IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 November 2022, accepted 3 December 2022, date of publication 7 December 2022,
date of current version 13 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3227386

RESEARCH ARTICLE

High-Fidelity Decision-Making and Simulation for Cooperative Autonomous Air Combat Considering the Effect of Flight Controller

ZHIHAO WANG[1], **LIANG WANG**[2], **JIANLIANG AI**[1], **AND YIQUN DONG**[1]
[1]Department of Aeronautics and Astronautics, Fudan University, Shanghai 200433, China
[2]Shanghai Institute of Aerospace Systems Engineering, Shanghai 201109, China

Corresponding author: Yiqun Dong (yiqundong@fudan.edu.cn)

This work was sponsored by the Shanghai Sailing Program under Grant No. 20YF1402500, and the Shanghai Natural Science Fund under Grant No. 22ZR1404500.

ABSTRACT In the past decade, how to improve the fidelity of maneuver gaming and the synergy of target allocation has become a key issue in cooperative autonomous air combat researches. To address the problem, this study proposes a maneuver decision-making algorithm based on an optimized dynamic Bayesian network and a target allocation decision-making algorithm based on an optimized hybrid particle swarm optimization. In maneuvering decision-making, the state transition's reliability and the air combat's autonomy are enhanced through considering the effect of sliding mode control. The Bayesian network is improved through introducing a strategy for dynamic prior probability updating. The computation is reduced and the efficiency is increased through pruning the minimax search tree according to visual prediction. In target allocation decision-making, the algorithm's convergence speed is greatly accelerated and the solution's global optimality is improved through introducing immigrant particles. The algorithm's application scope is expanded through proposing a solution principle about unequal quantity combat situations. Furthermore, the end criterion is specially designed to fit real-world combats through introducing a fire control. The simulation results show that the designed decision-making algorithms are more effective in solving the problem of cooperative autonomous air combat, which indicates that the various improvements introduced in this study are reasonable and effective.

INDEX TERMS Cooperative autonomous air combat, maneuver decision-making, target allocation decision-making, sliding mode controller, dynamic Bayesian network, hybrid particle swarm optimization.

I. INTRODUCTION

At the beginning, the advantages of UAVs over manned aircraft give birth to the studies on autonomous air combat. Afterwards, the disadvantages and limitations of a single UAV in combat further promote researches on cooperative autonomous air combat (CAAC). Autonomous decision-making is the most vital procedure in CAAC. It is responsible for providing UAVs with offensive/defensive strategy selections based on real-time battlefield situations, so that these UAVs can overcome enemies or save themselves [1], [2], [3]. Though great efforts have been devoted in researches on CAAC in recent decades, some flaws still exist in currently adopted autonomous decision-making methods that hinder CAAC from being large-scaled used in military [4]. Autonomous decision-making in air combat can be divided into two parts: maneuver decision-making and target allocation decision-making.

Maneuver decision-making is about UAVs' behaviors to plan specific flight instructions and track specific flight trajectories for certain combat purposes. Various algorithms have been applied. For example, differential game method can ensure an optimal decision result, but needs extremely accurate models that are hard to build [5]. Approximate

The associate editor coordinating the review of this manuscript and approving it for publication was Ming Xu.

TABLE 1. Comparisons of related works with our work.

	Related works	Our work	Main advantages of our work
Maneuver decision-making	Differential game [5]	Optimized dynamic Bayesian network	Lower accuracy requirement for air combat model
	Approximate dynamic programming [6]		Wider application range of air combat scenarios
	Bayesian network [2], [3]		Lower sensitivity to the accuracy of prior probabilities, more consideration of historical decisions and deeper decision-making level
Target allocation decision-making	Hungarian algorithm [7]	Optimized hybrid particle swarm optimization	Less computation and higher efficiency
	Distributed algorithm [8]		Low requirements for airborne hardware equipment
	Genetic algorithm [9]; Particle swarm optimization [10], [11]; Simulated annealing [12],[13]		Faster convergence speed and less possibility of falling into local optimization

dynamic programming method can alleviate the computation pressure, whereas the results tend to have insufficient generalization abilities [6]. Bayesian network (BN) enables machines to take correct decision-making operations by emulating the expert's actions [2], [3], and is appropriate for air combat at various distances from a theoretical point of view [4], whereas the accuracy of decisions highly depends on the the acquisition of empirical knowledge.

Target allocation decision-making refers to the matching between ourselves and enemies, and aims to achieve group optimization under a certain scale. Three kinds of method are in common use presently, including mathematical programming method, negotiation method and swarm intelligence method. Mathematical programming method, such as Hungarian algorithm [7], has the features of simple structure and easy implementation. Its essence is to traverse all feasible spaces, which leads to large-scale calculation and low operation speed. Negotiation method, such as distributed algorithm [8], has strong robustness but requires higher computing and communication abilities of UAVs themselves. Swarm intelligence method, such as genetic algorithm [9], particle swarm optimization algorithm [10], [11], and simulated annealing algorithm [12], has the advantages of high convergence speed and simple operation, but may fall into local optimization and cause the decreased accuracy of the solution attributed to the randomness of initial settings [13].

To solve some aforementioned flaws in those existed methods, we propose an optimized dynamic Bayesian network (DBN) method for maneuver decision-making and an optimized hybrid particle swarm optimization (HPSO) method for target allocation decision-making. Table 1 shows the main advantages of our work compared with related works. Firstly, in the optimized maneuver decision-making method: (1)To enhance the state transition's reliability and the air combat's autonomy, we propose a method of executing combat maneuvers according to sliding mode flight controller (SFC). (2) To improve the dynamic performance and decision-making ability of BN, we propose a method of updating the posterior probabilities as the combat going on. (3) To raise the computing efficiency and decision timeliness, we propose a method of pruning the minimax search tree based on visual prediction. Secondly, in the optimized target allocation decision-making method: (1) To accelerate the convergence speed and ensure the global optimum solution, we propose a method of introducing immigrant particles when the iteration falls into local optimization. (2) To reinforce the cooperation between UAVs and expand the application scope of algorithm, we propose an allocation strategy when the UAV numbers of two sides are unequal. Last but not the least, in the combat end criterion: To narrow the gap between simulations and real air combats, we propose a method of introducing a fire control system to determine the combat end. The simulation results prove that the designed CAAC system has higher fidelity and better decision-making capability. The system could provide a simple and convenient solution to maneuver decision.

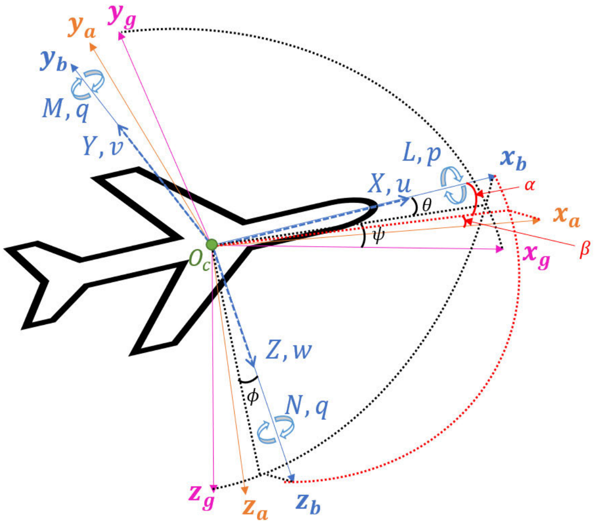


FIGURE 1. Force model of the UAV.

II. METHODS

A. FLIGHT MODELING

Flight modeling depends on force analysis [14]. As shown in Fig. 1, $O_c x_b y_b z_b$ is the Body Axis system, $O_c x_g y_g z_g$ is the Earth Axis system, and $O_c x_a y_a z_a$ is the Wind Axis system. Regarding the aircraft as a rigid body, its force model can be established on the basis of these three coordinate systems. Main variables required for modeling are summarized in Table 2.

According to flight dynamic theories, the motions of the barycenter (O_c in Fig.1) and the rotations around the barycenter under Body Axis system can be written in scalar form

TABLE 2. Variables and their meanings.

Variables	Meanings
X, Y, Z, L, M, N	Forces and moments under Body Axis system.
u, v, w, p, q, r	Velocity and angular velocity under Body Axis system.
x_g, y_g, z_g	Position coordinates under Earth Axis system.
$\theta, \phi, \psi, \alpha, \beta$	Pitch, roll, yaw, attack and sideslip angle.
I_x, I_y, I_z, I_{xz}	Moment of inertia and product of inertia.

as (1) [15], [16]. For the convenience of writing, $\sin\theta$ and $\cos\theta$ are abbreviated as S_θ and C_θ, et al.

$$\begin{cases} \dot{u} = rv - qw - gS_\theta + \dfrac{X}{m} \\ \dot{v} = pw - ru + gC_\theta S_\phi + \dfrac{Y}{m} \\ \dot{w} = qu - pv + gC_\theta C_\phi + \dfrac{Z}{m} \\ \dot{p} = \dfrac{I_z L + I_{xz} N + I_{xz}(I_z + I_x + I_y)pq - [I_z(I_z - I_y) + I_{xy}^2]qr}{I_x I_z - I_{xz}^2} \\ \dot{q} = \dfrac{M + (I_z - I_x)rp - I_{xz}(p^2 - r^2)}{I_y} \\ \dot{r} = \dfrac{I_{xz} L + I_x N + [I_z(I_z - I_y) + I_{xy}^2]pq - I_{xz}(I_x - I_y + I_z)qr}{I_x I_z - I_{xz}^2} \\ \alpha = \arctan w/u \\ \beta = \arcsin v/(u^2 + v^2 + w^2) \end{cases} \tag{1}$$

Then the motions of the aircraft under Earth Axis system can be derived through coordinate axis transformation [15], [16]:

$$\begin{cases} \dot{x_g} = C_\theta S_\psi u + \left(S_\phi S_\theta C_\psi - C_\phi S_\psi\right) v \\ \qquad + \left(C_\phi S_\theta C_\psi + S_\phi S_\psi\right) w \\ \dot{y_g} = C_\theta S_\psi u + \left(S_\phi S_\theta S_\psi + C_\phi C_\psi\right) v \\ \qquad + \left(C_\phi S_\theta S_\psi - S_\phi C_\psi\right) w \\ \dot{z_g} = -S_\theta u + S_\phi C_\theta v + C_\phi C_\theta w \\ \dot{\theta} = qC_\phi - rS_\phi \\ \dot{\phi} = p + qS_\phi \tan\theta + rC_\phi \tan\theta \\ \dot{\psi} = \left(qS_\phi + rC_\phi\right) \sec\theta \end{cases} \tag{2}$$

Autonomous flight controller is often underappreciated in previous reports [17]. To highlight the unmanned concept in CAAC, we manage to apply a second-order sliding mode flight controller (SFC) to the given six-DOF flight model [18]. The logic is to ensure that the system states can converge to the sliding mode manifolds $s = \dot{s} = 0$ in limited time with reference to appropriate control laws [19], [20], [21]. Rewrite (1) into the following form:

$$\dot{x} = f(x) + g(x)u \tag{3}$$

where $x \in R^n, u \in R^n$ are the state vector and the control vector, respectively. Assume the state to be tracked is x_c, then the tracking error can be defined as $e = x_c - x$. Now that the control problem has been transformed into finding the input u that makes e stable:

$$u = g^{-1}(e)\left[\dot{e} - f(e)\right] \tag{4}$$

If the sliding mode surface function and the approach law are selected as:

$$\begin{cases} s = e + \lambda \displaystyle\int edt \\ \dot{s} = -\rho \cdot \operatorname{sign}(s) - ks \end{cases} \tag{5}$$

then control vector u can be designed as:

$$u = \begin{cases} g^{-1}(e)[-f(e) - \lambda(e) - \rho \cdot \operatorname{sign}(s) - ks] & \dot{s} \neq 0 \\ g^{-1}(e)[-f(e) - \lambda(e)] & \dot{s} = 0 \end{cases} \tag{6}$$

The basic structure of the SFC applied in this paper is shown in Fig. 2. Assume that $\dot{p_c}$, $\dot{q_c}$ and $\dot{r_c}$ are the desired angular accelerations of the aircraft. Combing (1) and (6), the control quantities of pitch, roll and yaw channels can be worked out as δ_x, δ_y and δ_z, respectively. Usually, δ_x is equivalent to δ_r (rudder deflection angel), δ_y is equivalent to δ_a (aileron deflection angle), and δ_z is equivalent

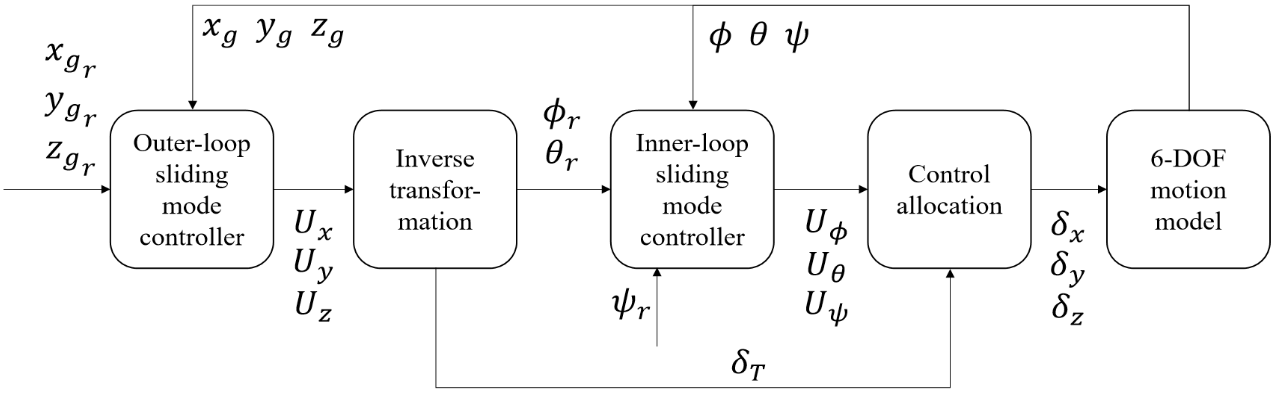


FIGURE 2. A second-order sliding mode flight controller.

to δ_e (elevator deflection angle).

$$\begin{cases} \delta_x = [(I_z^2 - I_z I_y + I_{xz}^2)q\mathrm{r} - I_{xz}(I_z + I_x - I_y)pq - I_z L^p p \\ \qquad + (I_x I_z - I_{xz}^2)(\rho_1 \mathrm{sign}(s_1))/\lambda_1 + \dot{p}_c)]/I_z L^{\delta_x} \\ \delta_y = [I_{xz}(p^2 - r^2) - (I_z - I_x)pr - I_x M^\alpha \alpha \\ \qquad - I_x M^q q + I_y(\rho_2 \mathrm{sign}(s_2)/\lambda_2 + \dot{q}_c)]/M^{\delta_y} \\ \delta_z = [I_{xz}(I_x - I_y + I_z)qr - (I_z^2 - I_z I_y + I_{xz}^2)pq \\ \qquad - N^r r - I_x N^\beta \beta + I_z \rho_3 \mathrm{sign}(s_3)/\lambda_3 + I_z \dot{r}_c]/N^{\delta_z} \end{cases} \tag{7}$$

where λ and ρ with numeric subscript are proportional gain coefficients. L^p, M^β, N^α, et al. are aerodynamic derivatives.

The aerodynamic derivatives' form indicates that the aerodynamic data and the flight model are directly linked with the rudder deflection angles [22]. In this case, the controller can imitate the operation of pilots in manned air combat, which is consistent with the concept of DBN (described later).

B. MANEUVER DECISION-MAKING

Compared with differential games and other methods, the maneuver decision-making model based on Bayesian network is able to be applied to the decision-making of near, medium and long range air combat at the same time, which is a great advantage [23], [24]. However, BN has very strong dependent on the empirical data, which means BN cannot provide valid decisions for UAVs throughout the whole battle process if in lack of authentic empirical data [25]. To address this problem, a more effective DBN system is designed here. The optimized DBN enables UAVs to modify and update the prior probabilities in real-time with regard to their current situations during the air combat.

First and foremost, the blue side is regarded as our side whereas the red side is regarded as the enemy side under normal circumstances. A one-to-one maneuver decision-making model based on DBN is shown in Fig. 3. With the help of influence diagram [26], the DBN model can be presented in the form of an acyclic directed graph composed by nodes and directed arcs. These nodes and arcs demonstrate the logic of maneuver decision-making in a very intuitive way. Taking the blue side as an example, the steps of making maneuver decisions within a time interval $k \sim (k+1)$ are as follow:

Step 1: At time k, get the observation state Z_k^R through observing the red side's flight state X_k^R;

Step 2: Calculate the current combat state for the blue side C_k^B through comparing blue side's flight state X_k^B and the observation status Z_k^R;

Step 3: Assess the current combat state and obtain the posterior probability $P(\Theta_k^B|C_k^B)$ according to C_k^B and the prior probabilities $P(\Theta_k^B)$;

Step 4: Establish an utility value function J_k^B about the posterior probability $P(\Theta_k^B|C_k^B)$ and combat state C_k^B, then solve out the maneuver u_k^B that maximizes the situation evaluation J_k^B through minimax search tree;

Step 5: Apply the automatic flight controller (SFC) to make the blue side UAV transfer from X_k^B to X_{k+1}^B based on u_k^B;

Step 6: Take the posterior probability $P(\Theta_k^B|C_k^B)$ of time k as the prior probabilities $P(\Theta_{k+1}^B)$;

Step 7: Repeat step 1 to step 6 until the air combat ends.

The core of decision-making behaviors based on dynamic Bayesian network is to determine the criterion of decision-making and the algorithm for solving out the maneuver. In other words, the core is to establish a reasonable utility function J_k and adopt an efficient method of searching

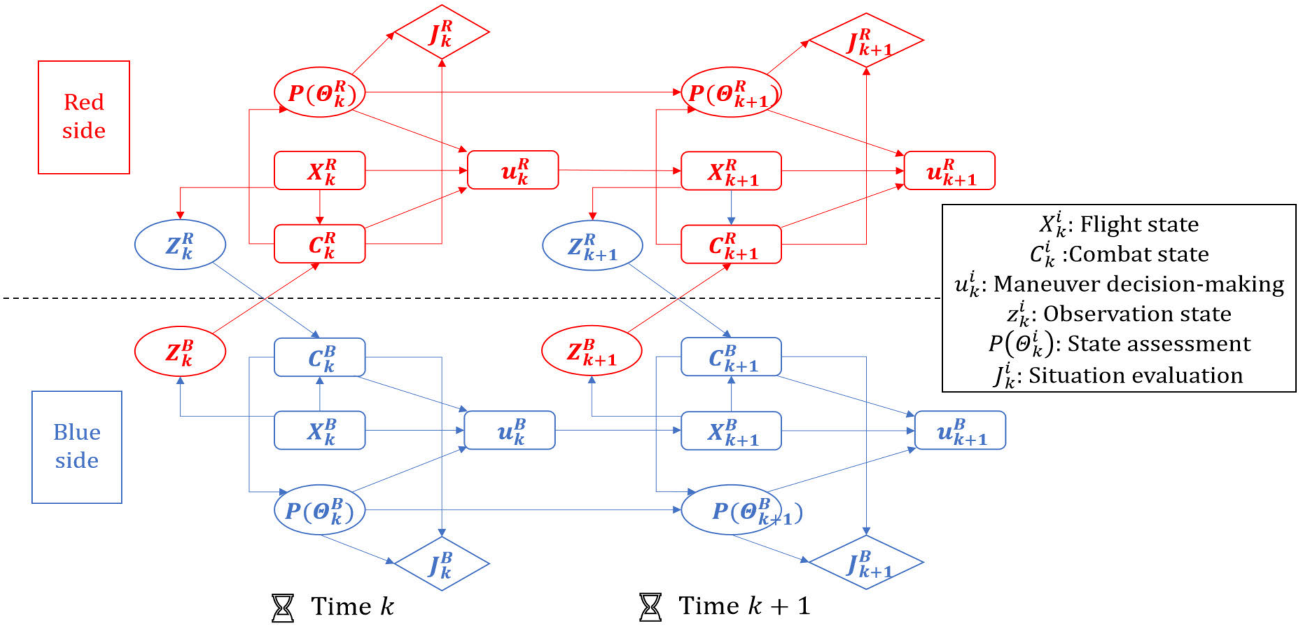


FIGURE 3. The influence diagram of a dynamic Bayesian network.

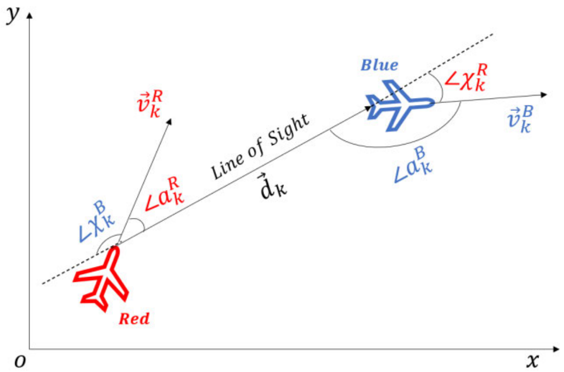


FIGURE 4. Top view of the combat state between two UAVs.

maneuver u_k [27], [28]. Therefore, step 4 is the most critical one of the above seven steps.

1) FLIGHT STATE

At time point k in an air combat, a plane's flight state can be described in the form of a vector:

$$X_k^i = \left[x_k^i, y_k^i, z_k^i, \psi_k^i, \phi_k^i, \theta_k^i, V_k^i\right] \quad i \in [B, R] \tag{8}$$

where state variables x_k^i,y_k^i, z_k^i are the positions of the aircraft in Earth Axis system. ψ_k^i, θ_k^i, ϕ_k^i are the Euler angles: yaw angle, pitch angle and roll angle respectively. V_k^i is the ground velocity.

2) COMBAT STATE

Combat state is the relative position of two UAVs in three-dimensional space at a certain moment. It includes the position and the direction information (Fig. 4).

At time point k in the air combat, the combat state is demonstrated as $C_k^i = f^i\left(X_k^B, X_k^R\right), i \in [B, R]$. For blue side, C_k^B can be defined as $C_k^B = \left(a_k^B, \chi_k^B, d_k, \eta_k^B\right), a_1^B, \chi_2^B \in [0,\pi]$. a_k and χ_k are the leading angle and lagging angle of blue side relative to red side respectively; d_k is the physical distance between the two UAVs; $\eta_k = E_B/E_R$ is the ratio of the blue side UAV's energy to the red side UAV's.

a_k, χ_k and d_k can be obtained according to geometric knowledge, as shown in (9).

$$\begin{cases} \vec{d}_k = \left(x_k^R - x_k^B, y_k^R - y_k^B, z_k^R - z_k^B\right) = (\Delta x, \Delta y, \Delta z) \\ \vec{V}_k^B = V_k^B\left(cos\phi_k^B sin\psi_k^B, cos\phi_k^B cos\psi_k^B, sin\phi_k^B\right) \\ \angle a_k^B = arccos\left[\vec{V}_k^B \cdot \vec{d}_k / \left(\left|\vec{V}_k^B\right| \times \left|\vec{d}_k\right|\right)\right] \\ \angle\chi_k^R = \pi - \angle a_k^B \end{cases} \tag{9}$$

η_k then can be calculated by definition, as shown in (10).

$$\begin{cases} \eta_k = E_B/E_R \\ E_i = H + V_i^2/g \quad i \in [B, R] \end{cases} \tag{10}$$

where H is the altitude and g is the gravitational acceleration.

3) MANEUVER DECISION-MAKING

Maneuvers in actual air combat determines how planes act during the battle. Taking Pougatcheff Cobra Maneuver as an example, it allows a plane to quickly change speed, so that the battlefield situation shall be reversed in an instant [29].

Previous studies commonly use typical maneuver library as the object of maneuver decision-making [4], [5], [6]. By comparison, we hope that the obtained result could cover as many positions as possible. From the perspective of overload, the maneuver flight of a UAV can be described with the tangential overload n_x and the normal overloads n_y, n_z. Tangential overload $n_x = X/mg$ is parallel to the direction of flight speed, which means the maneuver of changing the flight speed. Normal overload $n_y = Y/mg$ is perpendicular to the direction of flight speed, which means the maneuver of ascending or descending in the vertical plane. And normal overload $n_z = Z/mg$ is also perpendicular to the direction of flight speed, which means the maneuver of left or right turning in the horizontal plane. Therefore, rather than using the typical maneuver library, we choose to regard the current position of aircraft X_k^i as the reference point and then retrieve the position of next moment with seven kinds of overload commands within the decision-making time interval $k \sim (k+1)$: 1. n_{xc}, 2. n_{yc}, 3. n_{zc}, 4. n_{xc} and n_{yc}, 5. n_{xc} and n_{zc}, 6. n_{yc} and n_{zc}, 7. n_{xc}, n_{yc} and n_{zc}. Compared with the typical maneuver library, these seven commands have an advantage that they can provide a more complete description of the UAV's motions in three-dimensional space. The motion types considered in the typical maneuver library method are limited. This limitation usually results in a problem that the UAV may not be able to reach the most favorable position if adopting the typical maneuver library method in making maneuver decision-making. However, if adopting the above-proposed method which is based on the seven overload commands, the UAV is able to carry out more complex motions, and reach more potential positions. The most favorable position may exist in these extra potential positions. This design of optional maneuvers based on overloads is named as Overload-limited Maneuver. The Overload-limited Maneuver, to a certain extent, solves the above-mentioned problem caused by limited considerations of the motion types.

After working out what maneuver to do at the next time interval, the estimate position of the UAV at the next moment can be obtained. The estimate position is remarked as $\tilde{X}_{k+1}^i$. The following step is to consider the effect of the automatic flight controller. Only if the UAV do can fly to the position $\tilde{X}_{k+1}^i$ stably within the time interval, can the decided maneuver be effective. As mentioned in Chapter A, the SFC adopted in this paper is designed to track the desired accelerations and the desired angular accelerations, rather than the desired overloads. Therefore, it is needed to establish a relationship between the accelerations and the overloads.

According to the definitions of n_x, n_y, n_z, forces X_c, Y_c and Z_c can be obtained.

$$\begin{cases} X_c = gn_{xc} \\ Y_c = gn_{yc} \\ Z_c = gn_{zc} \end{cases} \tag{11}$$

Combining (1) and (11), a set of desired accelerations ($\dot{u_c}$, $\dot{v_c}$, $\dot{w_c}$) can be obtained. ($\dot{u_c}$, $\dot{v_c}$, $\dot{w_c}$) brings the change of velocities, resulting in a set of desired velocities u_c, v_c, w_c. u_c, v_c, w_c cause the change of moments, resulting in a set of desired moments L_c, M_c, N_c.

$$\begin{cases} L_c = \rho(u_c^2 + v_c^2 + w_c^2)bSC_L \\ M_c = \rho(u_c^2 + v_c^2 + w_c^2)\bar{c}SC_M \\ N_c = \rho(u_c^2 + v_c^2 + w_c^2)bSC_N \end{cases} \tag{12}$$

Combining (1) and (12), a set of desired angular accelerations $\dot{p_c}$, $\dot{q_c}$, $\dot{r_c}$ can be obtained. Then combining (7) with (12), a control quantity set u_k^i can be solved out, as shown in (13). The control quantity set (u_k^i) enables the UAV to move from the current position X_k^i to the future position $\tilde{X}_{k+1}^i$.

$$u_k^i = f_{SFC}\left(X_k^i, \tilde{X}_{k+1}^i\right) = \{\delta_r, \delta_a, \delta_e\} \tag{13}$$

Then combining u_k^i, (1) and (2) together, the actual position that the UAV can reach at time $k+1$ in fact can be derived and recorded as X_{k+1}^i. In this way, the UAV completes a state transition based on the maneuver decision considering the effect of the automatic flight controller.

4) OBSERVATION STATE

The enemy side's flight states cannot be obtained directly in fact, which is the main source of uncertainty in air combat [26]. To bring simulations closer to real combat, it's designed that one can only obtain the other one's state with noise.

At time point k in air combat, the observation state of red side for blue side is set as:

$$z_k^B = C_k^i\left(s_k^B, s_k^R\right) + \zeta_k \tag{14}$$

where $C_k^i\left(s_k^B, s_k^R\right)$ is the combat state mentioned above, ζ_k is the observation noise and is simulated by Gaussian white noise in this work.

5) STATE ASSESSMENT

State assessment is carried on by the aircraft in combat at each time, which refers to the probability distributions of one UAV's combat situation after a certain maneuver. Situations are divided into four categories: Superiority ($\Theta_k^i = 1$), Neutrality ($\Theta_k^i = 2$), Inferiority ($\Theta_k^i = 3$), and Mutual Disadvantage ($\Theta_k^i = 4$). For blue side, Superiority shows possibility to defeat red side; Neutrality represents they tied up; Inferiority indicates blue side is in great danger; Mutual Disadvantage means they both have opportunity to win.

Prior probabilities can be defined as $P\left(\Theta_k^i = j\right), j = 1, 2, 3, 4$. It is easy to get that $\sum_{j=1}^4 P\left(\Theta_k^i = j\right) = 1$. The posterior probability of the state assessment is given as:

$$P\left(\Theta_k^i = j \mid C_k^i\right) = \frac{P\left(\Theta_k^i = j\right) P\left(C_k^i \mid j\right)}{\sum_{j=1}^4 P\left(\Theta_k^i = j\right) P\left(C_k^i \mid j\right)} \triangleq P\left(\Theta_{k+1}^i = j\right) \tag{15}$$

Equation (15) means that the posterior probability is based on probability distributions, and can be more realistic by taking the combat state into consideration. $P\left(C_k^i \mid j\right)$ is the

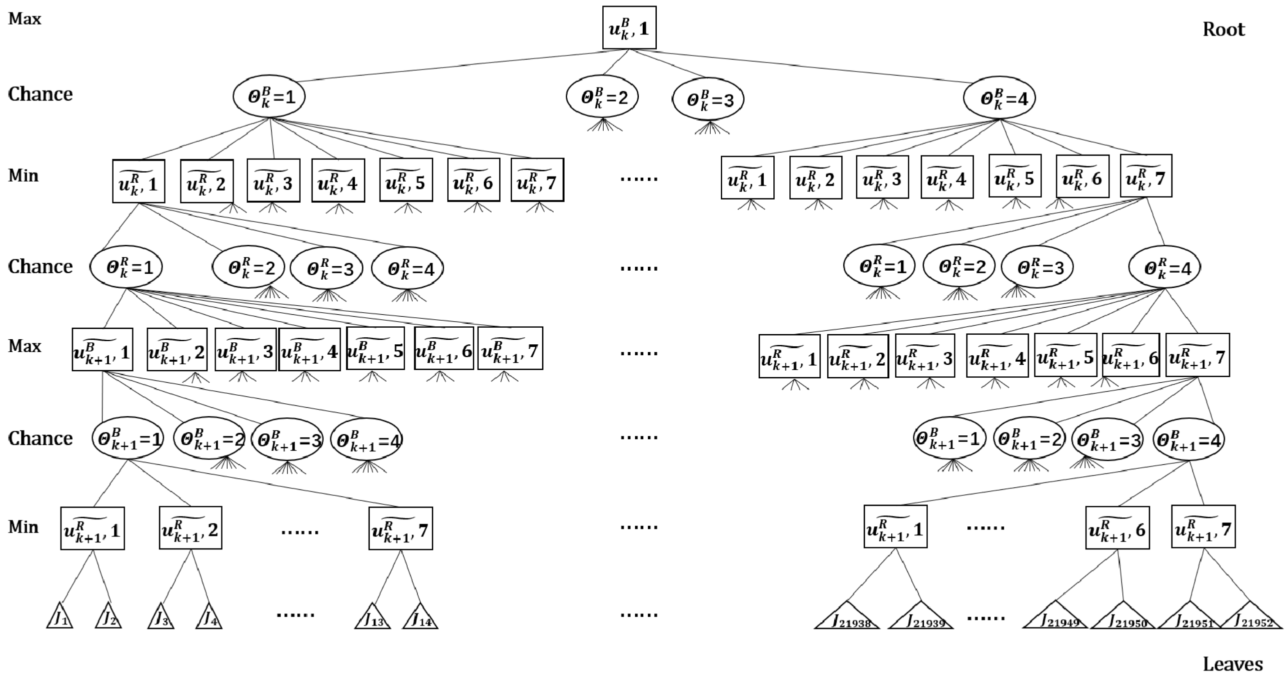

Max
Chance
Min
Chance
Max
Chance
Min
Root
Leaves
$u_k^B, 1$
$\Theta_k^B=1$
$\Theta_k^B=2$
$\Theta_k^B=3$
$\Theta_k^B=4$
$\widetilde{u_k^R}, 1$ $\widetilde{u_k^R}, 2$ $\widetilde{u_k^R}, 3$ $\widetilde{u_k^R}, 4$ $\widetilde{u_k^R}, 5$ $\widetilde{u_k^R}, 6$ $\widetilde{u_k^R}, 7$
$\Theta_k^R=1$ $\Theta_k^R=2$ $\Theta_k^R=3$ $\Theta_k^R=4$
$\widetilde{u_{k+1}^B}, 1$ $\widetilde{u_{k+1}^B}, 2$ $\widetilde{u_{k+1}^B}, 3$ $\widetilde{u_{k+1}^B}, 4$ $\widetilde{u_{k+1}^B}, 5$ $\widetilde{u_{k+1}^B}, 6$ $\widetilde{u_{k+1}^B}, 7$
$\widetilde{u_{k+1}^R}, 1$ $\widetilde{u_{k+1}^R}, 2$ $\widetilde{u_{k+1}^R}, 3$ $\widetilde{u_{k+1}^R}, 4$ $\widetilde{u_{k+1}^R}, 5$ $\widetilde{u_{k+1}^R}, 6$ $\widetilde{u_{k+1}^R}, 7$
$\Theta_{k+1}^B=1$ $\Theta_{k+1}^B=2$ $\Theta_{k+1}^B=3$ $\Theta_{k+1}^B=4$
$\widetilde{u_{k+1}^R}, 1$ $\widetilde{u_{k+1}^R}, 2$ $\widetilde{u_{k+1}^R}, 6$ $\widetilde{u_{k+1}^R}, 7$
......
J_1 J_2 J_3 J_4 J_{13} J_{14}
J_{21938} J_{21939} J_{21949} J_{21950} J_{21951} J_{21952}

FIGURE 5. Minimax search tree for maneuver decision-making.

conditional likelihood probability, which is often artificially defined according to C_k^i:

$$\begin{aligned} P\left(C_k^i|j\right) &= P\left(a_k^i, \chi_k^i, d_k, \eta_k^i\,|j\right) \\ &= p\left(a_k^i\,|j\right)\cdot p\left(\chi_2^i\,|j\right)\cdot p\left(d_k\,|j\right)\cdot p\left(\eta_k^i\,|j\right) \end{aligned} \tag{16}$$

where $p\left(a_k^i\,|j\right)$, $p\left(\chi_2^i\,|j\right)$, $p\left(d_k\,|j\right)$ and $p\left(\eta_k^i\,|j\right)$ denote conditional likelihood probabilities in combat situation Θ_k^i.

6) SITUATION EVALUATION

An utility function J_k^i is applied to reflect the preferences of the aircraft in combat, as shown in (17).

$$\begin{aligned} & J_k^i\left(U_k^r,\cdots,U_{k+N}^r,U_k^b,\cdots,U_{k+N}^b\right) \\ &\quad = \sum_k^{k+N}\sum_{j=1}^{4} P\left(\Theta_k^i=j\,|\,C_k^i\right)U^i\left(j,C_k^i\right) \end{aligned} \tag{17}$$

where N is the forward step number. $U^i\left(j,C_k^i\right)$ is the utility according to C_k^i:

$$\begin{aligned} U^i\left(j,C_k^i\right) &= \lambda_j^a U_j^i\left(a_k^i\right)+\lambda_j^\chi U_j^i\left(\chi_k^i\right)+\lambda_j^d U_j^i\left(d_k\right) \\ &\quad +\lambda_j^\eta U_j^i\left(\eta_k^i\right) \end{aligned} \tag{18}$$

where $\lambda_j^{()}$ are weighting factors which satisfy $\lambda_j^a+\lambda_j^\chi+\lambda_j^d+\lambda_j^\eta=1$.

The optimal maneuver action and the corresponding largest utility value for each time point shall be determined by the minimax method (Fig. 5). The minimax method assumes that the opponent is smart, which means that the maneuver is obtained under the circumstance that the opponent has taken a sufficiently advantageous maneuver [30].

Obviously, the computation of searching will increase exponentially as the expected level rises [31]. To solve the problem, the idea is that one could foresee the future movement of target aircraft to some degree via a visual check, which is inspired by human pilots' experience [32]. The basic principle of this method is that the position of enemy aircraft relative to our aircraft at the next moment $[x_{tob},y_{tob},z_{tob}]_{k+1}^T$ is able to be predicted by the information at the current moment. The predicted position is record as $[x_{tob},y_{tob},z_{tob}]_{k+1}^{*T}$.

$$\begin{bmatrix} x_{tob} \\ y_{tob} \\ z_{tob} \end{bmatrix}_{k+1}^{*} = \begin{bmatrix} x_{tob} \\ y_{tob} \\ z_{tob} \end{bmatrix}_{k} + v_t\Delta t\begin{bmatrix} (i_{at}\cdot i_{bo}) \\ (i_{at}\cdot j_{bo}) \\ (i_{at}\cdot k_{bo}) \end{bmatrix} - v_o\Delta t\begin{bmatrix} cos\alpha_o cos\beta_o \\ sin\beta_o \\ sin\alpha_o cos\beta_o \end{bmatrix} \tag{19}$$

where $[x_{tob},y_{tob},z_{tob}]_k^T$ is the target aircraft's position relative to our aircraft's body axis at time point k; $[i_{at},j_{at},k_{at}]^T$ is the aerodynamic coordinates of the target aircraft; $[i_{ao},j_{ao},k_{ao}]^T$ is the aerodynamic coordinates of our aircraft; v_o and v_t are the velocity of our own aircraft and the target one respectively; α_o and β_o are the angle of attack and the angle of sideslip for our aircraft respectively.

Based on this idea, we can eliminate some maneuvers that the target enemy cannot take, thereby realiz-

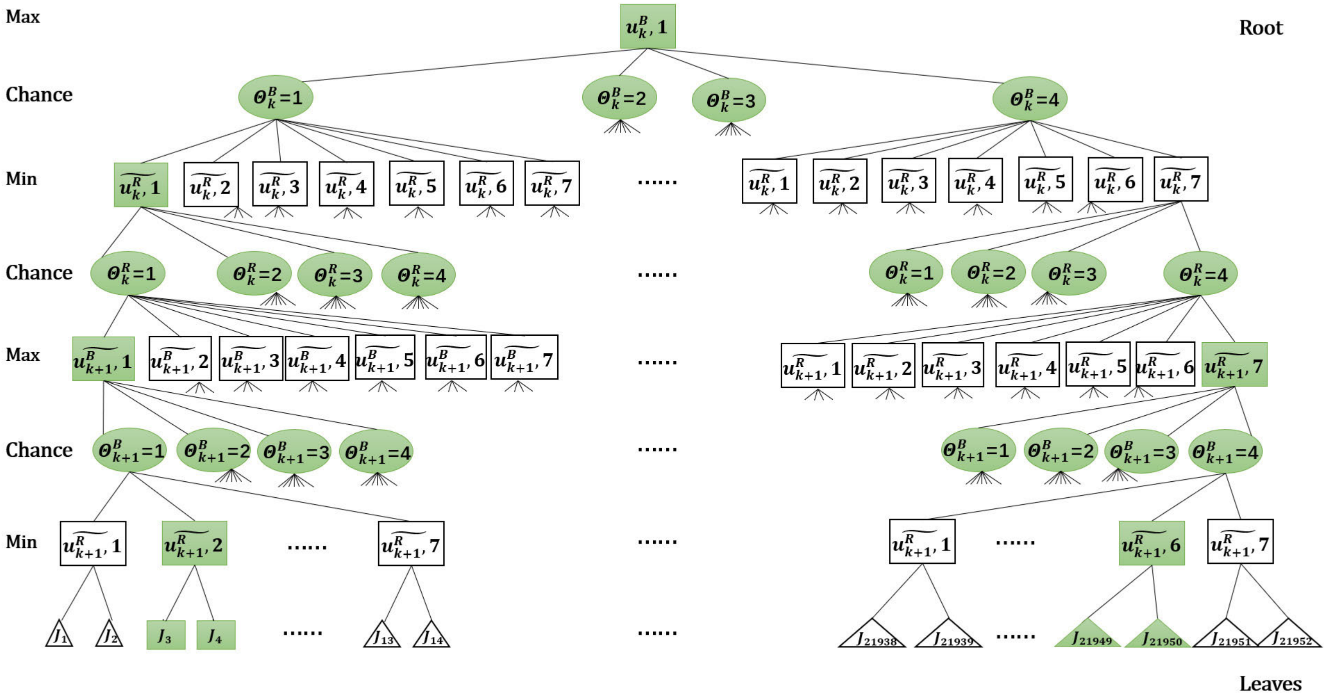


FIGURE 6. Minimax search tree after pruning.

ing pruning the search tree and reducing the amount of calculation (Fig. 6).

C. TARGET ALLOCATION DECISION-MAKING

The target allocation decision-making of many-to-many CAAC refers to assigning all enemy UAVs to our UAVs, so as to reach the optimal overall attack utility. The matching function (objective optimization function) is given as below:

$$\begin{cases} \max\left(\sum_{j=1}^{n} U_{ij} td_i^j\right) & i = 1, 2 \cdots m;\ td_i^j \in [0, 1] \\ \sum_{i=1}^{m} td_i^j \leq l & j = 1, 2 \cdots, n;\ 1 \leq l \leq m \\ \sum_{i=1}^{m} \sum_{j=1}^{n} td_i^j \geq n \end{cases} \tag{20}$$

where td_i^j represents whether enemy UAV j is allocated to our UAV i or not, $td_i^j = 1$ means yes and $td_i^j = 0$ means no; U_{ij} is the utility function value when our UAV i fight against enemy UAV j; l is the number of UAVs fighting against enemy UAV j at the same time after allocation; m is the total number of our UAVs; n is the total number of enemy UAVs.

Equation (20) describes a typical non-deterministic polynomial complete (NP-C) problem [33]. To address it, a HPSO algorithm is adopted here. And an optimized HPSO is proposed to accelerate the change of particles in the later iteration stages by introducing a judgement parameter.

1) STANDARD HPSO

The HPSO algorithm is developed on the basis of the PSO algorithm. A PSO algorithm is an intelligent algorithm which simulates the foraging behavior of birds [34]. It can achieve the extremum optimization by following the individual extremum and group extremum [35]. The method of updating particles is given as below:

$$\begin{cases} \mathcal{V}_{k+1} = w\mathcal{V}_k + l_1 r_1 \left(P_{best}^k - \mathcal{X}_k\right) + l_2 r_2 \left(G_{best}^k - \mathcal{X}_k\right) \\ \mathcal{X}_k = \mathcal{X}_k + \mathcal{V}_{k+1} \end{cases} \tag{21}$$

where w is the inertia weight coefficient; k is the current number of iterations; $\mathcal{X}$ is the position of particles; $\mathcal{V}$ is the velocity of particles; l_1 and l_2 are the learning factors; r_1 and r_2 are random numbers distributed in interval [0, 1]; P_{best} and G_{best} are the individual extremum and group extremum respectively.

Based on (21), the standard HPSO algorithm draws lessons from genetic algorithm about crossover operation and mutation operation when updating particles [36].

2) OPTIMIZED HPSO

Compared with PSO algorithm, the standard HPSO algorithm does not update the particle position by tracking the extreme value, so as to prevent the influence of contrived inertia weight, particle speed and learning rate [37]. However, the convergence speed of the standard HPSO algorithm is fast at the early stage of iteration and is slow at the later stage of iteration. With the increase of iteration numbers, the particles become more and more similar as the population converges, which may lead to local optimal solutions [38].

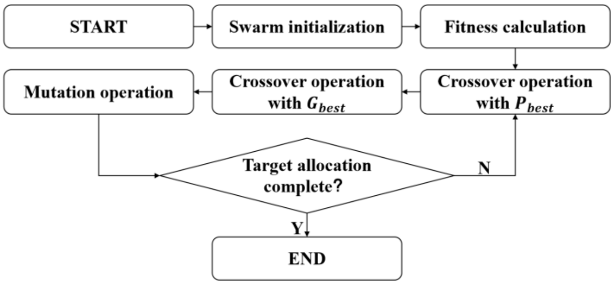

START
Swarm initialization
Fitness calculation
Mutation operation
Crossover operation with G_{best}
Crossover operation with P_{best}
Target allocation complete?
N
Y
END

FIGURE 7. Flow chart of the optimized HPSO algorithm.

The way to avoid the condition is introducing a judgement parameter named ϖ. When iterating, if the difference between the group extrema of several adjacent iterations is less than ϖ, it is considered that the algorithm falls into local optimization. The immigrant particle should be introduced by this moment. Their fitness ought to be calculated and then compared with existed particles. The particle with higher fitness will be selected and retained. It is worth mentioning that immigrant particle is only introduced when local optimization happens, which means the global search ability is improved whereas the calculation quantities do not increase a lot.

Steps of the optimized HPSO algorithm are as follows (see Fig. 7):

Step 1: Initialize, set the maximum number of iterations Num, set the number of particle populations K, generate K numbers of distribution schemes as the initial particle swarm, calculate the fitness of each particle, record the initial individual extremum P_{best} and the global extremum G_{best};

Step 2: Cross over each particle in the swarm with P_{best}, calculate the fitness of the two particles generated by crossover operation, select and retain the one with better fitness as a new particle, update P_{best} if the fitness of the new particle is better than the original P_{best};

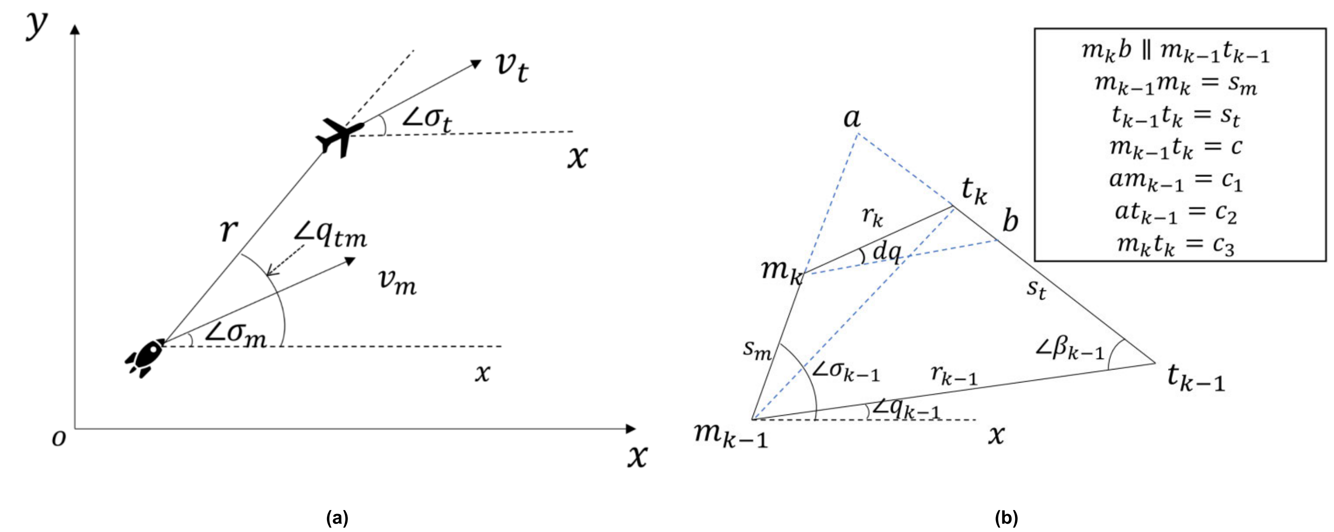
Step 3: Cross over the new particle obtained in step 2 with G_{best}, calculate their fitness, select and retain the one with better fitness as a new particle, update P_{best} if the fitness of the new particle is better than the original P_{best};

Step 4: Mutate the new particle obtained in step 3, calculate its fitness; update P_{best} if the fitness of the new particle is better than the original P_{best};

Step 5: Determine G_{best} according to the updated P_{best}, end the iteration if the maximum number of iterations is reached, go to step 6 if not;

Step 6: Calculate the difference between G_{best}s of several iterations, go to step 2 directly if the difference is greater than ϖ, go step 2 after introducing immigrant particles and updating the swarm if not;

However, numbers of UAVs on both sides are usually different in real air combats [39]. It is prone to have difficulties



(a)

(b)

FIGURE 8. (a) Principle and (b) graphical solution of the proportional guidance method.

in initializing the particle swarm due to constraints. Solutions are as follows:

First of all, suppose that our side have m number of UAVs and the enemy side has n number of UAVs. Then, a matrix $\mathfrak{U}$ about utility is established to describe the relationship between UAVs.

$$\mathfrak{U} = \begin{bmatrix} U_{11} & U_{12} & \cdots & U_{1n} \\ U_{21} & U_{22} & \cdots & U_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ U_{m1} & U_{m2} & \cdots & U_{mn} \end{bmatrix} \tag{22}$$

When $m > n$, supplement $m - n$ columns of constants $\mathcal{C}_1$ on the right side of the matrix, as shown in (23).

$$\mathfrak{U} = \begin{bmatrix} U_{11} & U_{12} & \cdots & U_{1n} & \overbrace{\mathcal{C}_1 \quad \cdots \quad \mathcal{C}_1}^{m-n} \\ U_{21} & U_{22} & \cdots & U_{2n} & \mathcal{C}_1 \quad \cdots \quad \mathcal{C}_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \quad \cdots \quad \vdots \\ U_{m1} & U_{m2} & \cdots & U_{mn} & \mathcal{C}_1 \quad \cdots \quad \mathcal{C}_1 \end{bmatrix} \tag{23}$$

$\mathcal{C}_1$ is a constant less than all other elements in the matrix. In this case, there will be another $m - n$ number of our UAVs after target allocation. Thus, the next step is to allocate the redundant UAVs regarding real-time mission requirements. For example, if adopting defense principle, they will be assigned to the enemy UAV which poses the greatest threat to our targets; if adopting marking principle, they will be assigned to the enemy UAV which has the strongest flight performance; if adopting annihilation principle, they will be assigned to the enemy UAV which has the poorest combat advantage.

When $m < n$, supplement $n - m$ rows of constants $\mathcal{C}_2$ on the bottom of the matrix, as shown in (24).

$$\mathfrak{U} = \begin{bmatrix} U_{11} & U_{21} & \cdots & U_{m1} & \overbrace{\mathcal{C}_2 \quad \cdots \quad \mathcal{C}_2}^{n-m} \\ U_{12} & U_{22} & \cdots & U_{m2} & \mathcal{C}_2 \quad \cdots \quad \mathcal{C}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \quad \cdots \quad \vdots \\ U_{1n} & U_{2n} & \cdots & U_{mn} & \mathcal{C}_2 \quad \cdots \quad \mathcal{C}_2 \end{bmatrix}^{\mathrm{T}} \tag{24}$$

$\mathcal{C}_2$ is a constant greater than all other elements in the matrix. In this case, there is no need to do additional target allocation operations.

Aforementioned operations ensure that the generated initial particle swarm can meet the needs of target allocation as long as the swarm is a positive integer random arrangement from 1 to the utility matrix's order.

D. END CRITERIA

In previous studies on air combat, the end criteria are generally simple. There are mainly two kinds: one is setting a cone-shaped attack zone to represent the attack performance of a fighter [1], [2], [3], another is constructing a value function that describes the payoff of an attack [23], [29].

To fight closer to the real world, a fire control system is especially involved here. It is designed according to the principle of proportional guidance method. As shown in Fig. 8(a), the basic law is:

$$d\sigma_m/dt = K \cdot dq_{tm}/dt \tag{25}$$

where K is the proportional constant; σ_m is the angle between the missile velocity vector and reference line; q_{tm} is the angle between the sight line and reference line [40], [41].

The graphical solution of proportional guidance method is given in Fig. 8(b). According to the trigonometric relationship, we can obtain:

$$\begin{cases} c = [(x_k^t - x_{k-1}^m)^2 + (y_k^t - y_{k-1}^m)^2 + (z_k^t - z_{k-1}^m)^2]^{1/2} \\ \beta_{k-1} = \arccos[(r_{k-1}^2 + s_t^2 - c^2)/2r_{k-1}s_t] \\ \angle m_k a t_k = \pi - (\sigma_{k-1} - q_{k-1} + \beta_{k-1}) \\ c_1 = r_{k-1} \sin\beta_{k-1} / \sin \angle m_k a t_k \\ c_2 = r_{k-1} \sin(\sigma_{k-1} - q_{k-1}) / \sin \angle m_k a t_k \\ am_k = c_1 - s_m, \quad at_k = c_2 - s_t \\ c_3 = (am_k^2 + at_k^2 - 2am_k \cdot at_k \cdot \cos \angle m_k a t_k)^{12} \\ \widetilde{dq} = \angle t_k m_k b \approx \angle t_k m_{k-1} t_{k-1} = \arccos \dfrac{r_{k-1}^2 + c^2 - s_t^2}{2r_{k-1}c} \end{cases} \tag{26}$$

Compared with the real dq, the $\widetilde{dq}$ obtained at this moment may have a large error, especially when the missile is close to the target [42], [43]. Equation (27) provides a method to correct $\widetilde{dq}$ as dq^*. dq^* is closer to dq much more than $\widetilde{dq}$.

$$\begin{cases} \widetilde{q_k} = q_{k-1} + \widetilde{dq} \\ \widetilde{\sigma_k} = \sigma_{k-1} + K \cdot \widetilde{dq} \\ \widetilde{\angle m_k a t_k} = \arccos \dfrac{(c_1 - s_m)^2 + c_3^2 - (c_2 - s_t)^2}{2(c_1 - s_m)c_3} \\ dq^* = \widetilde{dq} - \widetilde{\sigma_k} - \widetilde{q_k} - \dfrac{\angle m_k a t_k}{K+1} \\ q_k = q_{k-1} + dq^* \\ \sigma_k = \sigma_{k-1} + K \cdot dq^* \end{cases} \tag{27}$$

The number of iterations depends on the requirement. Usually, two iterations are enough.

$$\begin{cases} x_k^a = x_{k-1}^t + c_2/s_t \cdot \left(x_k^t - x_{k-1}^t\right) \\ y_k^a = y_{k-1}^t + c_2/s_t \cdot \left(y_k^t - y_{k-1}^t\right) \\ z_k^a = z_{k-1}^t + c_2/s_t \cdot \left(z_k^t - z_{k-1}^t\right) \\ x_k^m = x_{k-1}^m + s_m/c_1 \cdot \left(x_k^a - x_{k-1}^m\right) \\ y_k^m = y_{k-1}^m + s_m/c_1 \cdot \left(y_k^a - y_{k-1}^m\right) \\ z_k^m = z_{k-1}^m + s_m/c_1 \cdot \left(z_k^a - z_{k-1}^m\right) \end{cases} \tag{28}$$

Then, a condition ($\mathfrak{FS}$) is defined to represent the final stage of an autonomous air combat.

$$\mathfrak{FS} \triangleq \begin{cases} \left|X_k^B(1:3) - X_k^B(1:3)\right| < 100 \text{ m} \\ P\left(\Theta_k^i = 1 \middle| C_k^i\right) > \sum_{j=2}^{4} P\left(\Theta_k^i = j \middle| C_k^i\right), \\ \qquad i = B, R \end{cases} \tag{29}$$

When one side enters the final stage, it will begin to conduct an additional decision while doing Maneuver Decision-making [44], [45], [46]: shoot a virtual missile that follows the above-mentioned fire control law. When the calculation shows the missile can hit the enemy's aircraft (the distance between missile and target d_{mt} is less than 10 m) within next five simulation time intervals, then it is determined that one side wins and the air battle is in the end.

TABLE 3. Calculation formulas of conditional likelihood probabilities.

Combat state	Θ_k^i	Conditional likelihood probability
Bearing angle	1,3	$p(a_k^i \mid j) = (\alpha a_k^i/\pi + 1 - \alpha/2)/\pi$
	2,4	$p(a_k^i \mid j) = (-\alpha a_k^i/\pi + 1 + \alpha/2)/\pi$
Angle-off	3,4	$p(\chi_k^i \mid j) = (\alpha\chi_k^i/\pi + 1 - \alpha/2)/\pi$
	1,2	$p(\chi_k^i \mid j) = (-\alpha\chi_k^i/\pi + 1 + \alpha/2)/\pi$
Distance	1	$p(d \mid j) = 1/D$
	2,3,4	$p(d \mid j) = (-\alpha d/D + 1 + \alpha/2)/D$
Energy	1,2,4	$p(\eta_k^i \mid j) = (\eta_k^i/2 - \beta - 1/5)/5$
	3	$p(\eta_k^i \mid j) = \eta_k^i/\beta$

Where $D = 20000$, $\alpha = 0.075$, β =2.

TABLE 4. Utility functions of four combat situations.

Utility	Superiority	Neutrality	Inferiority	Mutual Disadvantage
$U(a_k^i)$	$(\pi - a_k^i)/\pi$	$(\pi - a_k^i)/\pi$	a_k^i/π	$(\pi - a_k^i)/\pi$
$U(\chi_K^i)$	$(\pi - \chi_k^i)/\pi$	$(\pi - \chi_k^i)/\pi$	$(\pi - \chi_k^i)/\pi$	$(\pi - \chi_k^i)/\pi$
$U(d_k)$	$(D - d_k)/D$	$(D - d_k)/D$	d_k/D	d_k/D
$U(\eta_K^i)$	$\eta - \beta$	$\eta - \beta$	2	$\eta - \beta$

Where $D = 2000$, $\beta = 2$.

TABLE 5. Weighting factors of four combat situations.

Weight	Superiority	Neutrality	Inferiority	Mutual Disadvantage
$\lambda_j(a_k^i)$	0.25	0.30	0.00	0.05
$\lambda_j(\chi_k^i)$	0.05	0.10	0.15	0.10
$\lambda_j(d_k)$	0.60	0.55	0.55	0.70
$\lambda_j(\eta_k^i)$	0.10	0.05	0.30	0.15

TABLE 6. Initial states of two aircraft in Mutual Disadvantage case.

Initial state	x_0^i /m	y_0^i /m	z_0^i /m	ψ_0^i /°	θ_0^i /°	ϕ_0^i /°	V/m $\cdot s^{-1}$
Blue	0	0	800	90	10	0	250
Red	0	1700	800	-70	10	0	250

III. SIMULATION RESULTS

Conditional likelihood probabilities, utility functions and weighting factors in this paper are designed according to different situations (see Table 3–5).

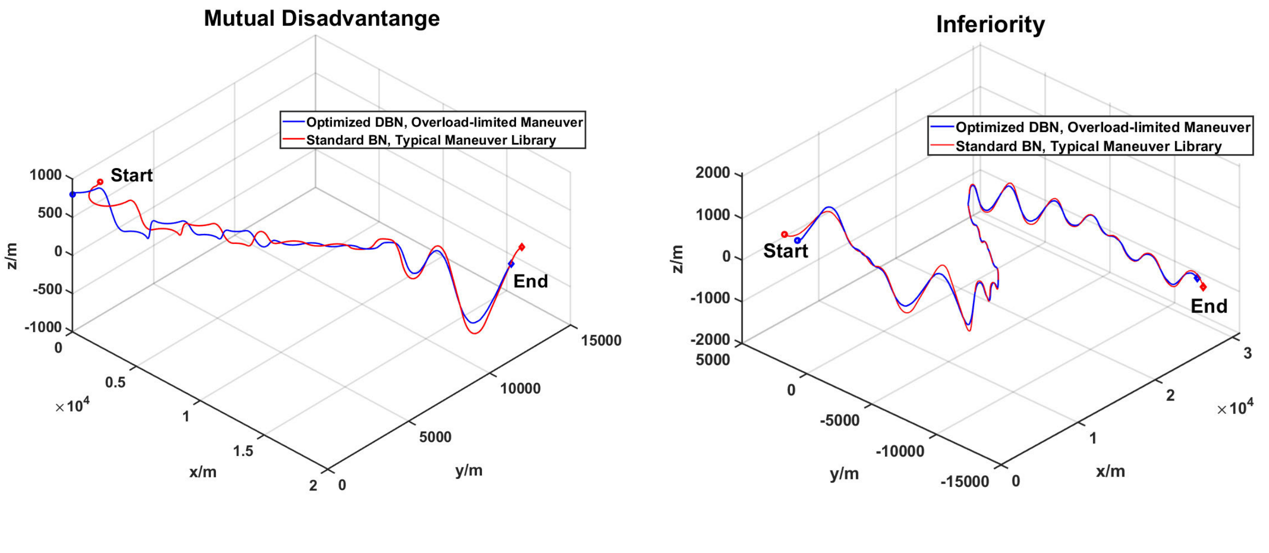
To reiterate, blue side is regarded as our side while red side is regarded as enemy side in all the following simulation cases.

A. 1-TO-1 MANEUVER DECISION-MAKING SIMULATIONS

In all simulations below, our plane fights on the basis of the optimized DBN and the Overload-limited Maneuver, while the enemy plane relies on the standard BN and the typical maneuver library. In order to verify our optimization, we selected two typical initial situations (Mutual Disadvantage and Inferiority) for simulation.

1) MUTUAL DISADVANTAGE CASE

Mutual Disadvantage means they both have opportunity to win. This initial situation (see Table 6) shows that both UAVs are in great dangerous.



(a) (b)

FIGURE 9. Flight trajectories of (a) Mutual Disadvantage case and (b) Inferiority case.

TABLE 7. Initial states of two aircraft in Inferiority case.

Initial state	x_0^i /m	y_0^i /m	z_0^i /m	ψ_0^i /°	θ_0^i /°	ϕ_0^i /°	$V/\mathrm{m}\cdot\mathrm{s}^{-1}$
Blue	500	1000	1000	170	5	0	235
Red	500	2000	1000	180	-5	0	250

The simulation result is presented in Fig. 9(a). Both sides tried to escape the risk through scissor maneuvers. After a series of scissor maneuvers and S-breaking maneuvers, the situation changed. The blue UAV entered a slight advantage situation through smaller and faster turns and the red UAV began to defend. The red UAV then expected to get rid of the blue UAV through climbing and decelerating, or diving and accelerating. However, the red UAV was still closely followed by the blue UAV. Finally, after several rounds of decision-making, with the gradual update of the blue UAV's prior probability, the blue UAV successfully predicted the red UAV's actions, and achieved an absolute advantage by taking smaller but faster maneuvers. Then the fire control system judged that the red UAV shall be shot down by blue side in no time. The combat ended. The whole air combat process cost about 100 steps, which equals to about 25 s in real world.

Considering that there is uncertainty in the observation of each other's position and state in air combat (see the section of observation state), this Mutual Disadvantage case was simulated for 20 times. Each time the blue side could win the combat, which shows that the maneuver decision-making algorithm adopted by the blue side is more advanced.

2) INFERIORITY CASE

Inferiority indicates the blue side is in great danger. This initial situation (see Table 7) shows that the enemy plane is occupying a favorable position and is chasing us.

The simulated trajectories are shown in Fig. 9(b). The red UAV was in a superior position at initial, in chasing of the blue UAV. Meanwhile, blue side tried to get rid of red side through climbing and diving. The blue UAV got rid of the red UAV through a sudden change of direction. That is to say, although the blue UAV was at an absolute disadvantage at first, it still could reverse the situation in less than one minute through better maneuvers, and finally turned defeat into victory. The whole process cost about 220 steps (55s).

Likewise, 20 simulation tests were conducted, and the overturn occurred in 14 times out of 20. This result shows that even if the blue UAV is at a disadvantage at the beginning, it still has a high probability of winning. This result also proves that the blue UAV has better maneuver decision-making algorithm.

B. TARGET ALLOCATION DECISION-MAKING SIMULATIONS

Two different simulation cases are carried out in this part. Both cases are simulated many times based on three different algorithms. The parameter settings of the algorithm are as follows:

(1) Genetic algorithm: The initial number of individuals is set as 50, the crossover probability is set as 90%, and the mutation probability is set as 5%;

(2) Standard HPSO: The initial particle number is set as 5;

(3) Optimized HPSO: The initial particle number is set as 5, the number of consecutive adjacent iterations is set as 15, the decision parameter ϖ is set as 0.1.

The number of iterations is all set as 50 for the three algorithms.

1) 8-TO-8 TARGET ALLOCATION CASE

The initial states of eight blue UAVs (our side) and eight red UAVs (enemy side) are given in Table 8.

TABLE 8. UAVs' initial states of 8-to-8 case.

Initial state	x_0^i /m	y_0^i /m	z_0^i /m	ψ_0^i /°	θ_0^i /°	ϕ_0^i /°	V/m $\cdot\,s^{-1}$
Blue 1	1500	500	1800	-40	0	0	220
Blue 2	4600	600	1300	-30	0	0	220
Blue 3	2700	0	1600	-20	0	0	240
Blue 4	3900	720	1400	-10	0	0	260
Blue 5	1300	270	1700	0	0	0	250
Blue 6	660	450	1500	10	0	0	270
Blue 7	3000	590	1700	20	0	0	230
Blue 8	3700	110	1400	30	0	0	230
Red 1	4500	1500	1700	-10	0	0	250
Red 2	2900	1000	1500	-10	0	0	260
Red 3	3200	1250	1400	-5	0	0	220
Red 4	1500	800	1600	-5	0	0	220
Red 5	1000	800	1400	0	0	0	270
Red 6	3300	560	1700	0	0	0	260
Red 7	3500	1200	1300	10	0	0	260
Red 8	2000	700	1500	15	0	0	240

TABLE 9. Fitness value results of 8-to-8 target allocation.

Algorithm	Exact solution	Genetic algorithm	Standard HPSO	Optimized HPSO
Maximum value	5.9955	5.8861	5.9955	5.9955
Minimum value	5.9955	5.6113	5.7251	5.8604
Mean value	5.9955	5.7127	5.8933	5.9149
Standard deviation	0	0.171	0.113	0.089
Maximum value difference/%	0	1.85	0	0
Mean value difference/%	0	4.72	1.71	1.34
Average computing time/s	0	0.216	0.099	0.105

TABLE 10. UAVs' initial states of 8-to-6 case.

Initial state	x_0^i /m	y_0^i /m	z_0^i /m	ψ_0^i /°	θ_0^i /°	ϕ_0^i /°	V/m $\cdot\,s^{-1}$
Blue 1	4300	-450	1700	20	0	0	280
Blue 2	2300	-300	1500	30	0	0	270
Blue 3	3000	-50	1900	40	0	0	260
Blue 4	4600	-200	1300	50	0	0	290
Blue 5	3700	-600	1600	80	0	0	270
Blue 6	5200	0	1900	70	0	0	270
Blue 7	3300	-290	1500	60	0	0	260
Blue 8	4200	-600	1400	90	0	0	290
Red 1	3500	2200	1600	60	0	0	240
Red 2	3950	900	2000	50	0	0	250
Red 3	3300	1860	1800	40	0	0	280
Red 4	2600	1000	1500	60	0	0	280
Red 5	5300	1400	1500	70	0	0	270
Red 6	3300	560	1400	80	0	0	290

All three algorithms are simulated for 20 times, as shown in Table 9. As will be readily seen, the result of the genetic algorithm is the worst and that of the optimized HPSO algorithm is the best. The standard deviation of the optimized HPSO algorithm completely is the smallest. And the maximum, minimum and mean value of the optimized HPSO algorithm's solution are all better than those of the other two's. Besides, the average computing time of the optimized HPSO algorithm (0.105 s) is much less than that (0.216 s) of the genetic algorithm, but just about 6% higher than that (0.099 s) of the standard HPSO. That is, the optimized HPSO algorithm can bring better target allocation results, while ensure a relatively short computing time.

Fig. 10(a) and Fig. 11(a)–(c) show the results of one certain simulation. As shown in Fig. 10(a), the final fitness value of the optimized HPSO algorithm is the highest. A higher fitness value means our side poses a larger threat to the enemy. Fig. 11(a)–(c) are the three different allocation results. Obviously, the allocation result of Fig. 11(a) is not as reasonable as that of the other two figures, which is consistent with its minimum fitness value. Comparing Fig. 11(b) and Fig. 11(c), it can be found that the matching combination between B_1, B_5, B_6 and R_4, R_5, R_8 delivers an obvious difference. The standard HPSO algorithm's allocation result is $B_1 \rightarrow R_5$, $B_5 \rightarrow R_4$, $B_6 \rightarrow R_8$, and the corresponding fitness value of this matching combination is 1.953. The optimized one's allocation result is $B_1 \rightarrow R_4$, $B_5 \rightarrow R_8$, $B_6 \rightarrow R_5$, and the fitness value is 2.167, which is better for blue side. This is because the UAVs' initial states, especially the ψ_0^i, will result in their preferring to deal with enemies in specific areas.

Therefore, the optimized HPSO algorithm has better result than the other two, which shows that our optimization is successful.

2) 8-TO-6 TARGET ALLOCATION CASE

The initial states of eight blue UAVs (our side) and six red UAVs (enemy side) are given in Table 10.

All three algorithms are simulated 20 times as well and the relevant data are shown in Table 11. Similar to the 8-to-8 case, the average computing time of the optimized HPSO algorithm (0.125 s) is much less than that of the genetic algorithm, but merely a little more than that (0.112 s) of the standard HPSO. And the maximum, minimum and mean value of the optimized HPSO algorithm's solution are better as well. The smallest standard deviation proves once again that our optimization on the HPSO algorithm is valuable and meaningful. That is, the performance of the improved HPSO algorithm is also superior when it comes to uneven air combat cases.

Fig. 10(b) shows the fitness value change of the three algorithms. Fig. 11(d)–(f) show the target allocation results. Similarly, the optimized HPSO algorithm obtains best result. Comparing Fig. 11(e) and Fig. 11(f), there are two main differences. Firstly, in one-to-one matching, the optimized HPSO algorithm's results are $B_8 \rightarrow R_3$, $B_1 \rightarrow R_1$ while the standard HPSO's results are $B_8 \rightarrow R_1$, $B_1 \rightarrow R_3$. The speed of B_1 is much faster than that of R_1, which results in a greater advantage and a higher fitness value when B_1 fights against R_1 rather than R_3. Secondly, in many-to-one matching, the optimized HPSO algorithm's results are $B_3, B_5 \rightarrow R_6, B_5 \rightarrow R_2$ while the standard HPSO's results are $B_3, B_7 \rightarrow R_6$, $B_7 \rightarrow R_2$. The initial direction of B_7 is much closer to R_2 than

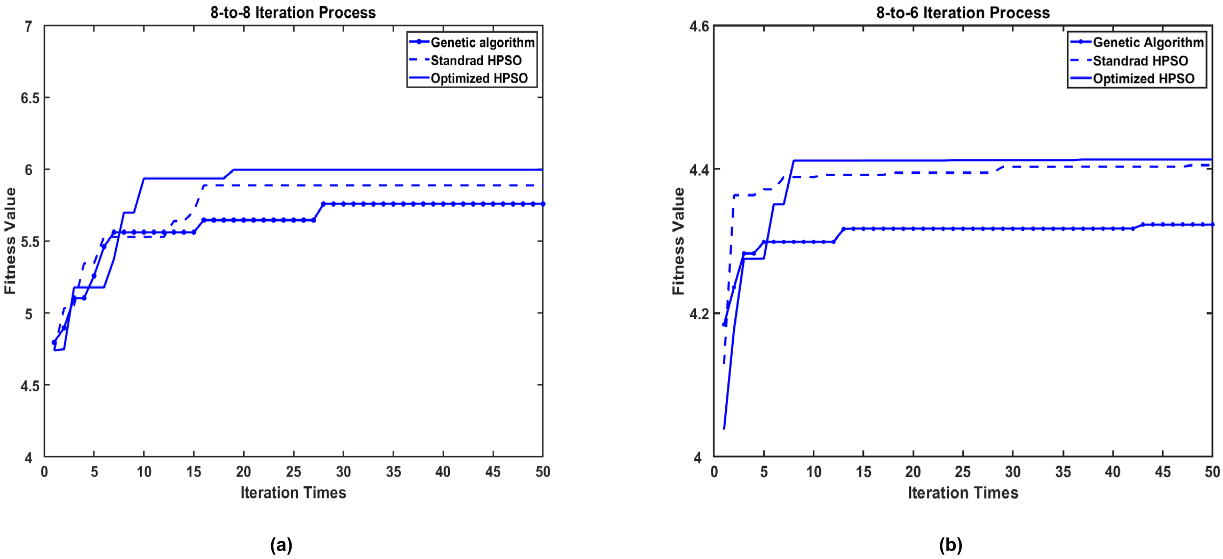


FIGURE 10. Iteration process of (a) 8-to-8 target allocation case and (b) 8-to-6 target allocation case.

TABLE 11. Fitness value results of 8-to-6 target allocation.

Algorithm	Exact solution	Genetic algorithm	Standard HPSO	Optimized HPSO
Maximum value	4.4136	4.3205	4.4136	4.4136
Minimum value	4.4136	4.0912	4.1987	4.3055
Mean value	4.4136	4.2441	4.3213	4.3429
Standard deviation	0	0.166	0.126	0.097
Maximum value difference/%	0	2.11	0	0
Mean value difference/%	0	3.84	2.09	1.60
Average computing time/s	0	0.237	0.112	0.125

R_6. Therefore, there would be a small advantage in direction if B_7 fights against R_2, but a great advantage if B_7 fights against R_6. It is more suitable for B_5 to fight against R_6 instead of B_7.

3) 8-TO-8 COOPERATIVE COMBAT CASE

The 8-to-8 case is selected for the simulation example of many-to-many CAAC. The initial states information of blue and red UAVs have been shown in Table 8. The target allocation results derived from the optimized HPSO algorithm are $B_1 \to R_4, B_2 \to R_1, B_3 \to R_6, B_4 \to R_7, B_5 \to R_8, B_6 \to R_5, B_7 \to R_2, B_8 \to R_3$, as shown in Fig. 11(c). The simulated trajectories are shown in Fig. 12.

Soon after the air combat began (11 steps of simulation), R_2 was quickly defeated by B_7 due to the great disadvantage caused by their initial states. Then, according to the annihilation principle, the target allocation decision-making algorithm determined a new target R_3 for B_7. After 47 steps, R_1 was defeated by B_2. B_2 then chose R_8 as the next target. After 91 steps, B_4 defeated R_7 and began to attack R_4. After 94 steps, R_3 soon fell into a disadvantage under the siege of B_7 and B_8, and was finally defeated by B_7. Based on the proximity principle, R_4 was also assigned to R_7 and R_8 as the next target. R_7 and R_8 took this opportunity to approach other friendly UAVs. After 104 steps, R_6 was shot down by B_3. B_3 then began to chase R_4 as well. At this time, R_4 was the target of B_1, B_3, B_4, B_7 and B_8 at the same time, and was falling into a huge disadvantage. However, B_7 and B_8 were far away, the main threat to R_4 actually came mainly from B_1, B_3, B_4. Not surprisingly, after 191 steps, R_4 was defeated by B_3. At this very moment, there were only R_5 and R_8 left in the red side. Through the calculation of the target allocation decision-making algorithm, the new match result was $B_1 \to R_5, B_2 \to R_8, B_3 \to R_8, B_4 \to R_5, B_5 \to R_8, B_5 \to R_5, B_7 \to R_8, B_8 \to R_8$. After 201 steps, B_5 and B_6 had achieved great advantages in fighting, indicating that they could quickly defeat their targets on their own without help of other UAVs. Therefore, it is considered that the 8-to-8 air combat is over. Total 201 steps are equivalent to 50 s in real world.

The utility value changes of the blue and red side UAVs are shown in Fig. 13. The utility value of the blue side was greater than that of the red side in the process of the 8-to-8 air combat, indicating advantageous situations in the combat. Consistent with the analysis in the previous paragraph, the UAVs' utility value would suddenly change at time step 11, 47, 91, 94, 104, 191 and 201, indicating the following three situations: (1) The UAV was defeated and its utility value would be reset to 0, such as R_2 at time step 11 and R_1 at time step 47; (2) The UAV was pursued by several enemy UAVs and its utility would suddenly decrease a lot, such as R_3 at time step 91 and R_6 at time step 97; (3) The UAV got a new target after defeating its original target, such as B_3 at time step 104 and B_7 at time step 191. This strongly proves that both the

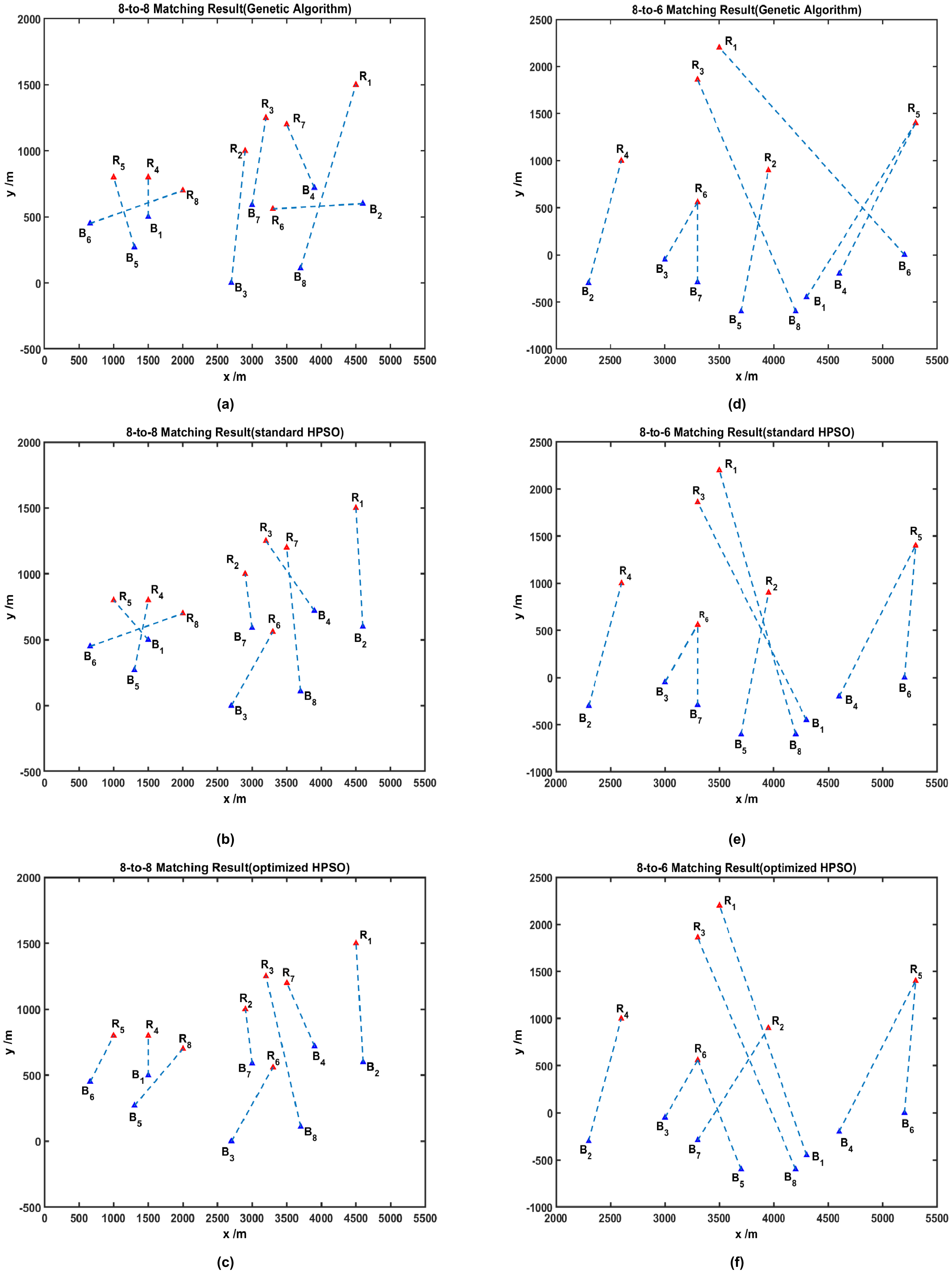


FIGURE 11. Results of 8-to-8 target allocation case based on (a) Genetic Algorithm, (b) standard HPSO and (c) optimized HPSO; Results of 8-to-6 target allocation case based on (d) Genetic Algorithm, (e) standard HPSO and (f) optimized HPSO.

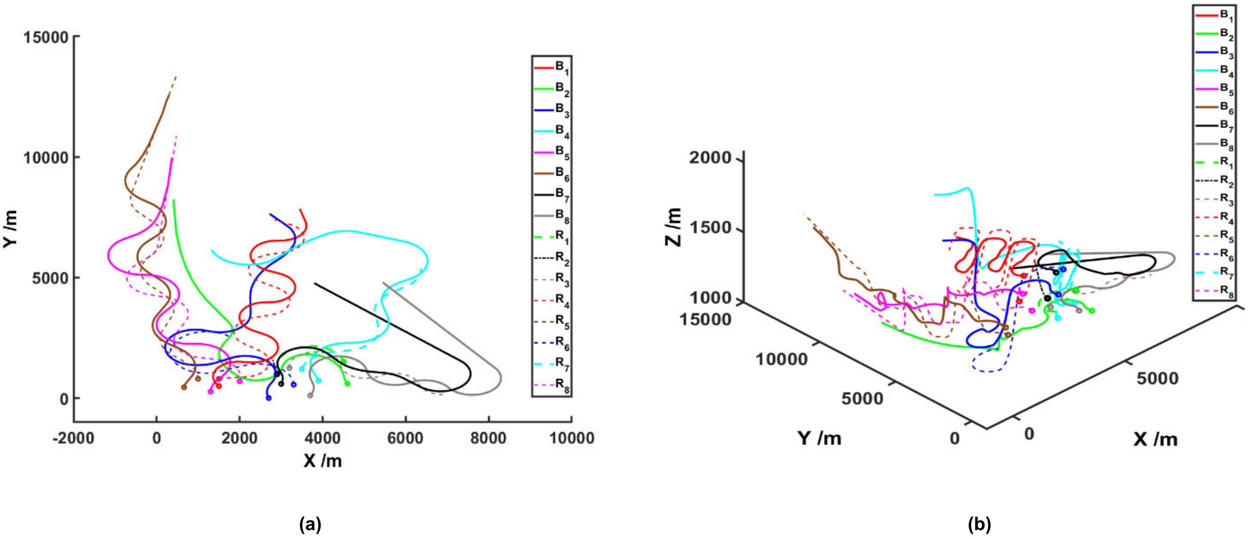


FIGURE 12. Flight trajectories of the 8-to-8 case: (a) Top view and (b) 3D view.

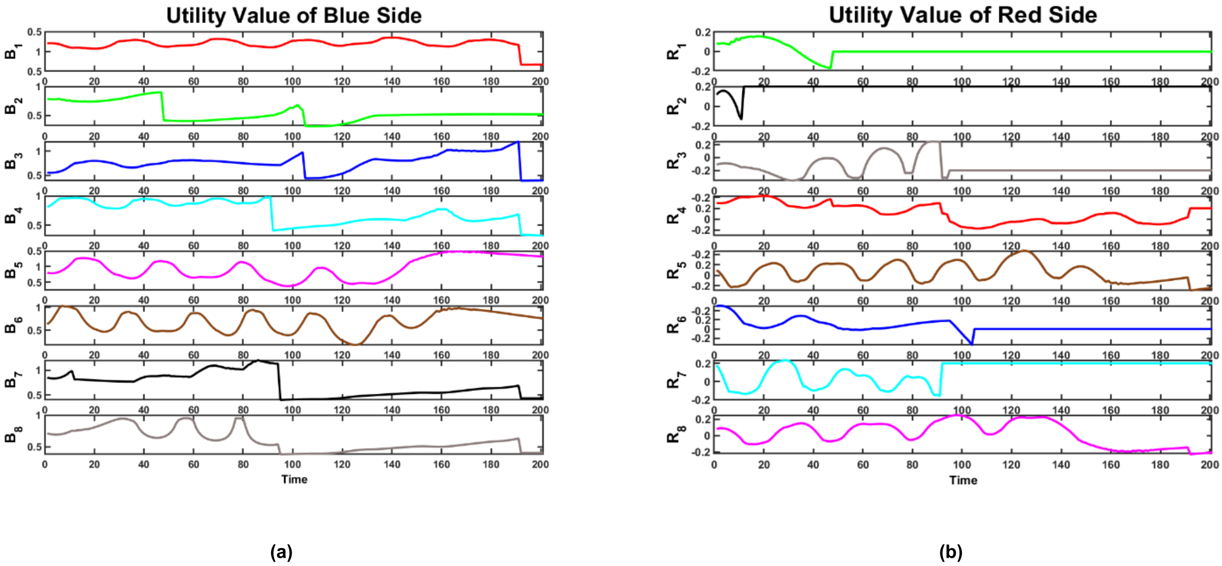


FIGURE 13. Utility value curve of the (a) blue side and (b) red side UAVs.

maneuver and the target allocation decision-making method designed above can meet the requirements of handling many-to-many cooperative autonomous air combat.

IV. CONCLUSION

In this paper, we have successfully designed a modular and high-fidelity autonomous air combat system. We have made some proper improvements on both the maneuver decision-making and target allocation decision-making of cooperative autonomous air combat. A sliding mode control is especially involved when conducting maneuvers, so that the reliability and authenticity of the decision-making and trajectory results are greatly improved. A method to update the prior probability in the dynamic Bayesian network is applied, so that the problem of lacking enough expert knowledge is solved to a certain extent. An unique visual way to pruning the minimax search tree is applied, so that the decision-making model's complexity is appropriately reduced. A method of introducing immigration particles into the hybrid particle swarm optimization is proposed, so that the global optimality of the matching result is ensured. A principle for extending the target allocation algorithm's application scenario is proposed, so that the algorithm is able to handle air combat cases with unequal number of participants. We also put forward a new method about judging the end of air combat. A fire control is creatively introduced in the final stage of autonomous air combat. The fire control ensures that the intelligence of the enemy aircraft is not underestimated, hence one's attack can be evaded. This kind of combat end criterion enhances the fidelity of simulation model and the confidence of simulation results. Therefore, it can be said that our work offers promising opportunities for the development of autonomous air combat system.

However, there are still some limitations in our work. Some contents are worthy of in-depth study in the future:

(1) Combat situations are usually more comprehensive in real air combats. There is also a high probability that several situations may overlap. The combat situations in this paper are divided into four categories: Superiority, Inferiority, Neutrality and Mutual Disadvantage. These four situations are typical, but not precise and specific enough. And when making decisions, the four situations are regarded independent of each other. Therefore, it is needed to make a more detailed division of combat situations and take the coupling between situations into consideration in future works.

(2) There are various fighting concepts of many-to-many combat. Each concept has its own advantages and disadvantages, and its corresponding target allocation criteria also has differences. In this paper, we considered a fighting concept about utility, which means the criteria is to maximize the swarm's total utility value. However, combat efficiency and combat loss ratio, et al. can also be applied as criteria for target allocation. For example, when efficiency is the priority, it is reasonable for one side to pair several UAVs in Superiority with one enemy UAV in Inferiority at the same time. Through forming multiple many-to-one pairs, the enemy will be downsized rapidly. Therefore, the impact of different fighting concepts on target allocation results should be studied in the follow-up.

REFERENCES

[1] F. Austin, G. Carbone, M. Falco, H. Hinz, and M. Lewis, "Game theory for automated maneuvering during air-to-air combat," *J. Guid., Control, Dyn.*, vol. 13, no. 6, pp. 1143–1149, Nov. 1990, doi: 10.2514/3.20590.

[2] K. Virtanen, T. Raivio, and R. P. Hamalainen, "Modeling Pilot's sequential maneuvering decisions by a multistage influence diagram," *J. Guid., Control, Dyn.*, vol. 27, no. 4, pp. 665–677, Jul. 2004, doi: 10.2514/1.11167.

[3] K. Virtanen, J. Karelahti, and T. Raivio, "Modeling air combat by a moving horizon influence diagram game," *J. Guid., Control, Dyn.*, vol. 29, no. 5, pp. 1080–1091, Sep. 2006, doi: 10.2514/1.17168.

[4] Y. Dong, J. Ai, and J. Liu, "Guidance and control for own aircraft in the autonomous air combat: A historical review and future prospects," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 233, no. 16, pp. 5943–5991, Dec. 2019, doi: 10.1177/0954410019889447.

[5] Y.-F. Li, J.-P. Shi, W. Jiang, W.-G. Zhang, and Y.-X. Lyu, "Autonomous maneuver decision-making for a UCAV in short-range aerial combat based on an MS-DDQN algorithm," *Defence Technol.*, vol. 18, no. 9, pp. 1697–1714, Sep. 2022, doi: 10.1016/j.dt.2021.09.014.

[6] K. Yang, W. Dong, M. Cai, S. Jia, and R. Liu, "UCAV air combat maneuver decisions based on a proximal policy optimization algorithm with situation reward shaping," *Electronics*, vol. 11, no. 16, p. 2602, Aug. 2022, doi: 10.3390/electronics11162602.

[7] C.-P. Du, Z.-X. Dai, Y. Zheng, D.-Y. Yin, and Y.-X. Wang, "A target allocation method inspired by Hungarian algorithm," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, Aug. 2018, pp. 819–825, doi: 10.1109/ICInfA.2018.8812329.

[8] Q. Li, F. Wang, Z. Liu, and Y. Li, "Air combat manoeuvre strategy algorithm based on two-layer game decision-making and the distributed MCTS method with double game trees," *Syst. Sci. Control Eng.*, vol. 10, no. 1, pp. 811–821, Sep. 2022, doi: 10.1080/21642583.2022.2123058.

[9] Y. Cao, W. Wei, and Y. Bai, "Multi-base multi-UAV cooperative reconnaissance path planning with genetic algorithm," *Cluster Comput.*, vol. 22, pp. 5175–5184, May 2019, doi: 10.1007/s10586-017-1132-9.

[10] J. Schwarzrock et al., "Solving task allocation problem in multi unmanned aerial vehicles systems using swarm intelligence," *Eng. Appl. Artif. Intell.*, vol. 72, pp. 10–20, Jun. 2018, doi: 10.1016/j.engappai.2018.03.008.

[11] P. Wen and J. Zhang, "Research on task allocation of multi-UAVs based on improved particle swarm optimization algorithm," in *Proc. 13th Int. Conf. Signal Process. Syst. (ICSPS)*, May 2022, pp. 169–178, doi: 10.1117/12.2631463.

[12] Y. Wang, Y. Shi, and Y. Liu, "Research on improved genetic simulated annealing algorithm for multi-UAV cooperative task allocation," in *Proc. ISMII*, Dalian, China, Jan. 2022, Art. no. 012081, doi: 10.1088/1742-6596/2246/1/012081.

[13] Q.-V. Pham, D. C. Nguyen, S. Mirjalili, D. T. Hoang, D. N. Nguyen, P. N. Pathirana, and W.-J. Hwang, "Swarm intelligence for next-generation networks: Recent advances and applications," *J. Netw. Comput. Appl.*, vol. 191, Oct. 2021, Art. no. 103141, doi: 10.1016/j.jnca.2021.103141.

[14] W. B. Herbst, "Dynamics of air combat," *J. Aircr.*, vol. 20, no. 7, pp. 594–598, Jul. 1983, doi: 0.2514/3.44916.

[15] B. L. Stevens and F. L. Lewis, "Building the aircraft model," in *Aircraft Control and Simulation*, vol. 4, 2nd ed. New York, NY, USA: Wiley, 1992, ch. 2, sec. 2, pp. 80–81.

[16] B. L. Stevens and F. L. Lewis, "Building the aircraft model," in *Aircraft Control and Simulation*, vol. 3, 2nd ed. New York, NY, USA: Wiley, 1992, ch. 2, sec. 2, pp. 63–64.

[17] A. Ortiz, S. Garcia-Nieto, and R. Simarro, "Comparative study of optimal multivariable LQR and MPC controllers for unmanned combat air systems in trajectory tracking," *Electronics*, vol. 10, no. 3, p. 331, Feb. 2021, doi: 10.3390/electronics10030331.

[18] A. Raza, F. M. Malik, N. Mazhar, and R. Khan, "Two-time-scale robust output feedback control for aircraft longitudinal dynamics via sliding mode control and high-gain observer," *Alexandria Eng. J.*, vol. 61, no. 6, pp. 4573–4583, Jun. 2022, doi: 10.1016/j.aej.2021.10.017.

[19] B. Kang, Y. Miao, F. Liu, J. Duan, K. Wang, and S. Jiang, "A second-order sliding mode controller of quad-rotor UAV based on PID sliding mode surface with unbalanced load," *J. Syst. Sci. Complex.*, vol. 34, no. 2, pp. 520–536, Nov. 2020, doi: 10.1007/s11424-020-9306-6.

[20] J. Zhou and J. Yang, "Smooth sliding mode control for missile interception with finite-time convergence," *J. Guid., Control, Dyn.*, vol. 38, no. 7, pp. 1311–1318, Jul. 2015, doi: 10.2514/1.G000912.

[21] L. Pugi, "A sliding mode course control for robust path following of a fixed wing UAV," in *Proc. IFToMM*, Naples, Italy, Sep. 2022, pp. 370–378, doi: 10.1007/978-3-031-10776-4_43.

[22] G. He, L. Yu, H. Huang, and X. Wang, "A nonlinear robust sliding mode controller with auxiliary dynamic system for the hovering flight of a tilt tri-rotor UAV," *Appl. Sci.*, vol. 10, no. 18, p. 6551, Sep. 2020, doi: 10.3390/app10186551.

[23] L.-X. Hou, L.-X. Mao, H.-C. Liu, and L. Zhang, "Decades on emergency decision-making: A bibliometric analysis and literature review," *Complex Intell. Syst.*, vol. 7, no. 6, pp. 2819–2832, Jul. 2021, doi: 10.1007/s40747-021-00451-5.

[24] A. Munir, A. Aved, and E. Blasch, "Situational awareness: Techniques, challenges, and prospects," *AI*, vol. 3, no. 1, pp. 55–77, Jan. 2022, doi: 10.3390/ai3010005.

[25] Q. Pan, D. Zhou, J. Huang, X. Lv, Z. Yang, K. Zhang, and X. Li, "Maneuver decision for cooperative close-range air combat based on state predicted influence diagram," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, Jul. 2017, pp. 726–731, doi: 10.1109/ICInfA.2017.8079001.

[26] S. Kabir and Y. Papadopoulos, "Applications of Bayesian networks and Petri nets in safety, reliability, and risk assessments: A review," *Saf. Sci.*, vol. 115, pp. 154–175, Jun. 2019, doi: 10.1016/j.ssci.2019.02.009.

[27] Z. Zhang et al., "Review of system of systems combat effectiveness evaluation and optimization methods," *J. Simul.*, vol. 34, no. 2, pp. 303–313, Oct. 2022, doi: 10.1016/j.ssci.2019.02.009.

[28] Z. Sun, "Multi-agent hierarchical policy gradient for air combat tactics emergence via self-play," *Eng. Appl. Artif. Intel.*, vol. 98, pp. 1–2, Dec. 2020, doi: 10.1016/j.engappai.2020.104112.

[29] N. Rao, S. Kashyap, G. Gopalaratnam, and D. Mandal, "Situation and threat assessment in BVR combat," in *Proc. AIAA Guid., Navigat., Control Conf.*, Aug. 2011, pp. 1–6, doi: 10.2514/6.2011-6241.

[30] A. Mehta, Y. Kunjadiya, A. Kulkarni, and M. Nagar, "Exploring the viability of conversational AI for non-playable characters: A comprehensive survey," in *Proc. 4th Int. Conf. Recent Trends Comput. Sci. Technol. (ICRTCST)*, Feb. 2022, pp. 96–102, doi: 10.1109/ICRTCST54752.2022.9782047.

[31] M. Świechowski, K. Godlewski, B. Sawicki, and J. Mańdziuk, "Monte Carlo tree search: A review of recent modifications and applications," *Artif. Intell. Rev.*, early access. [Online]. Available: https://link.springer.com/article/10.1007/s10462-022-10228-y, doi: 10.1007/s10462-022-10228-y.

[32] Y. Dong, J. Huang, and J. Ai, "Visual perception-based target aircraft movement prediction for autonomous air combat," *J. Aircr.*, vol. 52, no. 2, pp. 538–552, Mar. 2015, doi: 10.2514/1.C032764.

[33] A. Kline, D. Ahner, and R. Hill, "The weapon-target assignment problem," *Comput. Oper. Res.*, vol. 105, pp. 226–236, May 2019, doi: 10.1016/j.cor.2018.10.015.

[34] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. ICNN*, Perth, WA, Australia, Nov. 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[35] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proc. Congr. Evol. Comput.-(CEC)*, Jul. 1999, pp. 1945–1950, doi: 10.1109/CEC.1999.785511.

[36] M. Dadvar, H. Navidi, H. H. S. Javadi, and M. Mirzarezaee, "A cooperative approach for combining particle swarm optimization and differential evolution algorithms to solve single-objective optimization problems," *Int. J. Speech Technol.*, vol. 52, no. 4, pp. 4089–4108, Jul. 2021, doi: 10.1007/s10489-021-02605-x.

[37] L. Kong, J. Wang, and P. Zhao, "Solving the dynamic weapon target assignment problem by an improved multiobjective particle swarm optimization algorithm," *Appl. Sci.*, vol. 11, no. vol. 19, p. 9524, Oct. 2021, doi: 10.3390/app11199254.

[38] W. He, X. Qi, and L. Liu, "A novel hybrid particle swarm optimization for multi-UAV cooperate path planning," *Int. J. Speech Technol.*, vol. 51, no. 10, pp. 7350–7364, Mar. 2021, doi: 10.1007/s10489-020-02082-8.

[39] L. Yeding and D. Jiyang, "Research on UAV task assignment based on hybrid particle swarm algorithm," in *Proc. 4th Int. Conf. Intell. Auto. Syst. (ICoIAS)*, May 2021, pp. 361–365, doi: 10.1109/ICoIAS53694.2021.00071.

[40] G. Aiqiang, G. Xinbao, and L. Tianpeng, "Proportional guidance method and application of gravity compensation," in *Proc. IEEE Asia–Pacific Conf. Image Process., Electron. Comput. (IPEC)*, Apr. 2021, pp. 555–559, doi: 10.1109/IPEC51340.2021.9421147.

[41] Z. A. Malik, M. F. Haydar, and J. Riaz, "Cooperative guidance scheme for flight of multiple combat UA vs against a non-accelerating moving target," in *Proc. Int. Bhurban Conf. Appl. Sci. Technol. (IBCAST)*, Jan. 2021, pp. 606–612, doi: 10.1109/IBCAST51254.2021.9393309.

[42] E. Garcia, D. W. Casbeer, and M. Pachter, "Cooperative strategies for optimal aircraft defense from an attacking missile," *J. Guid., Control, Dyn.*, vol. 38, no. 8, pp. 1510–1520, Aug. 2015, doi: 10.2514/1.G001083.

[43] A. C. Andersen, K. Pavlikov, and T. A. M. Toffolo, "Weapon-target assignment problem: Exact and approximate solution algorithms," *Ann. Oper. Res.*, vol. 312, no. 2, pp. 581–606, Jan. 2022, doi: 10.1007/s10479-022-04525-6.

[44] H. Huang, X. Zhao, and X. Zhang, "Intelligent guidance and control methods for missile swarm," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–9, Jan. 2022, doi: 10.1155/2022/8235148.

[45] Y. Wang, Z. Tang, J. Chi, Q. Chen, P. Wang, and Z. Hou, "A three dimensional guidance and control method for suicide UAV and the flight test verification," in *Proc. 6th Int. Conf. Robot. Autom. Sci. (ICRAS)*, Jun. 2022, pp. 41–46, doi: 10.1109/ICRAS55217.2022.9842262.

[46] X. Fu, J. Zhu, Z. Wei, H. Wang, and S. Li, "A UAV pursuit-evasion strategy based on DDPG and imitation learning," *Int. J. Aerosp. Eng.*, vol. 2022, pp. 1–14, May 2022, doi: 10.1155/2022/3139610.

ZHIHAO WANG received the Ph.D. degree in flight system design and engineering from Fudan University, Shanghai, China, in 2017, where he is currently pursuing the Ph.D. degree in flight dynamics and flight control. His research interests include flight simulation, nonlinear control theory, and autonomous decision-making.

LIANG WANG received the Ph.D. degree in fluid mechanics from the Department of Aeronautics and Astronautics, Fudan University, Shanghai, China, in 2021. He is currently working at the Shanghai Aerospace Systems Engineering Institute. His research interest includes flight control.

JIANLIANG AI received the bachelor's, master's, and Ph.D. degrees in aircraft design from Northwestern Polytechnical University, Xi'an, China, in 1986, 1989, and 1997, respectively. He visited the Moscow State Aviation Institute as a Visiting Scholar, from 1996 to 1997. He is currently a Professor with the Department of Aeronautics and Astronautics, Fudan University. His current research interests include aircraft design and nonlinear control and intelligent control with application on flight dynamics.

YIQUN DONG received the Ph.D. degree in flight dynamics and flight control from Fudan University, Shanghai, China, in 2016. He conducted postdoctoral research at Nanyang Technological University, Singapore, from 2016 to 2017, and University of Michigan, Ann Arbor, USA, from 2017 to 2019. He is currently an Associate Professor with the Department of Aeronautics and Astronautics, Fudan University. His current research interests include implementation and application of intelligent autonomous system based on man–machine hybrid intelligence.