the minimum value of G_α when $c_r = -\cos\delta$ or $\cos\delta G_{\alpha\,\text{min}}^{c_r}$

$$G_{\alpha\,\text{min}}^{c_r} = \sqrt{\alpha_{se}^2 r_{t,xyt}^2\left(1 - \cos^2\delta\right)} \quad (53)$$

In phase2, we analyze δ.

It is easy to obtain that when $\cos\delta = 1\text{or} - 1$, $G_{\alpha\,\text{min}}^{c_r}$ obtains the minimum value $G_{\alpha\,\text{min}}^{\delta}$.

When $\cos\delta = 1\text{or} - 1$ is $\delta = 0\text{or}\pi$, the angular positional relationship is such that the airborne unmanned platform, maritime cooperative unmanned platform, and target are in a straight line.

When $\cos\delta = 1\text{or} - 1$ is $r_{c,xyt} = r_{t,xyt}$, the distance positional relationship is that the horizontal projection from the airborne unmanned platform to the maritime cooperative unmanned platform is equal to the horizontal projection from the airborne unmanned platform to the marine target.

Among the above positional relationships, the angular positional relationship is easy to satisfy and should be satisfied as the primary position, and the distance positional relationship should be considered according to the actual observation environment. Therefore, the angular positional relationship is a primary consideration for trajectory optimization.

Objective function based on receding horizon optimization

Receding horizon optimization is an optimization method for dealing with dynamic systems. Based on the receding horizon optimization and the error analysis in 3.1, the performance index of unmanned platforms cooperatively in observing marine targets at moment k is obtained as:

$$I_{o,k} = \sum_{m=k}^{k+N_p}\|\mathbf{e}_m\| \quad (54)$$

where N_p is the number of predicted steps and $\mathbf{e}_m$ is the tracking error of the unmanned platform cooperative observation at time m.

The hazardous area indicator J_h measures the ability of the airborne unmanned platform to avoid the hazardous area, and the value of J_h is inversely related to the distance from the hazardous area, i.e. The larger J_h is, the closer the airborne unmanned platform is to the obstacle, and the smaller J_h is, the further the airborne unmanned platform is from the obstacle. In order to make the airborne unmanned platform avoid the hazardous area, combined with the hazardous area model, we can get the hazardous area indicator at time k is:

$$J_{h,k} = \sum_{n=k}^{k+N_p-1} O_n \quad (55)$$

Among them:

$$O_n = \begin{cases} \sum_{m=1}^{M} \left(\frac{R_a^2 - \|\chi_n - \mathbf{P}_m\|_2^2}{\|\chi_n - \mathbf{P}_m\|_2^2 - R_c^2} \right)^2, R_c < \|\chi_n - \mathbf{P}_m\| \le R_a \\ 0 \qquad , \qquad \|\chi_n - \mathbf{P}_m\| \ge R_a \end{cases} \tag{56}$$

where:$\mathbf{P}_m$ denotes the location of the $m\mathrm{th}$ obstacle observed, and M denotes the number of obstacles observed.

Therefore, when considering the hazardous area, the optimization problem of cooperative marine target observation by multiple unmanned platforms can be transformed into a nonlinear model optimal control problem of minimizing the objective function under constraints, and the mathematical model is as follows:

$$\begin{cases} \min\limits_{\boldsymbol{u}_k^*} I_{o,k} + I_{h,k} \\ s.t. \ \boldsymbol{x}_{i,k+1} - f_d(\boldsymbol{x}_{i,k}, \boldsymbol{u}_{i,k}) = 0 \\ L_v(\boldsymbol{u}_{i,k}) = \frac{|u_{i,vk} - v_0| - v_{\max}}{v_{\max}} \le 0 \\ L_\omega(\boldsymbol{u}_{i,k}) = \frac{|u_{i,\omega k}| - \omega_{\max}}{\omega_{\max}} \le 0 \end{cases} \tag{57}$$

where:$\boldsymbol{u}_k$ is the control input variable of the airborne unmanned platform at time $k.v_0$ is the cruising speed of the airborne unmanned platform. $v_{\max}$ is the maximum speed value.$\omega_{\max}$ is the maximum value of the angular velocity of the heading.

Optimal control of the airborne unmanned platform based on AEDBO

Adaptive enhanced dung beetle optimization

Improved population initialization

DBO employs low-randomness, high-uncertainty initialization, which often results in uneven initial populations with poor diversity and narrow search ranges. To ensure the convergence accuracy and speed of the algorithm, it is necessary to make the population distribution more uniform and increase population diversity during initialization. We propose an improved adaptive Tent chaotic mapping to achieve more uniform distributions and enhanced diversity.

The original Tent chaotic mapping expression is:

$$xtm_{i+1} = f_{Tent}(xtm_i) = \begin{cases} xtm_i/\kappa & 0 < xtm_i < \kappa \\ (1 - xtm_i)/(1-\kappa) & \kappa \le xtm_i \le 1 \end{cases} \tag{58}$$

An improved version is:

$$xtm_{i+1} = f_{Tent}(xtm_i) = \begin{cases} 1 - \zeta * rand(0,1) + rand(0,1)/N & xtm_i = 0 \\ 1 - xtm_i/\kappa + rand(0,1)/N & 0 < xtm_i < \kappa \\ (1 - xtm_i)/(1-\kappa) + rand(0,1)/N & \kappa \le xtm_i \le 1 \end{cases} \tag{59}$$

where:$\zeta \in (0,1]$ is the adjustment factor.$\kappa \in (0,2]$ is the chaos factor. This paper sets ζ to 1and κ to 0.5.

Mapped populations are initialized as

$$x_{i,j} = lb + (ub - lb) \times xtm_i \tag{60}$$

where:lb and ub denote lower and upper bound of search space respectively.

Figure 3 shows frequency distributions of different Tent chaotic sequences. Tent chaotic mapping 1 is from Ref.[44] The Tent chaotic mapping 2 is from Ref.[45] . As seen in Fig. 3, the improved chaotic mapping in this paper generates more uniform initial population distributions versus other common improved Tent chaotic mappings, enhancing initial search ability and the likelihood of obtaining the global optimum.

Moreover, algorithm development can entail significant meaningless search from chaotic inverse learning, increasing computational cost and impeding convergence. To address this, we introduce center-of-mass adversarial inverse learning to enhance initial population quality using fitness values.

The center of mass $\mathbf{C}$ is defined as:

$$\mathbf{C} = \frac{\mathbf{X}_1 + \mathbf{X}_2 + \cdots + \mathbf{X}_n}{n} \tag{61}$$

where:

$$M_j = \frac{\sum\limits_{j=1}^{D} x_{i,j}}{n}, i = 1, 2, \cdots, n \tag{62}$$

The center of gravity reversal point $\widehat{\mathbf{X}}_i$ is defined as

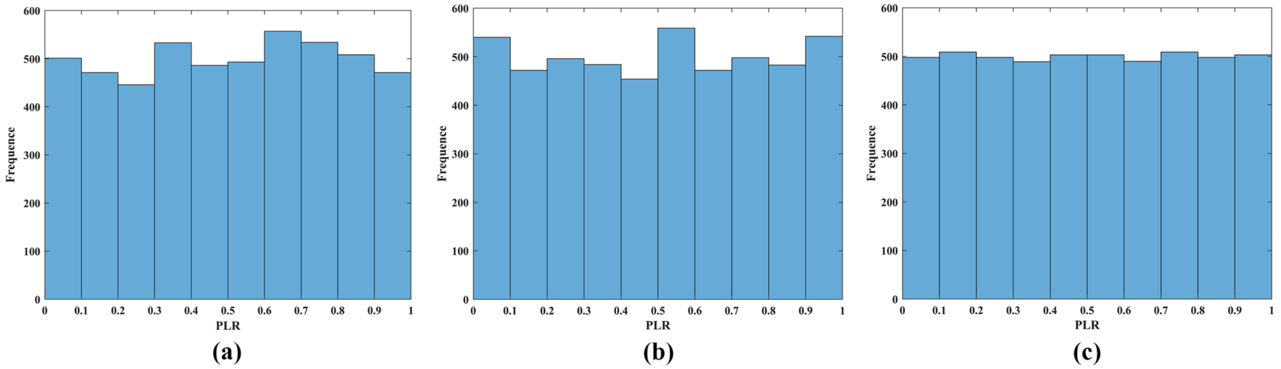


Fig. 3. Frequency distribution of different Tent chaotic sequences. (**a**) Tent chaotic mapping1, (**b**) Tent chaotic mapping2, (**c**) Improved Tent chaotic mapping.

$$\widehat{\mathbf{X}}_i = 2 \times \mathbf{C} - \mathbf{X}_i, i = 1, 2, \cdots, n \tag{63}$$

The reversal point lies in a dynamic boundary search space, denoted $x_{i,j} \in [a_j, b_j]$. A change in the dynamic boundary places the reversal point in a shrinking and constantly-changing space. The dynamic boundary expression is:

$$a_j = \min(x_{i,j}) \tag{64}$$

$$b_j = \max(x_{i,j}) \tag{65}$$

If the reversal point falls outside the dynamic boundary range, it is recalculated with the expression:

$$\overline{x}_{i,j} = \begin{cases} a_j + rand(0,1) \times (M_j - a_j) & \overline{x}_{i,j} < a_j \\ M_j + rand(0,1) \times (b_j - M_j) & \overline{x}_{i,j} > b_j \end{cases} \tag{66}$$

Adaptive convergence factor

The original convergence factor decreases linearly, and the area of the optimal spawning region decreases gradually, so the algorithm focuses on global search in the early stage and local search in the later stage. To enhance the search ability of the algorithm, a new adaptive convergence factor is proposed. When $0 \le t \le T_{\max}/2$, R decreases first slowly and then quickly, and the optimal spawning area decreases slowly, allowing the dung beetle to explore more unknown areas and avoid falling into the local optimum, improving the algorithm's search ability in the early stage; when $T_{\max}/2 \le t \le T_{\max}$, R decreases first quickly and then slowly, and the optimal spawning decreases fast, forcing the dung beetle to conduct a fine search in the vicinity of the current optimal solution to increase solution precision, improving the algorithm's local search ability in the later stage. The algorithm can improve the local search ability in the later stage. In summary, new adaptive convergence factor improves the global search ability of the algorithm in the early stage and the local search ability in the late stage, thus enhancing the algorithm's ability to obtain the optimal solution and convergence speed. Comparison of convergence factor is shown in Fig. 4. The expression of the new adaptive convergence factor is:

$$R = \begin{cases} 1 - \frac{e^{2t/T_{\max}} - 1}{2(e-1)} & 0 \le t \le \frac{T_{\max}}{2} \\ \frac{e^{2(1-t/T_{\max})} - 1}{2(e-1)} & \frac{T_{\max}}{2} \le t \le T_{\max} \end{cases} \tag{67}$$

Adaptive nonlinear rolling dung beetle quantity decreasing model

The number of ball-rolling dung beetles in DBO was kept constant. Ball-rolling dung beetles provide global search capability to the algorithm. To further enhance the algorithm's search capability in the early stage, an adaptive nonlinear ball-rolling dung beetle number decreasing model is designed. This model increases the proportion of ball-rolling dung beetles early, then gradually decreases it over iterations. The expression for rolling dung beetle quantity is:

$$P = \begin{cases} P_1 + P_2 * \left[\cos\left(\frac{t}{T_{\max}}\pi\right)\right]^q & 0 \le t \le \frac{T_{\max}}{2} \\ P_1 - P_2 * \left[-\cos\left(\frac{t}{T_{\max}}\pi\right)\right]^q & \frac{T_{\max}}{2} \le t \le T_{\max} \end{cases} \tag{68}$$

$$P_1 = \frac{P_{initial} + P_{end}}{2} \tag{69}$$

$$P_2 = \frac{P_{initial} - P_{end}}{2} \tag{70}$$

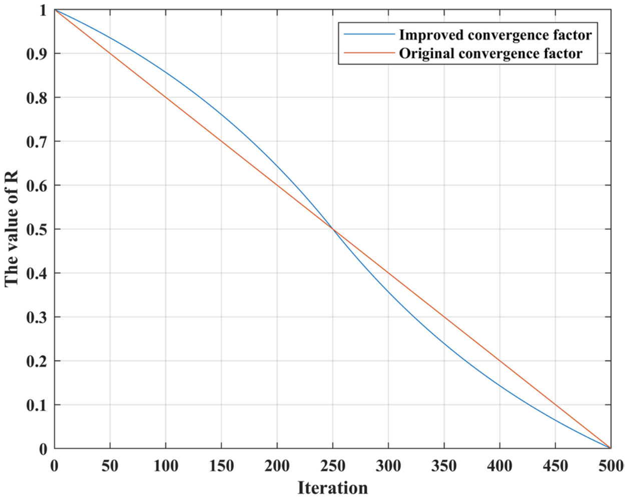


Fig. 4. Comparison of convergence factor.

where:$P_{initial}$ is the initial rolling dung beetle population proportion. P_{end} is the final rolling dung beetle population proportion. $q \in (0,1]$ is the decay index. In this paper,$P_{initial}$ is taken as 0.4,P_{end} is taken as 0.2and q is taken as 1.

The proportion of the number of dung beetles stolen is

$$P_t = 0.6 - P \tag{71}$$

Theft dung beetles contribute to the algorithm's local search capability. As iterations proceed, the number of theft dung beetles increases, enhancing algorithm's local search capability.

When $0 \le t \le T_{\max}/2$, the number of rolling dung beetles is kept higher in the early stages to enhance global search; when $T_{\max}/2 \le t \le T_{\max}$, the number of rolling dung beetles is reduced in the later stages and the number of stealing dung beetles is increased, freeing up resources to be used to improve localized search.

Adaptive cauchy-gaussian mutation strategy

In the later stage of DBO, dung beetles gradually move closer to optimal individuals, potentially resulting in lack of population diversity and premature convergence. To address this, a Cauchy-Gaussian mutation strategy is introduced. The Cauchy distribution has heavy-tailed characteristics and is suitable for long-distance jumps, dominating global exploration in the early stage; the Gaussian distribution is suitable for local fine-tuning, promoting local development in the late stage. The algorithm selects the current best individual for mutation, compares positions before and after mutation, and selects the better position for the next iteration. The Cauchy-Gaussian mutation strategy is mathematically defined as:

$$U_{best}^{t+1} = X_{best}^{t}\left[1 + \lambda_1 Cauchy\left(0,\sigma^2\right) + \lambda_2 Gauss\left(0,\sigma^2\right)\right] \tag{72}$$

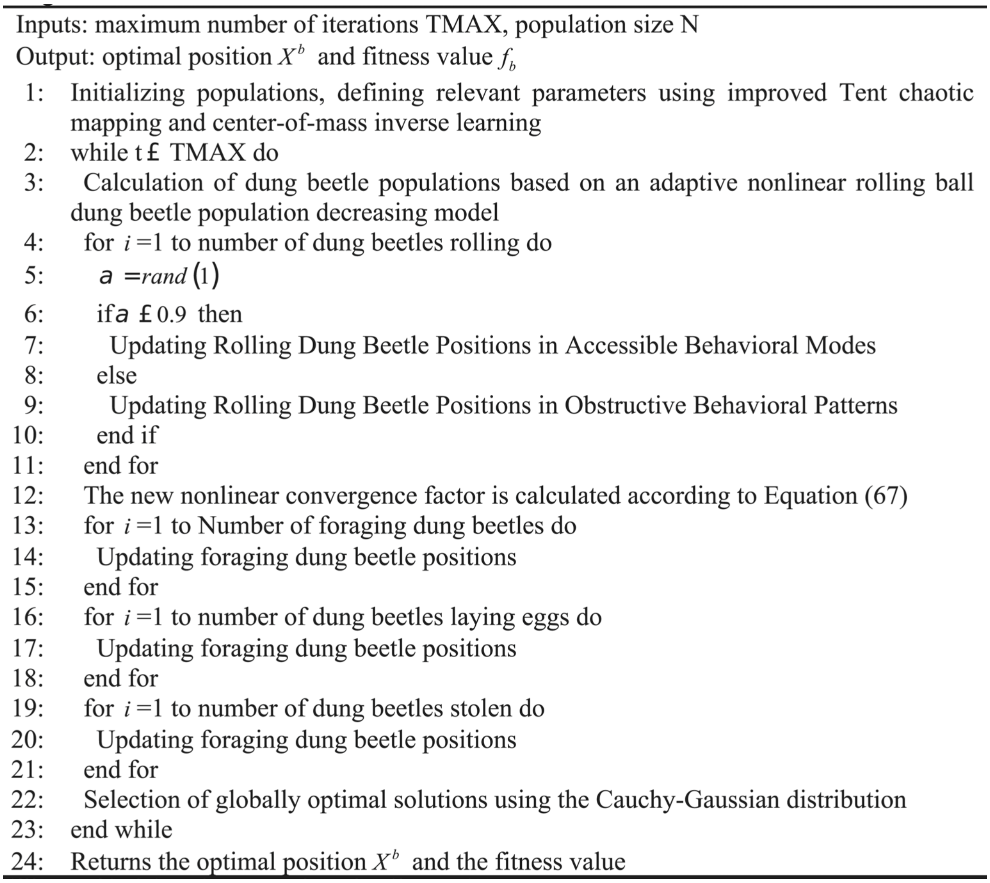
$$\sigma = \begin{cases} 1 & f\left(X_{best}\right) < f\left(X_i\right) \\ \exp\left(\frac{f(X_{best}) - f(X_i)}{|f(X_{best})|}\right) & otherwise \end{cases} \tag{73}$$

$$\lambda_1 = 1 - \frac{t^2}{T_{\max}^2} \tag{74}$$

$$\lambda_2 = \frac{t^2}{T_{\max}^2} \tag{75}$$

where:X_{best} is the position of the elite individual.U_{best}^{t+1} is the position of the elite individual after mutation.σ is the standard deviation of the Cauchy-Gaussian mutation strategy. $Cauchy\left(0,\sigma^2\right)$ is a random variable satisfying the Cauchy distribution.$Gauss\left(0,\sigma^2\right)$ is a random variable satisfying the Gaussian distribution. λ_1 and λ_2 are dynamic parameters adaptively adjusted with iterations. Initially,λ_1 is larger so the algorithm can explore the optimal solution over a larger range with larger mutation steps, and λ_2 is smaller so the algorithm can search near the optimal solution. During the search,λ_1 gradually decreases and λ_2 increases.

AEDBO framework is shown Algorithm.



```
Inputs: maximum number of iterations TMAX, population size N
Output: optimal position $X^b$ and fitness value $f_b$
Initializing populations, defining relevant parameters using improved Tent chaotic mapping and center-of-mass inverse learning
while t £ TMAX do
  Calculation of dung beetle populations based on an adaptive nonlinear rolling ball dung beetle population decreasing model
  for $i=1$ to number of dung beetles rolling do
    $a = rand(1)$
    if $a$ £ 0.9 then
      Updating Rolling Dung Beetle Positions in Accessible Behavioral Modes
    else
      Updating Rolling Dung Beetle Positions in Obstructive Behavioral Patterns
    end if
  end for
  The new nonlinear convergence factor is calculated according to Equation (67)
  for $i=1$ to Number of foraging dung beetles do
    Updating foraging dung beetle positions
  end for
  for $i=1$ to number of dung beetles laying eggs do
    Updating foraging dung beetle positions
  end for
  for $i=1$ to number of dung beetles stolen do
    Updating foraging dung beetle positions
  end for
  Selection of globally optimal solutions using the Cauchy-Gaussian distribution
end while
Returns the optimal position $X^b$ and the fitness value
```

Algorithm. AEDBO framework.

Computational complexity analysis

It is assumed that N,D,$Iter$ and represent the population size, optimization problem dimension and number of iterations respectively. The time complexity of DBO is O $(N * D * Iter)$ and the space complexity is O $(N * D)$.

Analyze the time complexity of AEDBO. AEDBO adopts the improved Tent chaotic mapping in the population initialization stage to generate a chaotic sequence with a time complexity of O $(N * D)$, the time complexity of the center-of-mass adversarial inverse learning strategy to compute the center-of-mass and generate the adversarial solution is O $(N * D)$, and the total time complexity of initialization is O $(N * D)$. In the single iteration stage, the adaptive convergence factor changes the calculation method of the convergence factor without introducing additional loop commands, which does not affect the computational complexity; the adaptive nonlinear rolling ball dung beetle number decreasing model only changes the proportion of different species of dung beetles, not the total number of dung beetles, and the computational time complexity of a single position update for the whole population is O $(N * D)$; the introduction of the adaptive Cauchy-Gaussian mutation strategy increases the computation of the elite individuals, and the time complexity is O (D), and the single iteration time complexity is O $(N * D)$. In summary, the total time complexity is $O\ (N * D * Iter)$.

Analyze the space complexity of AEDBO. When storing the population, the original population stores $N * D$ parameters; different dung beetles in the population generate temporary solutions during iteration, requiring O $(N * D)$ space for temporary storage, and the total population space is O $(N * D)$. When calculating intermediate variables, O (1) space is needed to store the D dimensional center-of-mass vector, the fitness values of N individual dung beetles, and other parameters such as the convergence and scaling, and the total intermediate variable space is . The adaptive Cauchy-Gaussian mutation strategy increases elite individual's positions and fitness, requiring O $(N + D)$ space. In summary, the total space complexity is O $(N * D)$.

The above analysis reveals that AEDBO and DBO have the same computational time complexity and space complexity, indicating that their superior performance does not depend on higher complexity, proving their feasibility in practical applications.

Optimal control of the airborne unmanned platform Based on AEDBO

Based on the above analysis and improvements, AEDBO is utilized to solve the optimal control problem of an airborne unmanned platform. Specific solution steps are as follows:

Step 1: Determine the initial parameters, including the unmanned platform system parameters (initial position of the airborne unmanned platform, initial position of the maritime unmanned platform), AEDBO parameters (population number, maximum number of iterations, etc.), and error reduction algorithm parameters;

Step 2: Real-time generation of observation information of the airborne unmanned platform on the maritime cooperative unmanned platform and target, and use AEDBO to obtain the optimal control sequence at the current moment;

Step 3: Control the movement of the airborne unmanned platform using the obtained control quantity to obtain the position of the airborne unmanned platform at the next moment;

Step 4: Repeat step 2 and 3 to obtain the optimal motion trajectory of the airborne unmanned platform.

Figure 5 shows the flowchart of optimal control of an airborne unmanned platform based on AEDBO.

Simulation results and discussions

Performance analysis of AEDBO

The performance of AEDBO is first analyzed, including comparisons with other swarm intelligence algorithms and ablation experiments.

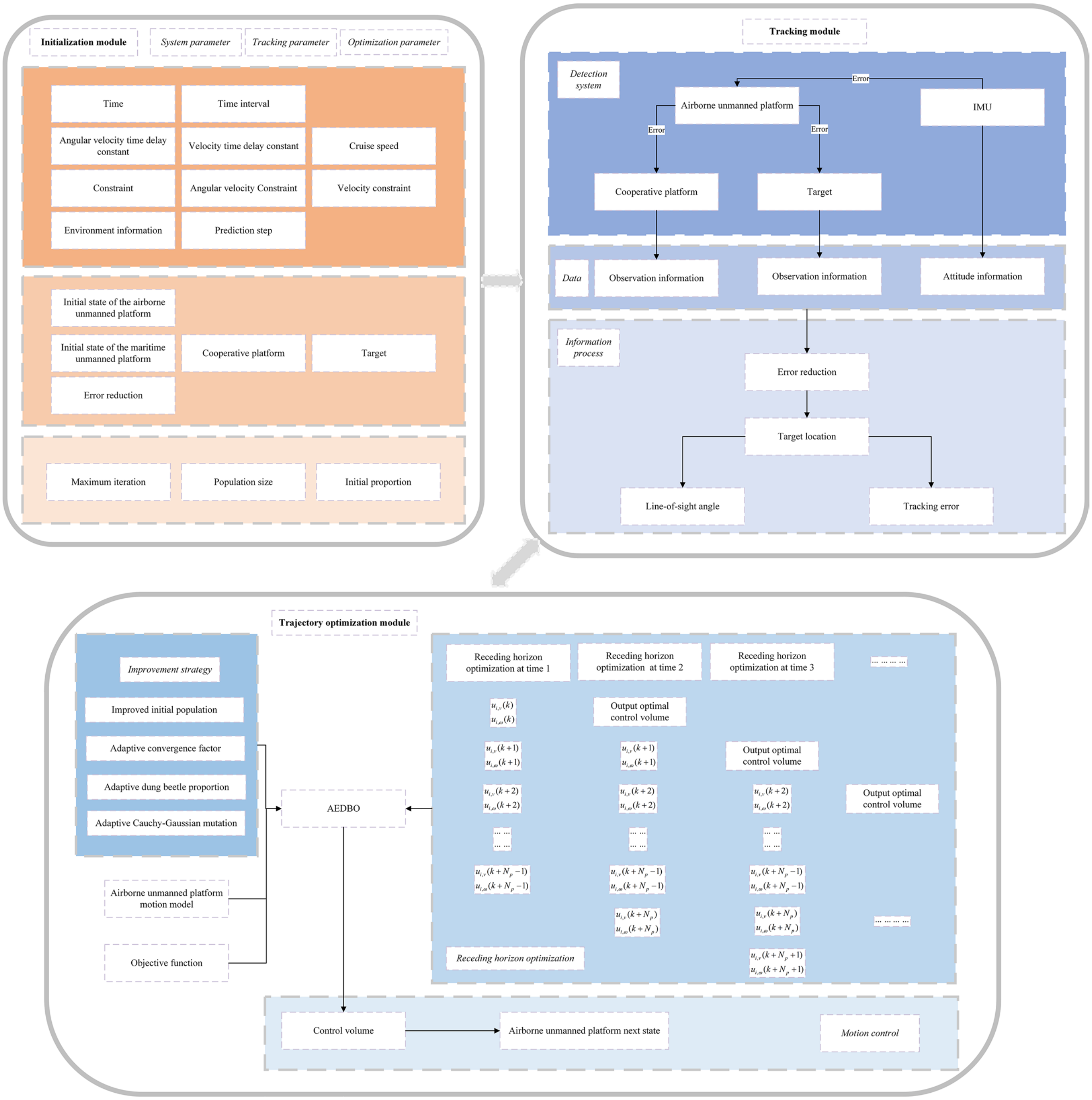

Initialization module
System parameter
Tracking parameter
Optimization parameter
Time
Time interval
Angular velocity time delay constant
Velocity time delay constant
Cruise speed
Constraint
Angular velocity Constraint
Velocity constraint
Environment information
Prediction step
Initial state of the airborne unmanned platform
Initial state of the maritime unmanned platform
Cooperative platform
Target
Error reduction
Maximum iteration
Population size
Initial proportion
Tracking module
Detection system
Error
Airborne unmanned platform
IMU
Error
Error
Cooperative platform
Target
Data
Observation information
Observation information
Attitude information
Information process
Error reduction
Target location
Line-of-sight angle
Tracking error
Trajectory optimization module
Improvement strategy
Improved initial population
Adaptive convergence factor
Adaptive dung beetle proportion
Adaptive Cauchy-Gaussian mutation
Airborne unmanned platform motion model
Objective function
AEDBO
Receding horizon optimization at time 1
Receding horizon optimization at time 2
Receding horizon optimization at time 3
... ...
$u_{i,v}(k)$ $u_{i,\omega}(k)$
Output optimal control volume
$u_{i,v}(k+1)$ $u_{i,\omega}(k+1)$
$u_{i,v}(k+2)$ $u_{i,\omega}(k+2)$
... ...
$u_{i,v}(k+N_p-1)$ $u_{i,\omega}(k+N_p-1)$
$u_{i,v}(k+N_p)$ $u_{i,\omega}(k+N_p)$
$u_{i,v}(k+N_p+1)$ $u_{i,\omega}(k+N_p+1)$
Receding horizon optimization
Control volume
Airborne unmanned platform next state
Motion control

Fig. 5. Optimal control of the airborne unmanned platform based on AEDBO.

Function	dimension	Scope	optimum
$f_1(x) = \sum_{i=1}^{D} x_i^2$	30	$[-100, 100]$	0
$f_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i-1}^{D} \lvert x_i \rvert$	30	$[-10, 10]$	0
$f_3(x) = \sum_{i=1}^{D} (\sum_{j-1}^{D} x_i)^2$	30	$[-100, 100]$	0
$f_4(x) = max_i \{\lvert x_i \rvert, 1 \leq i \leq D\}$	30	$[-100, 100]$	0
$f_5(x) = \sum_{i=1}^{D} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	$[-500, 500]$	-418.9829*D
$f_6(x) = 20 + e - 20 \exp(-0.2\sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2)} - \exp(\frac{1}{D} \sum_{i=1}^{D} \cos(2\pi x_i))$	30	$[-32, 32]$	8.8818e-16
$f_7(x) = \frac{1}{4000} \sum_{i=1}^{D} (x_i^2) - \left(\prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}})\right) + 1$	30	$[-600, 600]$	0
$f_8(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2} (x_i - a_{ij})^6}\right)^{-1}$	2	$[-65.53, 65.53]$	0.998004
$f_9(x) = \sum_{i=1}^{11} \left(a_i - \frac{x_1 (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right)^{-1}$	4	$[-5, 5]$	0.0003075
$f_{10}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times (30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-5, 5]$	3
$f_{11}(x) = -\sum_{i=1}^{4} \left(c_i \exp(-\sum_{j-1}^{3} a_{ij} (x_j - p_{ij})^2\right)$	3	$[0, 1]$	-3.8628
$f_{12}(x) = -\sum_{i=1}^{4} \left(c_i \exp(-\sum_{j-1}^{6} a_{ij} (x_j - p_{ij})^2\right)$	6	$[0, 1]$	-3.32
$f_{13}(x) = -\sum_{i=1}^{5} \left[(X - a_i) (X - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	-10.1532
$f_{14}(x) = -\sum_{i=1}^{7} \left[(X - a_i) (X - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	-10.4029
$f_{15}(x) = -\sum_{i=1}^{10} \left[(X - a_i) (X - a_i)^T + c_i\right]^{-1}$	4	$[0, 10]$	-10.5364

Table 1. Benchmark function.

Comparison of AEDBO with Other swarm intelligence optimization algorithms

To evaluate AEDBO performance, we test it on 15 benchmark function sets and compare it with 7 other swarm intelligence optimization algorithms. The 15 benchmark function sets are shown in Table 1. $f_1(x)$ ~ $f_4(x)$ are single-peak test functions for assessing algorithmic optimization performance; $f_5(x)$ ~ $f_7(x)$ are multi-peak test functions for evaluating local optimum avoidance ability; $f_8(x)$ ~ $f_{15}(x)$ are fixed-dimension multi-peak test functions for assessing local optimum avoidance and high-dimension problem handling performance. The 7 swarm intelligence optimization algorithms include both classical and contemporary algorithms, specifically PSO, FOA, GWO, WOA, HHO, SSA and DBO to provide a comprehensive performance comparison. Evaluation metrics include optimal value, mean value and standard deviation of calculation results. To ensure the experimental validity, the initial population size of all algorithms is set to 30, and maximum number of iterations is 500.To eliminate experimental chance, each algorithm is run independently on test functions for 100 times.

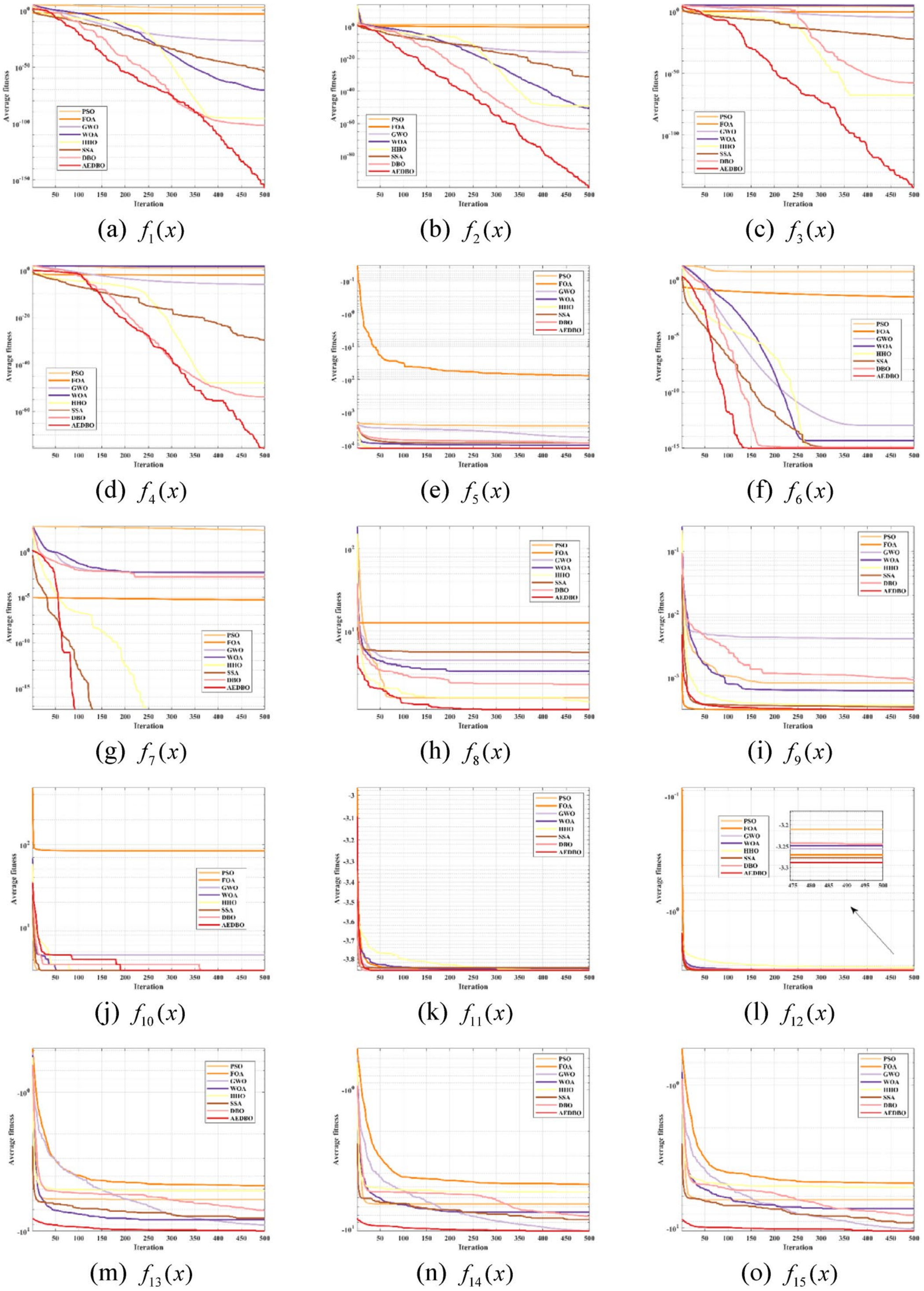
Figure 6 shows the convergence curves of AEDBO with other 7 swarm intelligence algorithms on 15 test functions. Table 2 shows the metrics statistics of AEDBO and other 7 swarm intelligence algorithms on 15 test functions. Figure 7 shows the performance ranking of different algorithms in comparison experiments on the test functions.

The following analysis can be made from the experimental results:

(1) The experimental results on the single-peak test functions ~ show that both AEDBO have better global optimality seeking ability and robustness. In the early stage, the ability of optimization seeking of AEDBO is slightly weaker than that of HHO and SSA, but in the middle and late stage, both can obtain optimal performance result. The experimental results verify the stronger global exploration ability of AEDBO compared to other algorithms.

(2) Experimental results on the multi-peak test functions ~ show that AEDBO has better performance compared to the other seven algorithms. In $f_5(x)$ and $f_6(x)$, AEDBO converges to the optimum in the fastest speed, and the convergence speed on $f_7(x)$ is slightly behind SSA. The results show that AEDBO performs better in terms of optimization seeking accuracy and robustness. It can converge to the global optimum quickly and effectively avoiding the problem of local optimality.

(3) The experimental results on the fixed dimensional multi-peaked test functions $f_8(x)$ ~ $f_{15}(x)$ show that AEDBO has better performance in dealing with high dimensional problems. Except for the lagging convergence speed on $f_{10}(x)$, AEDBO has a faster convergence speed on the other test functions. Except for the overall performance on $f_{11}(x)$ which is not as good as SSA, AEDBO has better performance, and especially



(a) $f_1(x)$ (b) $f_2(x)$ (c) $f_3(x)$

(d) $f_4(x)$ (e) $f_5(x)$ (f) $f_6(x)$

(g) $f_7(x)$ (h) $f_8(x)$ (i) $f_9(x)$

(j) $f_{10}(x)$ (k) $f_{11}(x)$ (l) $f_{12}(x)$

(m) $f_{13}(x)$ (n) $f_{14}(x)$ (o) $f_{15}(x)$

Fig. 6. Convergence curves for different test functions.

on $f_{13}(x)$, $f_{14}(x)$,$f_{15}(x)$, AEDBO has much better performance than other algorithms. The results show that search capability of AEDBO is greatly improved.

Ablation experimental analysis.

To further validate the algorithm's effectiveness, ablation experiments were conducted on AEDBO. There are four variants in the ablation experiments, DBO1, DBO2, DBO3, and DBO4. DBO1 combines improved Tent chaotic mapping and center-of-mass inverse learning with DBO. DBO2 combines adaptive inertia weight factors with DBO. DBO3 combines population adaptive scaling with DBO. DBO4 combines adaptive Cauchy-Gaussian

Function		PSO	FOA	GWO	WOA	HHO	SSA	DBO	AEDBO
$f_1(x)$	Mean	630.1029	0.0009	1.29E-27	4.10E-74	1.97E-94	3.15E-53	1.21E-106	**6.29E-200**
	Std	316.7393	2.73E-05	2.19E-27	1.43E-73	1.39E-93	2.23E-52	8.57E-106	**0.00E + 00**
$f_2(x)$	Mean	8.1367	0.2590	9.16E-17	3.28E-51	1.03E-49	6.70E-30	3.35E-54	**1.78E-94**
	Std	2.0742	0.0028	7.20E-17	1.16E-50	6.25E-49	4.55E-29	1.87E-53	**1.26E-93**
$f_3(x)$	Mean	8943.2227	0.2475	1.40E-05	4.33E + 04	2.34E-74	1.76E-25	2.34E-50	**9.43E-166**
	Std	4649.0802	0.0075	3.08E-05	1.39E + 04	1.63E-73	1.21E-24	1.66E-49	**0.00E + 00**
$f_4(x)$	Mean	7.1745	0.0084	9.11E-07	4.45E + 01	1.41E-48	1.77E-28	6.62E-51	**1.47E-87**
	Std	1.7255	0.0003	1.09E-06	2.73E + 01	8.55E-48	1.07E-27	3.93E-50	**1.00E-86**
$f_5(x)$	Mean	-2593.9256	-80.7111	-6165.7153	-10,408.3878	-12,550.7311	-8599.2955	-8493.3766	**-12,568.3969**
	Std	346.4069	63.8502	748.5036	1684.0398	124.6547	492.6521	2018.7956	**3.1265**
$f_6(x)$	Mean	6.3330	0.0333	1.04E-13	4.37E-15	**8.88E-16**	**8.88E-16**	9.59E-16	**8.88E-16**
	Std	1.1588	0.0004	2.28E-14	2.64E-15	**0.00E + 00**	**0.00E + 00**	5.02E-16	**0.00E + 00**
$f_7(x)$	Mean	245.5143	5.01E-06	5.10E-03	1.41E-02	**0.00E + 00**	**0.00E + 00**	1.23E-03	**0.00E + 00**
	Std	19.1067	4.17E-07	1.23E-02	4.28E-02	**0.00E + 00**	**0.00E + 00**	8.68E-03	**0.00E + 00**
$f_8(x)$	Mean	1.6498	12.6705	4.4034	3.2345	1.4529	4.8965	1.3940	**1.0179**
	Std	1.3744	**2.01E-15**	4.2952	3.4793	1.1995	5.1999	1.0377	**0.1406**
$f_9(x)$	Mean	0.0012	0.0003	0.0040	0.0008	0.0004	0.0003	0.0009	**0.0003**
	Std	0.0040	2.73E-06	7.74E-03	4.89E-04	1.63E-04	5.16E-05	3.88E-04	**1.52E-06**
$f_{10}(x)$	Mean	**3.0000**	84.1097	**3.0000**	**3.0000**	**3.0000**	3.5400	**3.0000**	**3.0000**
	Std	1.37E-15	0.1443	4.01E-05	0.0002	**0.00E + 00**	3.8184	**0.00E + 00**	3.00E-15
$f_{11}(x)$	Mean	-3.8623	-3.8608	-3.8616	-3.8569	-3.8599	**-3.8628**	-3.8614	-3.8614
	Std	0.0019	0.0013	0.0023	0.0077	0.0040	**9.54E-16**	0.0029	0.0031
$f_{12}(x)$	Mean	-3.1861	-3.2693	-3.2614	-3.2504	-3.1031	-3.2792	-3.2273	**-3.3063**
	Std	0.1438	0.0561	0.0797	0.1077	0.1116	0.0576	0.1018	**0.0498**
$f_{13}(x)$	Mean	-5.1524	-4.7391	-9.1147	-7.2798	-5.1527	-8.2526	-6.9028	**-10.1459**
	Std	3.3836	0.5312	2.2777	2.8092	0.7108	2.4656	2.5975	**0.0356**
$f_{14}(x)$	Mean	-5.9464	-4.7231	-10.0358	-7.9733	-5.2653	-8.9174	-8.6563	**-10.1903**
	Std	3.4773	**0.5581**	1.4853	3.0919	0.9011	2.4065	2.7370	1.0522
$f_{15}(x)$	Mean	-6.0885	-4.7112	-9.8821	-7.2692	-5.1651	-9.6711	-8.4609	**-10.4282**
	Std	3.5883	**0.7590**	2.2348	3.5204	0.7878	2.0027	2.8455	0.7648

Table 2. Statistical results of different algorithms.

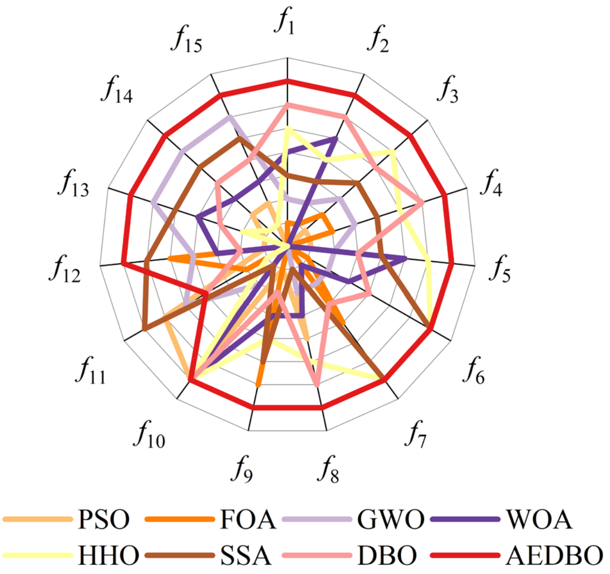


Fig. 7. Performance comparison ranking.

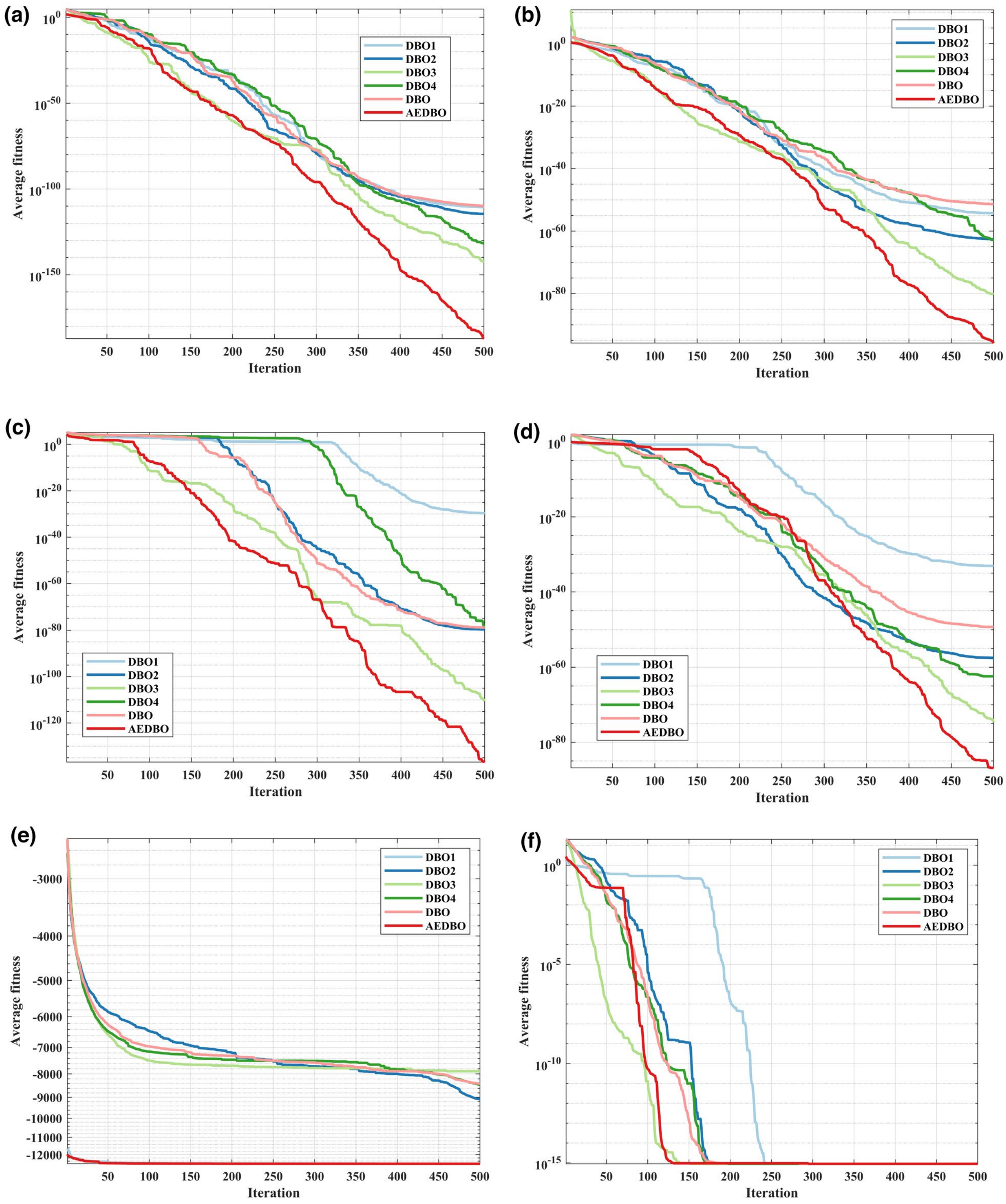


Fig. 8. Convergence curves of different test functions in ablation experiments.

mutation with DBO. The rest of the settings are the same as in Section "Comparison of AEDBO with Other swarm intelligence optimization algorithms".

Figure 8 shows the convergence curves of AEDBO with the 4 algorithm variants on 15 test functions. Table 3 shows the metrics statistics of AEDBO and the 4 algorithm variants on 15 test functions. Figure 9 shows the performance ranking of different algorithms in ablation experiments on the test functions.

Based on the experimental results the following analysis can be performed:

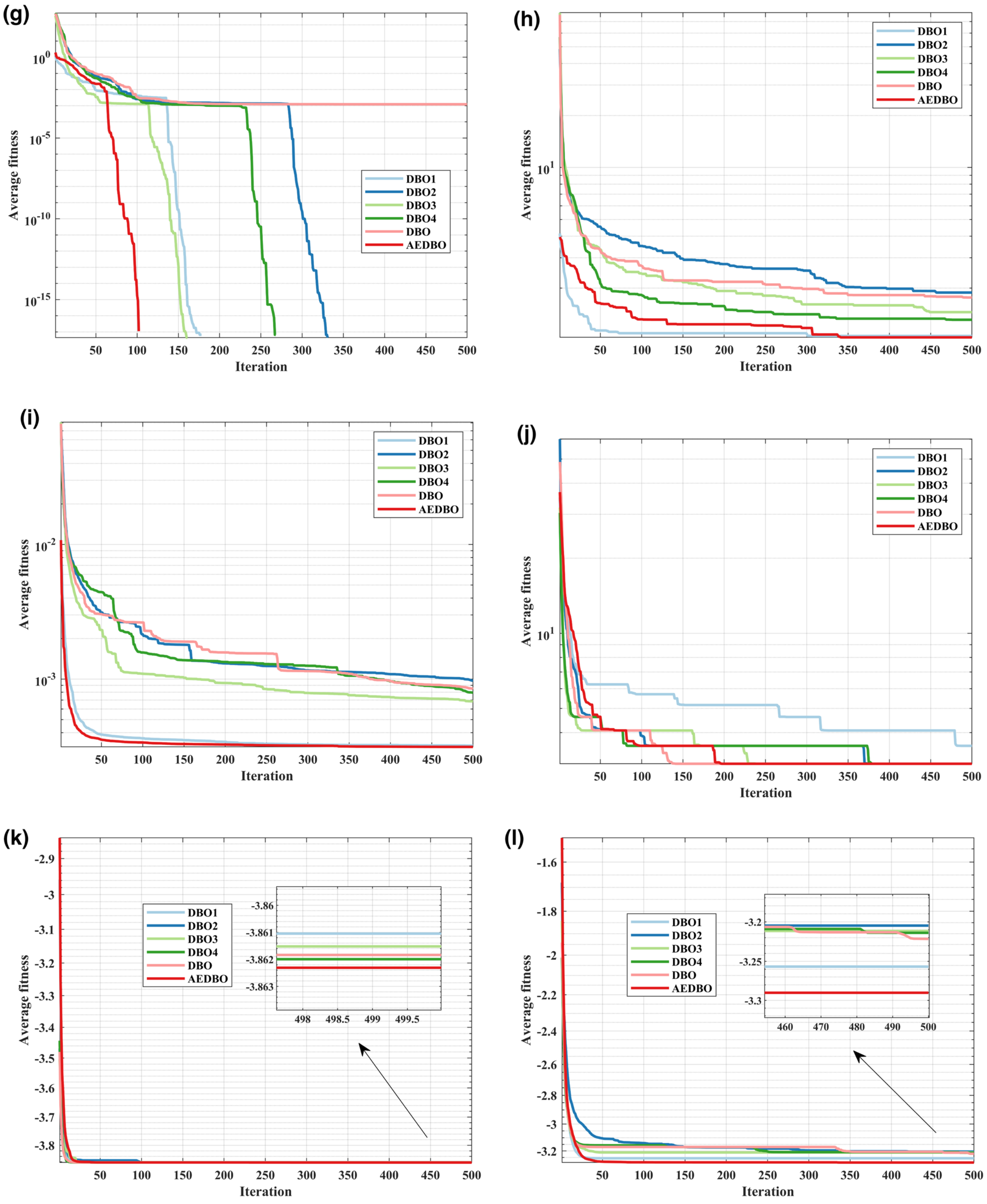


Fig. 8. (continued)

(1) The results of the ablation experiments on the single-peak test functions $f_1(x) \sim f_4(x)$ show that AEDBO improves the algorithm's global optimization-seeking ability, and the population adaptive ratio has the greatest impact on the algorithm's global optimization-seeking ability. In the pre-iteration period of $f_1(x)$, DBO3 and AEDBO have similar performance, and in the pre-iteration period of $f_4(x)$, AEDBO is slightly weaker than DBO3, but in the late period of the iteration, AEDBO performs better than DBO3 on all test functions. The experimental results verify the improvement of the population adaptive scaling strategy on the optimization seeking ability of DBO.

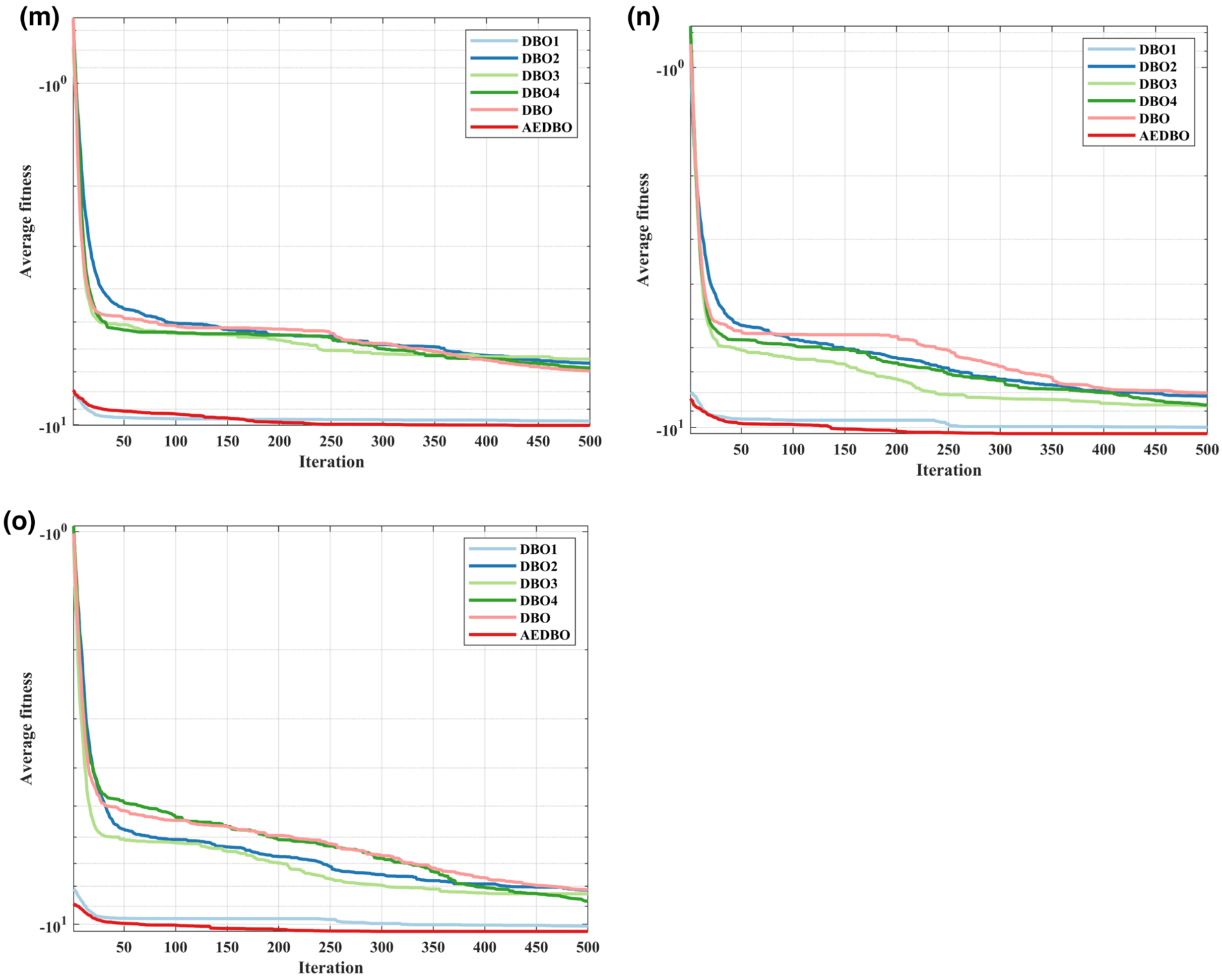


Fig. 8. (continued)

(2) The experimental results on multi-peak test functions $f_5(x) \sim f_7(x)$ show that AEDBO effectively solves the problem that DBO easily falls into local optimality. In $f_5(x)$, DBO, DBO3 and DBO4 all fall into the local optimum, and DBO1 and AEDBO can quickly jump out of the local optimum and converge to the optimum. In $f_6(x)$ and $f_7(x)$, DBO3 has similar performance to AEDBO. The results show that the population initialization method and the population adaptive scaling strategy based on improved Tent chaotic mapping and center-of-mass inverse learning improve the ability of DBO to avoid local optimum, followed by adaptive inertia weighting.

(3) The experimental results on fixed-dimensional multi-peak test functions $f_8(x) \sim f_{15}(x)$ show that AEDBO improves the ability of DBO to handle high-dimensional problems. On $f_8(x)$, $f_9(x)$, $f_{12}(x)$, $f_{13}(x)$, and $f_{15}(x)$, the optimization seeking ability of DBO1 is comparable to that of AEDBO. On $f_{10}(x)$, DBO3 and DBO4 have better optimization accuracy than AEDBO in the early stage. The results show that AEDBO's ability to handle high dimensions is enhanced.

(4) The experimental results show that in most of the test functions, each strategy improves the algorithm's optimization accuracy. Only in $f_4(x)$ and $f_6(x)$, the population initialization problem resulted in a decrease in the accuracy of the algorithm.

(5) Except f_4, DBO1 outperforms DBO in all test functions, and the results show that the population initialization strategy incorporating improved Tent chaos mapping and center-of-mass adversarial inverse learning can effectively improve the performance of the algorithm; Except f_9、f_{12}、f_{13}, DBO2 outperforms DBO in all test functions, and the results show that the proposed adaptive convergence factor can effectively improve the performance of the algorithm; Except f_5、f_{11}、f_{12}、f_{15}, DBO3 outperforms DBO, and the results show that the proposed adaptive nonlinear rolling dung beetle decreasing number model can effectively improve the performance of the algorithm; Except f_6、f_{11}、f_{12}、f_{13}, DBO4 outperforms DBO in the test functions, and the results show that the proposed adaptive Cauchy-Gaussian mutation strategy can effectively improve the performance of the algorithm.

function		**DBO1**	**DBO2**	**DBO3**	**DBO4**	**DBO**	**AEDBO**
$f_1(x)$	Mean	7.52E-111	1.16E-112	7.65E-150	1.30E-134	2.18E-105	**5.76E-188**
	Std	5.32E-110	6.08E-112	5.41E-149	9.17E-134	1.54E-104	**0.00E+00**
$f_2(x)$	Mean	1.75E-54	4.38E-57	2.58E-84	3.45E-67	2.80E-53	**1.57E-84**
	Std	1.20E-53	3.08E-56	1.82E-83	2.44E-66	1.93E-52	**1.11E-83**
$f_3(x)$	Mean	3.86E-54	2.13E-60	6.77E-119	2.34E-91	2.05E-45	**2.23E-166**
	Std	2.63E-53	1.50E-59	4.78E-118	1.65E-90	1.45E-44	**0.00E+00**
$f_4(x)$	Mean	3.12E-32	6.25E-50	3.12E-64	7.08E-70	3.16E-49	**2.48E-82**
	Std	2.03E-31	4.42E-49	2.21E-63	3.54E-69	2.23E-48	**1.72E-81**
$f_5(x)$	Mean	**-12,568.7418**	-9787.4516	-7899.3983	-8535.0550	-8365.5957	-12,566.8179
	Std	**2.25E+00**	2.11E+03	1.28E+03	1.82E+03	1.64E+03	9.27E+00
$f_6(x)$	Mean	**8.88E-16**	**8.88E-16**	**8.88E-16**	1.10E-15	9.59E-16	**8.88E-16**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.52E-16	5.02E-16	**0.00E+00**
$f_7(x)$	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.40E-09	**0.00E+00**
	Std	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.41E-08	**0.00E+00**
$f_8(x)$	Mean	1.1933	1.6099	1.6487	1.6068	1.8248	**1.0775**
	Std	1.3810	1.5707	1.7003	2.0039	2.0576	**0.2724**
$f_9(x)$	Mean	3.11E-04	9.63E-04	8.04E-04	8.42E-04	8.34E-04	**3.10E-04**
	Std	2.38E-05	5.78E-04	4.56E-04	4.30E-04	4.09E-04	**1.11E-05**
$f_{10}(x)$	Mean	**3.0000**	**3.0000**	**3.0000**	**3.0000**	**3.0000**	**3.0000**
	Std	4.76E-14	4.00E-15	2.81E-15	2.83E-15	3.08E-15	**2.49E-15**
$f_{11}(x)$	Mean	**-3.8622**	-3.8618	-3.8610	-3.8611	-3.8617	-3.8620
	Std	**2.16E-03**	2.59E-03	3.30E-03	3.29E-03	2.76E-03	2.39E-03
$f_{12}(x)$	Mean	-3.2871	-3.2228	-3.2301	-3.2131	-3.2392	**-3.2930**
	Std	**0.0738**	0.1135	0.1097	0.1045	0.1138	0.0854
$f_{13}(x)$	Mean	-9.9493	-6.8069	-7.2820	-6.9365	-7.1881	**-10.1506**
	Std	1.0091	2.5789	2.7206	2.4452	2.6472	**0.0172**
$f_{14}(x)$	Mean	-9.55796	-8.03272	-8.30672	-8.78507	-8.02025	**-10.4029**
	Std	1.9549	2.6762	2.7321	2.5020	2.8936	**1.13E-11**
$f_{15}(x)$	Mean	-10.1063	-9.0900	-7.5429	-8.8746	-8.1068	**-10.4121**
	Std	1.4725	2.3756	3.2891	2.5293	3.1016	**0.76526**

Table 3. Statistical results in ablation experiments.

Optimal observation trajectory of the airborne unmanned platform

Non-hazardous area

In this part, AEDBO is used to realize the control of the airborne unmanned platform to obtain the optimal flight trajectory. Table 4 shows the relevant parameters set of the unmanned platform during the simulation experiment.

Two typical postures are selected in a no-hazardous-area environment: Posture 1- the maritime cooperative and airborne unmanned platform and the target are on the same side, with the airborne initial position at [-5000,-10,000,3500]m; Posture 2 -the maritime cooperative and airborne unmanned platform and target are oppositely positioned, with the airborne initial position at [-5000,10,000,3500]m.

Figure 10 shows airborne unmanned platform trajectory optimization under Posture 1, including the overall posture, line-of-sight angle change, tracking error, and state vector / control volume change of the airborne unmanned platform under Posture 1, respectively. Figure 11 shows airborne unmanned platform trajectory optimization under Posture 2, including the overall posture, line-of-sight angle change, tracking error, and state vector/control volume change of the airborne unmanned platform under Posture 2, respectively.

From Fig. 10, the airborne unmanned platform quickly flies rearward of the maritime cooperative unmanned platform at maximum speed, with the line-of-sight angle gradually approaching 0 aligning the three on a horizontal plane. It maintains this observation configuration; thereafter, forming an optimal observation configuration around 77s; Compared to the initial tracking error of 50.51m, the error is 12.29m by 77s improving accuracy over 4 times compared to stationary observation. From Fig. 11, the airborne unmanned platform flies towards the marine target at maximum speed, with the line-of-sight angle gradually approaching 180 aligning the three on a horizontal plane. It maintains this observation configuration; thereafter, forming an optimal observation configuration around 200s. Compared to the initial tracking error of 36.88m, the error is 3.58m by 200s improving accuracy over 10 times compared to stationary observation.

From Fig. 10 and Fig. 11, the marine target tracking error decreases rapidly during observation configuration formation, and is smaller and slower after formation proving suitable observation configuration directly affects tracking accuracy. The above results show AEDBO has good trajectory optimization ability and optimal observation configuration formation effectively improves tracking accuracy.

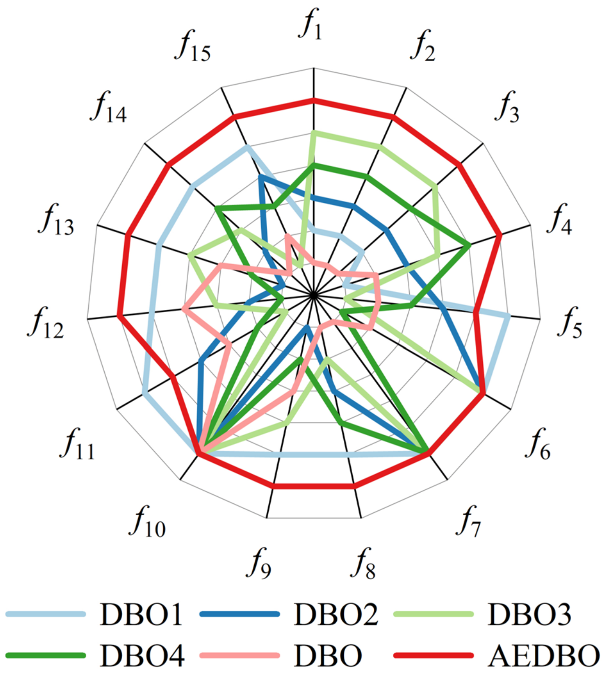


Fig. 9. Ranking of ablation experiments.

Parameter	Value
Initial heading of the airborne unmanned platform/(°)	0
Maritime cooperative unmanned platform location/(m)	[0,0,0]
Marine target position/(m)	[0,0,80000]
Speed control delay time constant	0.8
Angular velocity control delay time constant	0.8
Airborne unmanned platform cruising speed/(m/s)	100
Maximum speed change range/(m/s)	10
Maximum angular velocity/(°/s)	20

Table 4. Parameters related to unmanned platforms.

Hazardous area

Given the airborne unmanned platform may be impacted by hazardous areas during flight, this section simulates and analyzes the airborne unmanned platform in environments with hazardous areas. Two typical postures are selected in the environment with hazardous area: Posture 3- the maritime cooperative unmanned platform and target are on the same side of the unmanned platform and there is a hazardous area from 1 to 4. The initial position of the unmanned platform is [-5000,-10,000,3500]m; Posture 4- the maritime cooperative unmanned platform and target are on the opposite side of the unmanned platform and there is a hazardous area from 5 to 8. The initial position of the unmanned platform is [-5000,10,000,3500]m.

Hazardous area 1 is an ellipse with center point at (-2700m, -12000m), and horizontal radius is 400m, vertical radius is 500m; Hazardous area 2 is an ellipse with center point at (-4000m, -14000m), and horizontal radius is 1000m, vertical radius is 1200m; Hazardous area 3 is a rectangular area with side lengths of 1000m and 2,000m rectangular area, the lower right point is located at (500m, -15,000m); Hazardous area 4 is an elliptical area, the center point is located at (0m, -20,000m), the horizontal radius is 400m, and the vertical radius is 500m; Hazardous area 5 is an elliptical area, the center point is located at (-4,000m, 13,000m), the horizontal radius is 400m, the vertical radius is 500m; Hazardous area 6 is an elliptical area, the center point is located at (-3000m, 16000m), the horizontal radius is 1000m, and the vertical radius is 1200m; Hazardous area 7 is a rectangular area with the side lengths of 1000m and 2000m respectively, and the lower right point is located at (0m, 14000m); Hazardous area 8 is an elliptical area with center point at (0m, 21000m), and horizontal radius is 400m, vertical radius is 500m. The marine target is moving in the north direction with a speed of 5m/s, and the other simulation parameters are the same as in 5.2.1.

Figure 12 shows the simulation result of airborne unmanned platform trajectory optimization under Posture 3, including the overall posture diagram, line-of-sight angle change diagram, tracking error diagram, and the change process of state vector and control quantities of the airborne unmanned platform respectively. Figure 13

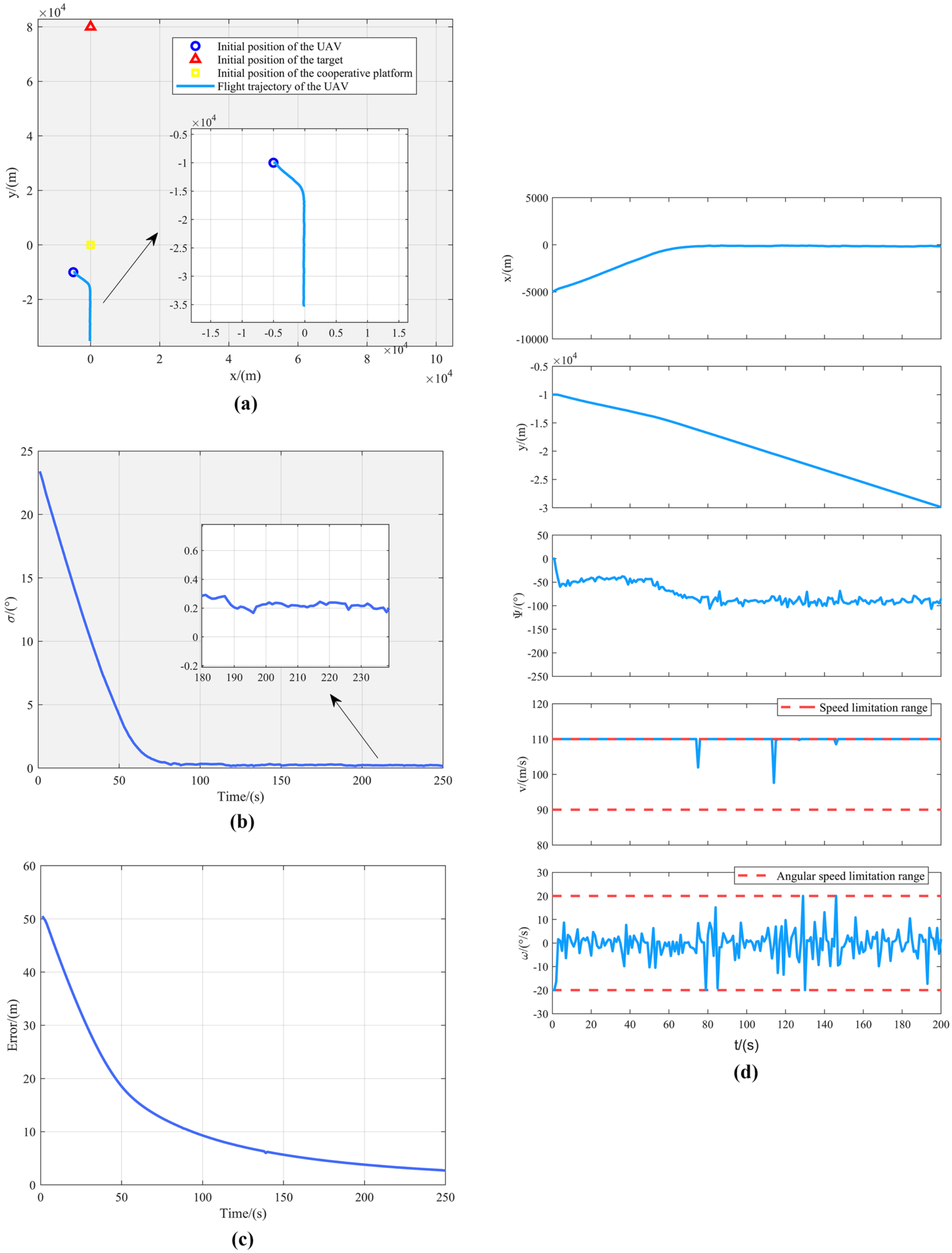


Fig. 10. Simulation result of the airborne unmanned platform optimization trajectory in posture 1, (**a**) Overall posture, (**b**)Line-of-sight angle, (**c**) Tracking error, (**d**) State vector and control.

shows the simulation result of the airborne unmanned platform trajectory optimization under posture 4, including the overall posture diagram, the line-of-sight angle change diagram, the tracking error diagram, and the state vector and control volume change process of the airborne unmanned platform trajectory optimization under posture 4 respectively.

From Fig. 12, it can be seen that the airborne unmanned platform can effectively avoid hazardous area and still form the optimal observation configuration under the influence of hazardous area. In the influence of hazardous areas in 21 ~ 25s, 53 ~ 57s, 102 ~ 153s, the line-of-sight angle of the airborne unmanned platform

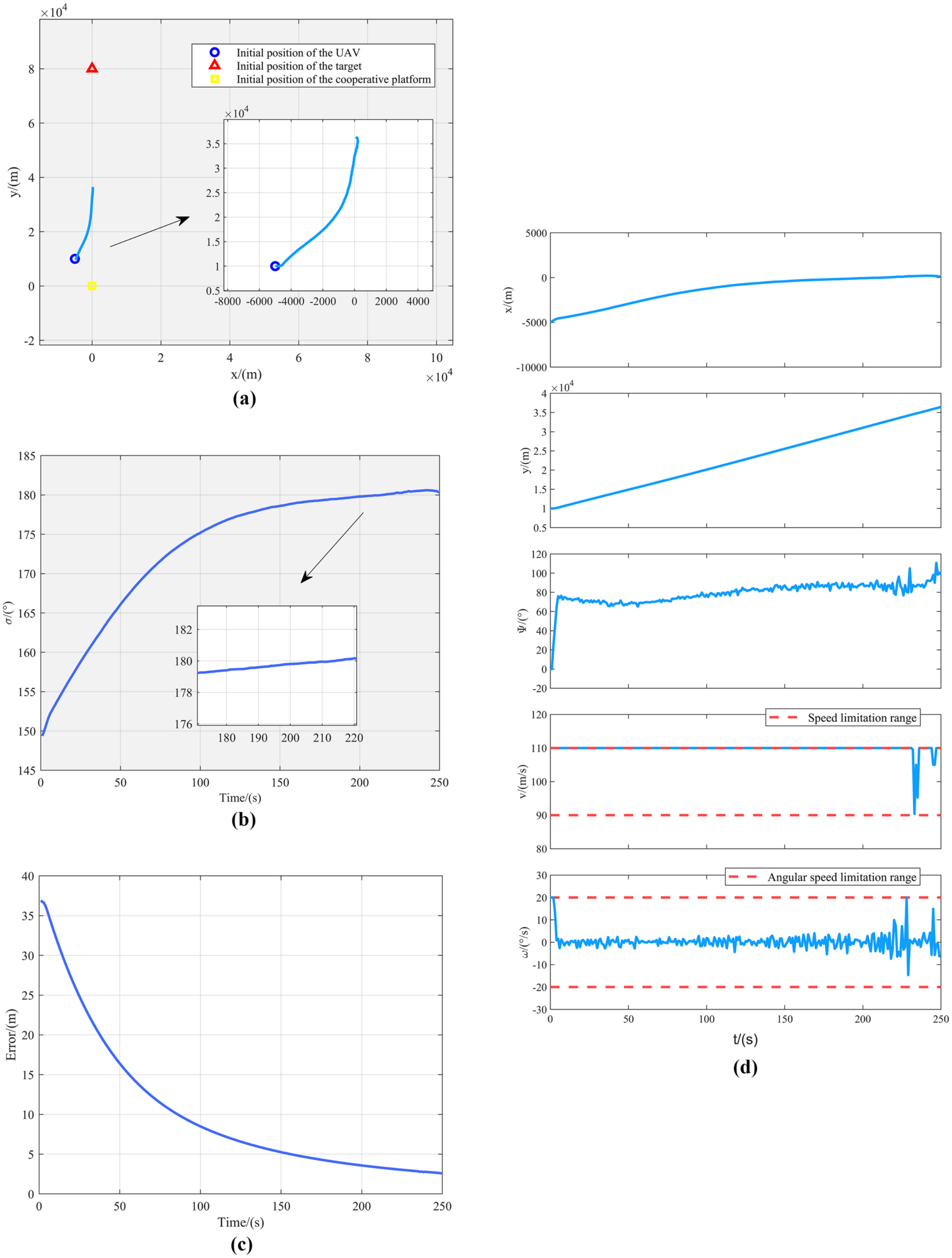


Fig. 11. Simulation result of the airborne unmanned platform trajectory optimization in posture 2. (**a**) Overall posture, (**b**) Line-of-sight angle, (**c**) Tracking error, (**d**) State vector and control.

tends to slow down towards 0, and even increases in the opposite direction in 102 ~ 153s, and the reduction speed of the tracking error slows down accordingly, but the overall still shows a decreasing trend, because in the process of obstacle avoidance, the tracking error has a smaller influence in the objective function, and the airborne unmanned platform makes obstacle avoidance as the main objective. From Fig. 13, it can be seen that the airborne unmanned platform can effectively avoid the hazardous area and still form the optimal observation configuration in the influence of hazardous area. In the influence of hazardous area, in 33 ~ 45s and 108 ~ 121s, the line-of-sight angle of the airborne unmanned platform tends to slow down towards 180, and even increases

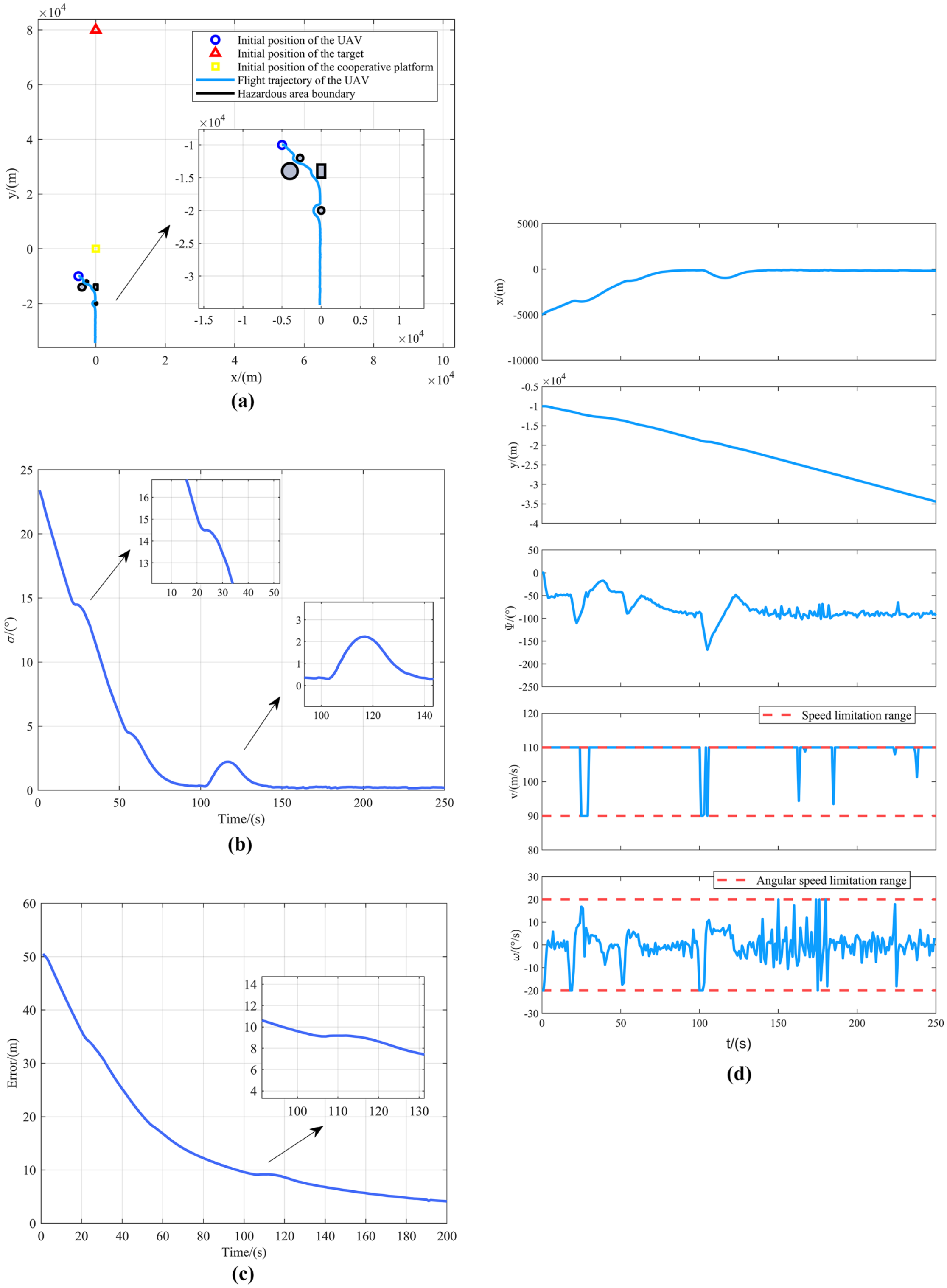
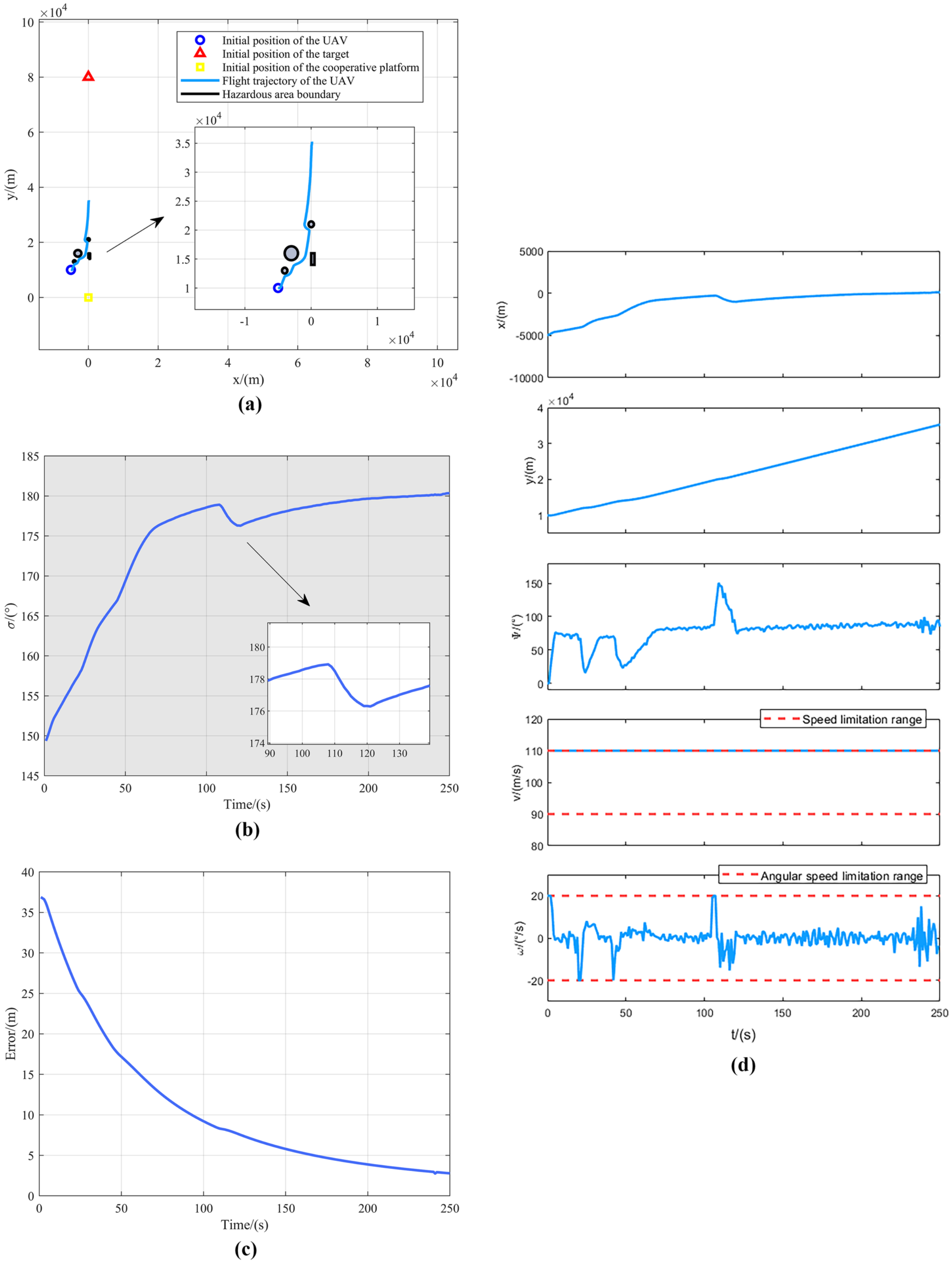


Fig. 12. Simulation result of the airborne unmanned platform trajectory optimization in posture 3. (**a**) Overall posture, (**b**) Line-of-sight angle, (**c**) Tracking error, (**d**) State vector and control.

in the opposite direction in 108 ~ 121s, and the reduction of tracking error slows down accordingly, but the overall still shows a decreasing trend, because in the process of obstacle avoidance, the tracking error has a smaller influence in the objective function, and the airborne unmanned platform takes obstacle avoidance as the primary objective.



(a)

(b)

(c)

(d)

Fig. 13. Simulation result of the airborne unmanned platform trajectory optimization in posture 4. (**a**) Overall posture, (**b**) Line-of-sight angle, (**c**) Tracking error, (**d**) State vector and control.

Overall, after considering the hazardous area, the airborne unmanned platform still forms the optimal observation configuration, and the tracking error keeps decreasing, proving the effectiveness of the algorithms in complex environments.

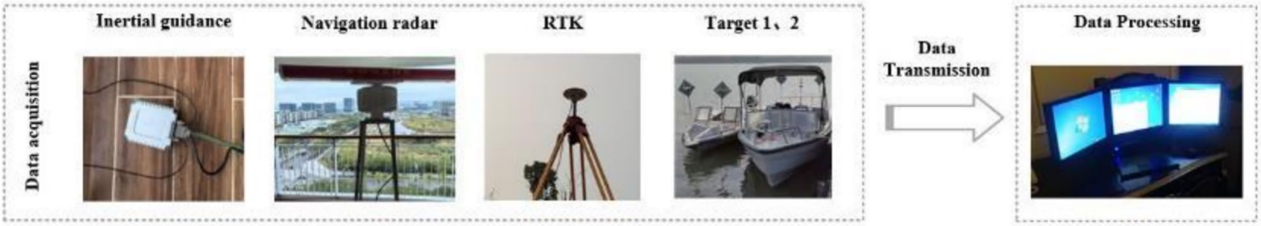


Fig. 14. Schematic diagram of the practical experiment.

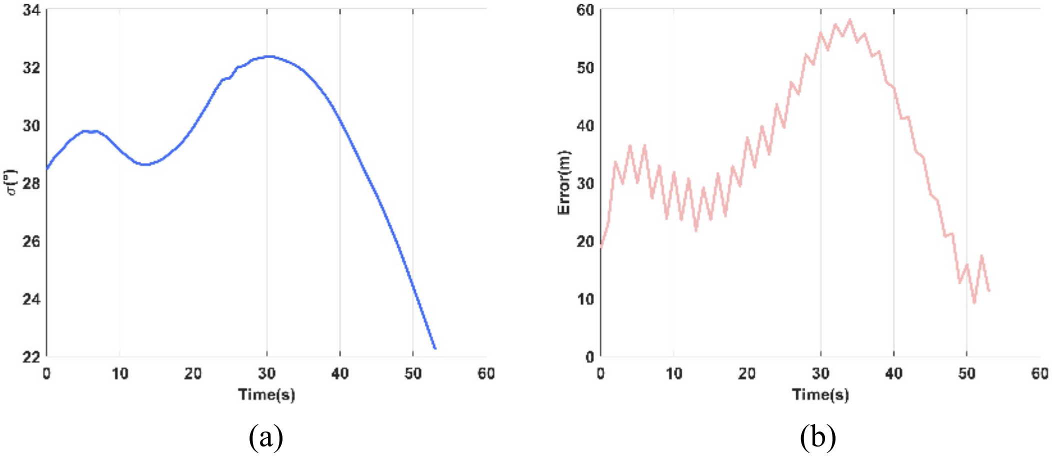


Fig. 15. Data 1 result. (**a**) Line-of-sight angle, (**b**) Tracking error.

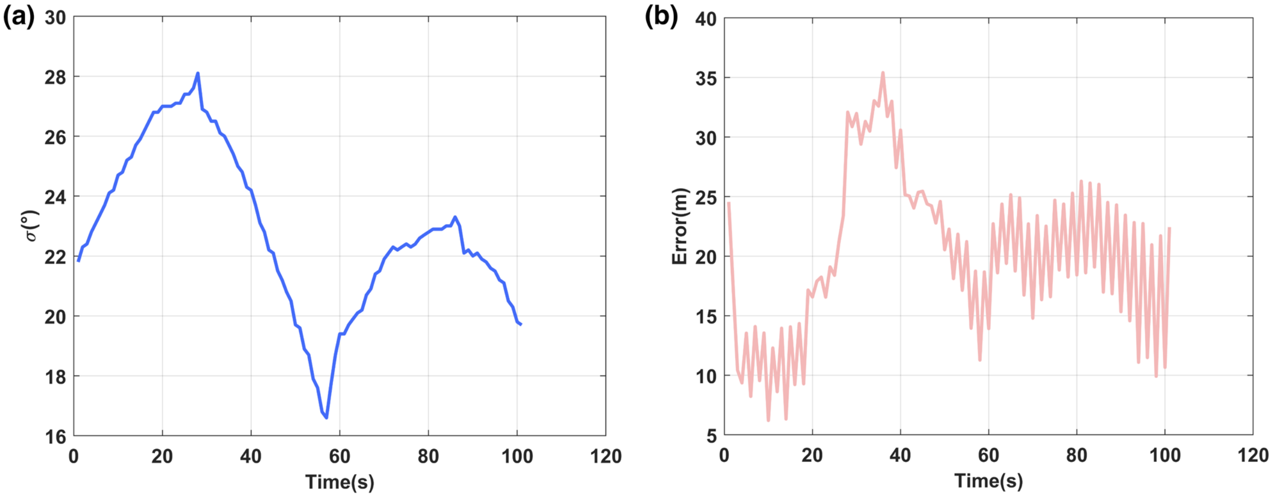
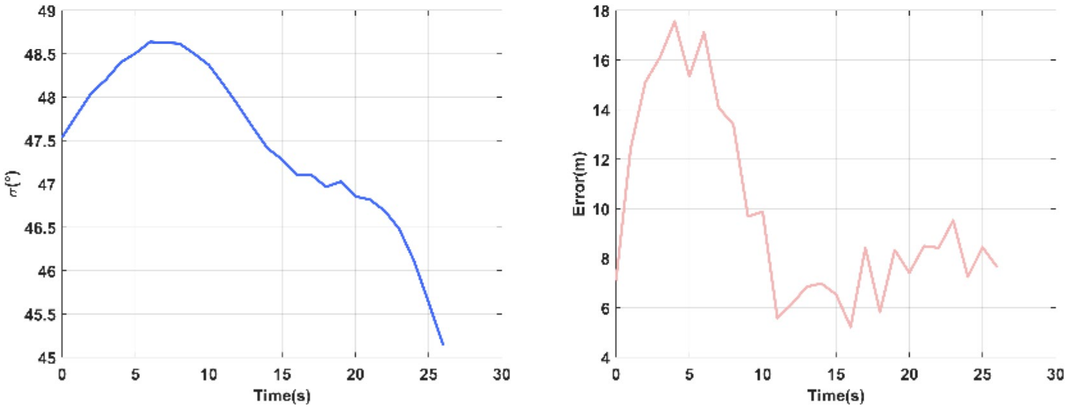


Fig. 16. Data 2 result. (**a**) Line-of-sight angle, (**b**) Tracking error.

Practical experiment

We developed a realistic scenario to collect data for verifying the practical application value. Due to the high cost of UAV radar observation, we chose inertial guidance plus radar to simulate UAV observation Two small boats served as the cooperation platform and the target moving on the lake, respectively. The high-precision positions of the observation platform and small boats were acquired by RTK measurements, as shown in Fig. 14 The processed data results are shown in Figs. 15, 16, 17.

The line-of-sight angle changes of the three datasets range from 0 to 90°, and the overall trend of line-of-sight angle changes is consistent with the trend of tracking error, which is manifested in the fact that the closer to 0°



(a) (b)

Fig. 17. Data 3 result. (**a**) Line-of-sight angle, (**b**) Tracking error.

the smaller the tracking error is, and the closer to 90° the larger the tracking error is. However, there are certain fluctuations present, which are caused by external factors such as environmental conditions and the hardware conditions of the equipment. In summary, the conclusion of the practical experiment is consistent with the conclusion of the simulation experiment.

Conclusion and future work

This paper takes air unmanned platform and maritime unmanned cooperative platform synergy as the research body to improve target tracking accuracy. It analyzes the optimal observation configuration based on error reduction algorithm and proposes AEDBO for optimal control of airborne unmanned platform. The following conclusions are drawn:

(1) Error analysis based on error reduction shows the tracking error mainly consists of three parts: sensor error, attitude error, and sensor and attitude coupling error. The latter is small compared to the first two errors, and can be ignored with limited computational resource.
(2) Optimal observation configuration considering azimuthal systematic error as the main error indicates all three platforms- the air unmanned platform, maritime unmanned cooperative platform, and target should be in a horizontal straight line. This means the observation line-of-sight angle of the airborne unmanned platform is 0 or π.
(3) AEDBO is proposed for control of unmanned airborne platform. We propose a population initialization method integrating improved Tent chaos mapping and center-of-mass contrastive inverse learning, an adaptive convergence factor, an adaptive nonlinear ball-rolling dung beetle number decreasing model, and an adaptive Cauchy-Gaussian mutation strategy to enhance DBO. This allows realization of the optimal flight trajectory of the airborne unmanned platform. Comparison with other swarm intelligence algorithms and ablation experiments prove the effectiveness of each improvement method and the superiority of algorithm performance;
(4) The proposed AEDBO effectively optimizes the trajectory of airborne unmanned platforms in different initial conditions and environments to achieve optimal observation configuration.

Future steps will focus on:

(1) Introducing more systematic error to further investigate optimal observation configuration under complex error conditions;
(2) Introducing more unmanned platforms to study trajectory optimization and marine target tracking under complex environments with multiple airborne unmanned platforms, multiple maritime cooperative unmanned platforms, and multiple marine targets.

Data availability

The datasets generated and/or analysed during the current study are not publicly available due to privacy issues. If someone wants to obtain data from this study, please contact Qiuyang Dai.

Received: 29 September 2024; Accepted: 11 June 2025
Published online: 02 July 2025

References

1. Wang, P., Liu, X. & Song, A. Actuation and locomotion of miniature underwater Robots: A survey. *Engineering* https://doi.org/10.1016/j.eng.2024.10.022 (2024).

2. Liang, H., Li, H., Shi, Y., Constantinescu, D. & Xu, D. Energy-efficient integrated motion planning and control for unmanned surface vessels. *IEEE Trans. Control Syst. Technol.* **32**, 250–257. https://doi.org/10.1109/tcst.2023.3292466 (2024).
3. Liang, H., Yu, J. & Li, H. Adaptive trajectory tracking control for small unmanned underwater vehicles with prescribed performance and dynamic compensation. *IEEE Trans. Ind. Electr.* https://doi.org/10.1109/tie.2024.3485626 (2024).
4. Hongtao, L., Fu, Y., Gao, J. & Cao, H. Finite-time velocity-observed based adaptive output-feedback trajectory tracking formation control for underactuated unmanned underwater vehicles with prescribed transient performance. *Ocean Eng.* **233**, 109071. https://doi.org/10.1016/j.oceaneng.2021.109071 (2021).
5. Bu, L. & Li, H. in *2024 IEEE 18th International Conference on Control & Automation (ICCA)* 930–935 (2024).
6. Dai, Q., Lu, F. & Xu, J. A novel sea target tracking algorithm for multiple unmanned aerial vehicles considering attitude error in low-precision geodetic coordinate environments. *Aerospace* https://doi.org/10.3390/aerospace11020162 (2024).
7. Kim, S., Tadiparthi, V. & Bhattacharya, R. Computationally efficient attitude estimation with extended H2 filtering. *J. Guid. Control. Dyn.* **44**, 418–427. https://doi.org/10.2514/1.G005140 (2021).
8. Wenkang, W., Jingan, F., Bao, S. & Xinxin, L. Vehicle state estimation using interacting multiple model based on square root cubature Kalman filter. *Appl. Sci.* https://doi.org/10.3390/app112210772 (2021).
9. Cui, Y., You, H. E., Tang, T. & Liu, Y. A new target tracking filter based on deep learning. *Chinese Journal of Aeronautics* (2021).
10. liang-Huai, P. Robust Least-Squares Bias Estimation for Radar Detecting Biases and Attitude Biases. *2013 2nd International Conference on Measurement, Information and Control* (2013).
11. Wei, Z., Wei, S., Luo, F., Yang, S. & Wang, J. in *IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*. (2019)
12. He, Y., Zhu, H. W. & Tang, X. M. Joint systematic error estimation algorithm for radar and automatic dependent surveillance broadcasting. *IET Radar Sonar Navig.* **7**, 361–370. https://doi.org/10.1049/iet-rsn.2012.0199 (2013).
13. Shi, H.-R., Lu, F.-X. & Wu, L. Cooperative trajectory optimization of UAVs in approaching stage using feedback guidance methods. *Def. Technol.* https://doi.org/10.1016/j.dt.2022.03.013 (2022).
14. Shi, H., Lu, F., Wang, H. & Xu, J. Optimal observation configuration of UAVs based on angle and range measurements and cooperative target tracking in three-dimensional space. *J. Syst. Eng. Electron.* **31**, 996–1008. https://doi.org/10.23919/jsee.2020.000074 (2020).
15. Hung, H.-A., Hsu, H.-H. & Cheng, T.-H. IEEEexample:BSTcontrol optimal sensing for tracking task by heterogeneous multi-UAV systems. *IEEE Trans. Control Syst. Technol.* **32**, 282–289. https://doi.org/10.1109/tcst.2023.3298487 (2024).
16. Chen, Y., Yu, H., Li, J., Ji, F. & Chen, F. TOA-based direct localization in shallow water multipath environments: CRLB analysis and optimal sensor deployment. *Ocean Eng.* https://doi.org/10.1016/j.oceaneng.2023.116556 (2024).
17. Wu, L., Sahu, N., Xu, S., Babu, P. & Ciuonzo, D. Optimization based sensor placement for multi-target localization with coupling sensor clusters. *Ieee Trans. Signal Inform. Process. over Netw.* **9**, 596–611. https://doi.org/10.1109/tsipn.2023.3307899 (2023).
18. Kim, S., Oh, H. & Tsourdos, A. Nonlinear model predictive coordinated Standoff tracking of a moving ground vehicle. *J. Guid. Control. Dyn.* **36**, 557–566. https://doi.org/10.2514/1.56254 (2013).
19. Wang, X. et al. A fixed-wing UAV formation algorithm based on vector field guidance. *IEEE Trans. Autom. Sci. Eng.* **20**, 179–192. https://doi.org/10.1109/tase.2022.3144672 (2023).
20. Shi, H. R., Lu, F. X., Wu, L. & Xia, J. W. Trajectory optimization of multi-UAVs for marine target tracking during approaching stage. *Math. Probl. Eng.* https://doi.org/10.1155/2022/5472105 (2022).
21. Wu, T. et al. A novel AI-based framework for AoI-optimal trajectory planning in UAV-assisted wireless sensor networks. *IEEE Trans. Wireless Commun.* **21**, 2462–2475. https://doi.org/10.1109/twc.2021.3112568 (2022).
22. Zhang, H. et al. Capacity maximization in RIS-UAV networks: A DDQN-based trajectory and phase shift optimization approach. *IEEE Trans. Wireless Commun.* **22**, 2583–2591. https://doi.org/10.1109/twc.2022.3212830 (2023).
23. Lun, Y., Yao, P. & Wang, Y. Trajectory optimization of SUAV for marine vessels communication relay mission. *IEEE Syst. J.* **14**, 5014–5024. https://doi.org/10.1109/jsyst.2020.2975565 (2020).
24. Griffiths, S. in *AIAA Guidance, Navigation, and Control Conference and Exhibit.*
25. Wang, T., Chen, Y., Liang, J., Wang, C. & Zhang, Y. Combined of vector field and linear quadratic Gaussian for the path following of a small unmanned helicopter. *IET Control Theory Appl.* **6**, 2696–2703. https://doi.org/10.1049/iet-cta.2012.0270 (2012).
26. Kyriakakis, N. A., Marinaki, M., Matsatsinis, N. & Marinakis, Y. Moving peak drone search problem: An online multi-swarm intelligence approach for UAV search operations. *Swarm Evolut. Comput.* https://doi.org/10.1016/j.swevo.2021.100956 (2021).
27. Luo, Q. & Duan, H. Distributed UAV flocking control based on homing pigeon hierarchical strategies. *Aerosp. Sci. Technol.* **70**, 257–264. https://doi.org/10.1016/j.ast.2017.08.010 (2017).
28. Yu, Y. et al. Distributed multi-agent target tracking: A nash-combined adaptive differential evolution method for UAV systems. *IEEE Trans. Veh. Technol.* **70**, 8122–8133. https://doi.org/10.1109/tvt.2021.3091575 (2021).
29. Wang, Y. A. et al. Tracking a dynamic invading target by UAV in oilfield inspection via an improved bat algorithm. *Appl. Soft Comput.* https://doi.org/10.1016/j.asoc.2020.106150 (2020).
30. Lin, J. & Pan, L. Multiobjective trajectory optimization with a cutting and padding encoding strategy for single-UAV-assisted mobile edge computing system. *Swarm Evolut. Comput.* https://doi.org/10.1016/j.swevo.2022.101163 (2022).
31. Xue, J. & Shen, B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *J. Supercomput.* https://doi.org/10.1007/s11227-022-04959-6 (2022).
32. Lynn, N. & Suganthan, P. N. Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. *Swarm Evol. Comput.* **24**, 11–24. https://doi.org/10.1016/j.swevo.2015.05.002 (2015).
33. Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **69**, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007 (2014).
34. Xue, J. & Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. & Control Eng.* **8**, 22–34. https://doi.org/10.1080/21642583.2019.1708830 (2020).
35. Yao, Y. et al. Fast optimization for large scale logistics in complex urban systems using the hybrid sparrow search algorithm. *Int. J. Geogr. Inf. Sci.* **37**, 1420–1448. https://doi.org/10.1080/13658816.2023.2190371 (2023).
36. Mirjalili, S., Mirjalili, S. M. & Hatamlou, A. Multi-Verse Optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **27**, 495–513. https://doi.org/10.1007/s00521-015-1870-7 (2015).
37. Heidari, A. A. et al. Harris hawks optimization: Algorithm and applications. *Futur. Gener. Comput. Syst.* **97**, 849–872. https://doi.org/10.1016/j.future.2019.02.028 (2019).
38. Jiang, H., Deng, J. & Chen, Q. Olfactory sensor combined with chemometrics analysis to determine fatty acid in stored wheat. *Food Control* https://doi.org/10.1016/j.foodcont.2023.109942 (2023).
39. Qin, L., Li, T., Shi, M., Cao, Z. & Gu, L. Internal leakage rate prediction and unilateral and bilateral internal leakage identification of ball valves in the gas pipeline based on pressure detection. *Eng. Fail. Anal.* https://doi.org/10.1016/j.engfailanal.2023.107584 (2023).
40. Li, Y., Sun, K., Yao, Q. & Wang, L. A dual-optimization wind speed forecasting model based on deep learning and improved dung beetle optimization algorithm. *Energy* https://doi.org/10.1016/j.energy.2023.129604 (2024).
41. Zhu, F. et al. Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems. *Expert Syst. Appl.* https://doi.org/10.1016/j.eswa.2023.121219 (2024).
42. Wang, Z. & Shao, P. A multi-strategy dung beetle optimization algorithm for optimizing constrained engineering problems. *Ieee Access* **11**, 98805–98817. https://doi.org/10.1109/access.2023.3313930 (2023).

43. Wang, Z. et al. A quasi-oppositional learning of updating quantum state and Q-learning based on the dung beetle algorithm for global optimization. *Alex. Eng. J.* **81**, 469–488. https://doi.org/10.1016/j.aej.2023.09.042 (2023).
44. Li, Z., Guo, J., Gao, X., Yang, X. & He, Y.-L. A multi-strategy improved sparrow search algorithm of large-scale refrigeration system: Optimal loading distribution of chillers. *Appl. Energy* https://doi.org/10.1016/j.apenergy.2023.121623 (2023).
45. Ma, J., Hao, Z. & Sun, W. Enhancing sparrow search algorithm via multi-strategies for continuous optimization problems. *Inform. Process. & Manag.* https://doi.org/10.1016/j.ipm.2021.102854 (2022).

Acknowledgements

This research is supported by of National Defense Science and Technology Field Foundation of China (2023-JC-JQ-JJ-0388).University Independent Research Project (2025500140).

Author contributions

All authors contributed to the study conception and design. Model construction was performed by J.X. and H.S. Simulation analysis was performed by Q.D. and J.Q. The first draft of the manuscript was written by Q.D. and F.L., and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding

National Defense Science and Technology Field Foundation of China, 2023-JCJQ-JJ-0388. University Independent Research Project, 2025500140.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Q.D.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.