Air combat manoeuvre strategy algorithm based on two-layer game decision-making and the distributed MCTS method with double game trees

Qiuni Li, Fawei Wang, Zongcheng Liu, and Yuqin Li

Aeronautics Engineering College, Air Force Engineering University, Xi’an, People’s Republic of China; Air Technical Sergeant Academy, Air Force Engineering University, Xinyang, People’s Republic of China

ABSTRACT
In view of the huge strategy space and high real-time requirement for multi-fighter air combat manoeuvre decisions, the target allocation and the manoeuvre decision model are established, respectively and the air combat strategy solving algorithm is proposed based on two-layer game decision-making and the distributed Monte Carlo strategy search method with double game trees. Moreover, the two-layer game decision-making method can precut the huge game tree strategy space, which improves the efficiency of strategy search. The distributed Monte Carlo strategy search method with double game trees can quickly search out the optimal decision scheme of an air combat game based on the opponent’s strategy. The experiment results show that the designed algorithm is effective and improves the efficiency of the decision compared with the Monte Carlo search algorithm of single-layer decision-making.

1. Introduction
In the process of air combat, if a long-range air combat nip and tuck and has no outcome, the combat will generally turn into close-range air combat. Close-range air combat is a game process with the target and has the characteristics of high dynamic and fierce confrontation, as pointed out in Zhang et al. (2020) and Ji et al. (2020). Fighters make corresponding manoeuvre decisions according to the rapidly changing situation information. The purpose of air combat manoeuvre decisions is to obtain the optimal air combat situation, to threaten the target fighter and carry out effective attacks and even to get rid of the lock of the target fighter and get out of danger by manoeuvring when the fighter is at the inferior position. Therefore, the implementation of every manoeuvre will directly affect the development of the whole process of air combat.

The decision algorithms of air combat manoeuvre can be actually divided into two categories. One is a game method that considers the strategies of two or more sides. The main feature of this type of method is to take the influence of the opponent’s strategy on the situation into the analysis and emphasize the antagonism between two or more sides. The other one is a unilateral optimization method, which focuses on the optimization of its own strategy and does not perform a predictive analysis of the opponent’s strategy.

For the above categories, the first category is mainly represented by matrix games and differential games. Mylvaganam et al. (2017) applied the differential game model to air combat decision-making. Xi et al. (2020) proposed a manoeuvre decision-making method that could reflect the characteristics of multi-fighter cooperative air combat before meeting the enemy combining multi-target threat and fighter equipment characteristics. An iterative algorithm of an online integral strategy combining approximate dynamic programming and a zero-sum game was proposed by Mei et al. (2017). Luo and Meng (2017) used a multi-state transition Markov network to construct a manoeuvring decision network, which satisfies the real-time requirements of air combat game decision-making but did not use network parameters for learning. Li, Ding, et al. (2019) established an intuitive fuzzy game model and proposed a nonlinear programming method for UAV (unmanned aerial vehicle) air combat manoeuvring decision-making. An algorithm combining game theory and deep reinforcement learning to study the manoeuvring decision problem of close-range air combat was proposed by Ma et al. (2020). The authors researched the problem of solving the Nash
Monte Carlo search algorithm is used to select the game to explosive growth of the game solution space. The lots of players are involved in the game, which will lead, the fighters are always in high-speed motion, and the combat styles are complicated and diverse. More-bat has no fixed strategies and undecided methods, and continuous air combat game, the multi-fighter air dynamics. In the process of the multi-fighter multi-round complex factors such as uncertainty, incompleteness and confrontational air combat environment often contains results of decision-making will be affected. The intensely and the real-time performance of decision-making and the development of air combat situations in real combat, is difficult for traditional methods to simulate and predict. It is worth noting that the above two categories of traditional methods do not consider the manoeuvre constraints in manoeuvre decisions, which may solve some decision results that pilots cannot fly in actual combat. In addition, some traditional methods refer to the sequential game as the chess and cards problem to study the air combat manoeuvre decision-making problem. Actually, in real air combat, the decision makers do not take turns to make decisions as in the chess and card problems, they make decisions at the same time. Therefore, it is difficult for traditional methods to simulate and predict the development of air combat situations in real combat, and the real-time performance of decision-making and results of decision-making will be affected. The intensely confrontational air combat environment often contains complex factors such as uncertainty, incompleteness and dynamics. In the process of the multi-fighter multi-round continuous air combat game, the multi-fighter air combat has no fixed strategies and undecided methods, and the combat styles are complicated and diverse. Moreover, the fighters are always in high-speed motion, and lots of players are involved in the game, which will lead to explosive growth of the game solution space. The Monte Carlo search algorithm is used to select the game strategies in decision-making. And the MCTS algorithm introduces the idea of reinforcement learning based on trial sampling into manoeuvring decision-making simulates and evaluates the air combat process through the iterative process of the algorithm, which is equivalent to filtering and optimizing the policy search space. It is suitable for solving this kind of problem with huge decision space. However, the traditional MCTS method solves the decisions of both sides on the same game tree, which is a typical rotational decision-making method. Therefore, this paper improves the traditional MCTS method and proposes a double game tree MCTS method. The air combat manoeuvre decision problem is regarded as a game problem in which both sides make decisions at the same time, which is closer to the actual combat situation. In order to reduce the search space and time and prevent the explosive growth of the game solution space, this paper improves the search strategy of the traditional MCTS method and proposes a two-layer game decision-making search strategy, which decomposes the manoeuvre decision problem into two stages: target allocation and manoeuvre decision. So decision-making is equivalent to pruning the game tree, which improves the real-time performance of the algorithm. Game decision-making refers to choosing the strategy that will lead to the most powerful result for one’s side in a competitive or adversarial relationship. In addition, this paper considers the manoeuvre constraints to prevent the resulting actions of manoeuvre decision from being actions that the pilot cannot fly.

The rest of our paper is organized as follows: In section 2, we present the dominant function model for air combat in detail; In section 3, we describe and analyze the proposed air combat decision scheme and algorithm based on two-layer game decision-making and double game tree distributed Monte Carlo search; After that, we confirm the effectiveness and convergence of the proposed method in section 4 while a conclusion is presented in the final section.

2. Dominant function model of air combat

This paper divides the common thinking of the multi-fighter air combat game decision-making process into two-layer of decision-making. The target allocation decision is made in the first layer and the air combat manoeuvring decision is made in the second layer, after determining the target allocation. First of all, the dominant function model of air combat should be established in order to achieve the situation assessment of air combat and to make multi-fighter air combat target allocation and manoeuvre decision based on the assessment results. Many factors affect the air combat situation. But in
comparison, the angle factor and the distance factor are usually of great significance to the air combat situation. Therefore, this paper establishes the dominant function model for air combat manoeuvre decisions based on the angle and distance factors. In addition to considering the situation of air combat, more factors, such as the performance of the opponent’s fighter and the multi-fighter synergy effect, should also be considered. Therefore, this paper makes air combat target allocation decisions based on the factors of angle, distance and fighter performance advantages, as well as the changes in the threat index of multi-fighter coordination.

2.1. The angular dominant function

The influence of the angle between two fighters on the attack situation is called the angular dominant function. The angular dominant function is established in the case that the angle is aiming at the opponent will have the advantage in air combat and is being aimed by the opponent will have the disadvantage. As shown in the figure below, suppose A is the attack fighter and D is the target fighter. The target line $d_{AD}$ is defined as the line between the attack fighter A and the target fighter D. $\phi_{AD}$ is the angle between the velocity vector of the attack fighter A and the target line. $q_{AD}$ is the angle between the velocity vector of the target fighter D and the target line. Then, the value of the angular dominant function is defined as follows:

$$S_a = 1 - \frac{|\phi_{AD}| + |q_{AD}|}{\pi}$$ (1)

2.2. The distance dominant function

The influence of the distance between two fighters on the attack situation is called the distance dominant function. The distance dominant function represents the distance advantage of the two fighters and the influence of the distance changing after the air combat decision. $d_{AD}$ is the distance between the i-th fighter of N and the j-th fighter of M. $d_{imissle}$ and $d_{irader}$ are the missile ranges of the i-th fighter of N and the radar maximum detection range of the i-th fighter of N, respectively. Generally, the maximum detection range of the radar is farther than the range of missiles. Suppose that the distance advantage is 0 outside the radar detection range and the distance advantage is 1 inside the missile range. When the distance is between the maximum detection range of the radar and the range of the missile, the distance advantage value increases with the decrease of the distance between the two fighters. Then the value of the distance dominant function is calculated as follows:

$$S_d = \begin{cases} 0, & d_{AD} > d_{irader} \\ \frac{(d_{irader} - d_{AD})}{(d_{irader} - d_{imissle})}, & d_{imissle} \leq d_{AD} \leq d_{irader} \\ 1, & \text{others} \end{cases}$$ (2)

2.3. The performance dominant function

Fighter performance is a key factor that air combat decision-makers need to consider. The fighter performance dominant function needs to comprehensively consider factors such as manoeuvrability, the detection ability and the firepower performance, the communication ability and the electronic countermeasure ability. Referring to Zhang et al. (2017), the performance dominant function of the fighter i is established as follows:

$$S_p = h_i / \max(h_i, h_j)$$ (3)

$$h_i = \left[ \ln B + \ln \left( \sum A_1 + 1 \right) + \ln \left( \sum A_2 + 1 \right) \right]$$

$$\times \epsilon_1 \epsilon_2 \epsilon_3 \epsilon_4 \epsilon_5$$ (4)

where $B, A_1, A_2, \epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5$ are the manoeuvring parameter, firepower parameter, detection capability parameter, control efficiency coefficient, viability coefficient, the fighter’s range coefficient, electronic countermeasure capability coefficient and communication ability coefficient, respectively. $h_i$ is the air combat performance advantage index of the i-th fighter of N and $h_j$ is the air combat performance advantage index of the j-th fighter of M. And $h_j$ is calculated analogously.

2.4. The threat index dominant function

When multiple fighters cooperate in air combat, the synergistic effect is considered in that the total threat index of one side decreases, while the total threat index of the other side increases. Therefore, after the targets are allocated, the threat index dominant function of the i-th fighter of N relative to the j-th fighter of M is calculated as follows:

$$P_{thij} = \begin{cases} 0, & d_{AD} > d_{frader} \\ \frac{(\epsilon_1 - h_i)}{\max(h_i, h_j)} / \max(h_i, h_j), & \text{asymmetry} \\ k * d_{AD} + 1, & d_{AD} \leq d_{frader} \leq d_{frader} \\ 1, & \text{others} \end{cases}$$ (5)

When the distance between the two fighters is larger than the maximum detection range of the opponent’s radar, the threat index is considered to be 0. $d_{frader}$ and $d_{frader}$ are the maximum radar detection ranges of the i-th fighter of
N and the j-th fighter of M, respectively. The parameter $k$ is the threat index and $k < 0$ suggests that the threat index decreases with the increase in distance. The threat index dominant function $P_{thi}$ of the j-th fighter of M relative to the i-th fighter of N is calculated analogously. When multiple fighters cooperate in air combat, the total threat index of N and the total threat index of M multi-fighter coordination are, respectively, calculated as the following equations, and they reflect the overall synergistic threat effect and synergistic capacity of both sides.

$$P_{thn} = 1 - \prod_{i=1}^{n} (1 - P_{thi}), i = 1, 2, \ldots, n$$

$$P_{thm} = 1 - \prod_{j=1}^{m} (1 - P_{thj}), j = 1, 2, \ldots, m$$

### 3. The air combat strategy algorithm based on two-layer game decision-making

In a multi-fighter air combat game, the target allocation is first determined according to the position, performance, threat of the opponent’s fighters and the synergy effect of their own fighters. Then, the manoeuvre decision about how to fly is determined according to the situation information such as the allocated target fighter angle and distance. Therefore, the two-layer game decision-making algorithm includes the target allocation decision and the manoeuvre decision.

#### 3.1. Target allocation decision

Suppose: a fighter from the N side should attack at least one fighter from the M side, and only one fighter from the M side can be attacked by it in a discrete short time interval. In the target allocation decision stage, three factors are considered comprehensively to make the target allocation decision. The first is the position factor of the opponent’s fighters such as the angle and distance. The second is the performance factor of the opponent’s fighters such as the manoeuvrability, electronic countermeasure capability and communication capability. The third is the effect factor of the threat index change after multi-fighter collaboration. Since the height dominant comes from the distance dominant and angle dominant, the height dominant is not considered in this paper. The speed dominant is considered in the basic manoeuvre library, and whether to accelerate or decelerate is determined through the choice of manoeuvre strategy.

Taking a fighter of N as an example, the comprehensive dominant function of the i-th fighter of N relative to the j-th fighter of M in air combat is constructed as follows:

$$S_{ij} = \lambda_1 S_{aij} + \lambda_2 S_{dij} + \lambda_3 S_{prij}$$

where $\lambda_1$, $\lambda_2$, $\lambda_3$ are weight coefficients and $\lambda_1 + \lambda_2 + \lambda_3 = 1$. $S_{aij}$, $S_{dij}$ and $S_{prij}$ are the angle dominant function, distance dominant function and performance dominant function of the i-th fighter of N relative to the j-th fighter of M under the current moment situation, respectively. They are all normalized values as shown below (Figure 1), suppose A is the attack fighter and D is the target fighter.

In summary, the total dominant function of multi-fighter cooperative air combat in the target allocation scheme can be established. The above three factors, as well as the influence of the whole system’s collaborative performance threat index, are mainly considered comprehensively in the target allocation stage. The optimal payoff of collaborative target allocation is determined by them. That is, the total payoff function of the game can be calculated as follows:

$$U_n = P_{thn} \sum_{i=0}^{n} S_{ij}$$

$$U_m = P_{thm} \sum_{j=0}^{m} S_{ji}$$

Both sides of the air combat game always try to optimize their own payoff function. Therefore, the target allocation decision function selects the scheme with the maximum total dominant function value of the multi-fighter cooperative air combat from many options. That is, the optimal response strategy of g is max $U_n$.

#### 3.2. Fighter manoeuvre decision

##### 3.2.1. Basic manoeuvre library

If an air combat game is considered as playing chess in three-dimensional space, then the game strategies can be considered as an optional position on the chessboard and choosing different game strategies will result in different payoffs. Referring to Ji et al. (2020), all the manoeuvres of fighters can usually be divided into 11 kinds of basic

![Figure 1. The position relationship between two fighters in air combat.](image-url)
manoeuvres in the tactical action planning layer in the three-dimensional space. The basic manoeuvre library can combine most tactical manoeuvres in air combat, and different combinations of manoeuvring sequences correspond to different tactical manoeuvres in air combat as expounded in Ji et al. (2020), Du et al. (2018). As shown in Figure 2, generally speaking, the possible manoeuvres of a fighter mainly include: 1. direct flight without any manoeuvre, 2. Climb, 3. Dive, 4. Turn left, 5. Climb to the left, 6. Dive to the left, 7. Turn right, 8. Climb to the right, 9. Dive to the right, 10. Flying with an acceleration. 11. Flying with a deceleration. The inclination angle can reach $-60^\circ$, $0^\circ$ and $60^\circ$, corresponding to the climbing, direct flight without any manoeuvre and dive in the manoeuvre decision, respectively, and the roll angle can reach $-30^\circ$, $0^\circ$ and $30^\circ$, corresponding to the turn left, direct flight without any manoeuvre and turn right in the manoeuvre decision, respectively.

In the existing literature on air combat decision-making, the main problem is that the constraint relationship between the current manoeuvre and the next manoeuvre is ignored. For example, if a fighter chooses to dive to the left manoeuvre, after performing it, the possible manoeuvre should only include manoeuvres 3 and 4 in the next decision cycle. Because the fighter cannot achieve the other manoeuvres such as turning right and diving to the right in the next decision-making cycle. Therefore, for the close-range air combat manoeuvre decision-making, the current manoeuvre will actually affect the range of optional manoeuvres in the next decision cycle. This paper establishes the constraint relationship between the current manoeuvre and the next manoeuvre, specifically as Table 1. To prevent the algorithm from searching out the manoeuvres is difficult for pilots.

Table 1. The constraints list of manoeuvre transfer.

| Case number | The current manoeuvre decision | The optional manoeuvres in the next decision cycle |
|-------------|--------------------------------|--------------------------------------------------|
| 1           | Direct flight without any manoeuvre | Anyone                                           |
| 2           | Climb                           | Direct flight without any manoeuvre, Climb to the left, Climb to the right |
| 3           | Dive                            | Direct flight without any manoeuvre, Dive to the left, Dive to the right |
| 4           | Turn left                       | Direct flight without any manoeuvre, Climb to the left, Dive to the right |
| 5           | Climb to the left               | Climb, Turn left                                |
| 6           | Dive to the left                | Dive, Turn left                                 |
| 7           | Turn right                      | Direct flight without any manoeuvre, Climb to the right, Dive to the right |
| 8           | Climb to the right              | Climb, Turn right                               |
| 9           | Dive to the right               | Dive, Turn right                                |
| 10          | Flying with an acceleration      | Direct flight without any manoeuvre, Flying with an acceleration |
| 11          | Flying with a deceleration      | Direct flight without any manoeuvre, Flying with a deceleration |

### 3.2.2. Distributed Monte Carlo search algorithm based on double game trees

The whole air combat process is described as a path from the root to the leaf of a multi-way game tree because it is the game process to choose a target fighter and a manoeuvre strategy for both sides. Both sides establish their own game tree in the air combat game process. After the target allocation of the first layer decision, the target of each fighter has been determined. This is equivalent to pre-pruning the game tree. Therefore, in the second layer of decision-making, we make the manoeuvring decision under the condition that the target fighter is determined. Compared with the one-layer decision, the game strategy searching space is greatly reduced and the searching efficiency is greatly improved.

Because pilots need to quickly make reasonable manoeuvre decisions in tense air combat, they need more autonomy. Therefore, the joint total payoff of each fighter is not considered, and the decision is only made based on the air combat situation and its own payoff in the second layer decision. Referring to Li, Yang, and Liu (2018), the game payoff function of fighter $i$ is designed as the following equation accordingly:

$$U_i = \theta_1 S_{aij} + \theta_2 S_{dij}$$ (11)

where $U_i$ is the payoff value of the manoeuvring decision. $\theta_1$ and $\theta_2$ are weight coefficients and $\theta_1 + \theta_2 = 1$. $S_{aij}$ and $S_{dij}$ are different from the decision process of target allocation. At present, they are the angle dominant function and distance dominant function of the $i$-th fighter of N relative to the $j$-th fighter of M under the next moment.
situation. They are also normalized values. To sum up, the air combat situation is mainly determined by the distance and angle between the two fighters.

The MCTS (Monte Carlo Tree Search) algorithm is used for the game strategy selection in air combat. The algorithm can keep a balance between exploitation and exploration, namely, it is both to ensure the best rewards from past decisions and to hope for greater rewards in future and it can master and predict game strategies more and more accurately in the game. In the process of air combat, the mastery and prediction of operational strategy are more accurate, it is more favourable to search out the best game strategy for the current and future situations. The MCTS algorithm is a method to set up a search tree to find the best decision based on the decision space of sampling in a specific field.

It includes four steps: selection, expansion, simulation and back-propagation. In order to search the air combat game node, this paper adopted a modified UCB (Upper Confidence Bound) algorithm expounded in Gameron (2012) in the MCTS frame. In the selection step, suppose the root node is the attacking side. The UCB value is calculated by Equation (12). The node with the maximum UCB value will be selected as the subsequent node.

\[
UCB = \max \{ U_i + C \sqrt{\frac{2 \ln n}{n_j}} \}
\]

where \(U_i\) is the average payoff of fighters \(i\) in the past \(t - 1\) times and this is a normalized value. \(n\) is the total number of times a game strategy is selected. \(n_j\) is the number of times for the selected game strategy. The regulatory factor is \(C\) and it is used to adjust the balance between the return value and the unexplored node. In the simulation step, the frequent change of target allocation results can be prevented by adjusting the simulation step size. In this paper, it is considered that setting the simulation step size as 8 is an ideal value after testing.

In the traditional Monte Carlo search method, both sides of the gameplay sequential games on the same game tree and make decisions, in turn, as expounded in Silver et al. (2017), Browne et al. (2012), Barriga et al. (2017). In this way, when both sides of the war make decisions on the same game tree, they will make game decisions in chronological order. For example, if \(t = 1\), the decision is made by the N side, then \(t = 2\), the decision is made by the M side and if \(t = 3\), the decision is made by the N side, then \(t = 4\), it is the M side’s turn to make a decision. Thus, the game decision is repeated, in turn, as shown in Figure 3(a). In sequential game decision-making, the player who makes the strategic choice and takes the action first usually occupies the advantageous position. The other player must choose its own strategy on the basis of the opponent’s accomplished action strategy. The most important characteristic of an air combat game is that both sides make decisions simultaneously under the current game situation environment. That is, there is no chronological order for game decisions. Due to the different mission characteristics and the high real-time requirement of the air combat game, it is not suitable to directly use the traditional Monte Carlo tree search. In the multi-fighter and multi-round continuous air combat game, not only the coordination of multi-fighter combat, but also the influence of the historical game strategies of both sides should be taken into account, and the decision of both sides should be made at the same time rather than in turn. Otherwise, it will lead to the problems such as lagging decision information. Therefore, a distributed double game tree Monte Carlo search algorithm is designed in this paper for manoeuvring decisions in air combat, which is a new MCTS method. As shown in Figure 3(b), both sides of the war establish a game tree. At \(t = 1\), \(t = 2, \ldots, t = n\), both sides make synchronous decisions in their respective game trees at every moment, and there is no need to wait for the opponent to make a decision. Meanwhile, the dominant value and decision function are calculated according to the real-time updated situation information. In this way, the battlefield situation can be perceived in real-time, the opponent’s strategies can be adapted to make decisions, and the optimal game strategy scheme can be obtained.

3.3. The final total payoff function

The final total payoff function of two-layer game decision-making is the synergistic effects of angle advantage, distance advantage and performance advantage. Therefore, the final total payoff function of both sides can be calculated by the following Equations:

\[
U_N = P_{thn} \sum_{i=0}^{n} (S_{aij} + S_{dij} + S_{pji}) \quad (13)
\]

\[
U_M = P_{thm} \sum_{j=0}^{m} (S_{aij} + S_{dij} + S_{pji}) \quad (14)
\]

where \(U_N\) and \(U_M\) are the final total payoff function value of the two-layer game decision-making of the N side and the M side, respectively. \(S_{aij}\), \(S_{dij}\) and \(S_{pji}\) are the angle dominant function, distance dominant function and performance dominant function of the \(i\)-th fighter of N relative to the \(j\)-th fighter of M. \(S_{aij}\), \(S_{dij}\) and \(S_{pji}\) are the angle dominant function, distance dominant function and performance dominant function of the \(j\)-th fighter of M relative to the \(i\)-th fighter of N, respectively. \(P_{thn}\) and \(P_{thm}\) are the total threat index of N and the total threat index of
M multi-fighter coordination. They are also all normalized values. But different from the decision process of target allocation, the variables of Equations (13) and (14) are not the values of air combat situation before manoeuvre decision. They are the values of air combat situation after manoeuvre decision.

3.4. The algorithm steps

Step 1: Initialization. Initialize the information such as the coordinate, speed and course of the combat fighters, and the weight parameters of the two-layer game decision payoff functions.

Step 2: According to the current air combat situation information, the current nodes of the double game trees were determined, and the game decision function values of various target allocation schemes under the current situation were calculated respectively, that is the total dominant function value of multi-fighter cooperative air combat.

Step 3: The first layer of game decision: determine the target allocation scheme according to the principle of maximizing their own total payoffs, and pre-pruning the subsequent MCTS.

Step 4: Combining with the basic manoeuvre library, the game payoff function values of all possible manoeuvre decision schemes at the next moment are calculated.

Step 5: The second layer of game decision: the UCB algorithm and MCTS are used to determine the manoeuvre decision scheme.

Step 6: Update the situation information of game decision trees of both sides.

Step 7: Determine whether the iteration termination condition has been met: the maximum number of iterations has been reached or the difference between the two payoff function values of two sides has reached a predetermined threshold. If the iteration terminates, the best game strategy scheme will be the output; otherwise, Step 2 to Step 6 will be repeated.

4. Experiment and result

In this experiment, we simulate a 2vs2 fighter’s air combat in three-dimensional space. The parameters in the simulation are set as follows. The initial coordinates of N side’s fighters and M side’s fighters are (1000, 1000, 6000), (2000, 0, 5500) and (0, 1000, 6000), (2000, 1000, 5500), respectively. And the initial yaw angle, track inclination angle and roll angle are set as (1,0,0), (1,0,0), (1,0,0) and (1,0,0), respectively. \( k = -1/400 \) and \( C = 0.1 \). \( \lambda_1, \lambda_2, \lambda_3 \) are set as 0.3, 0.4 and 0.3. \( \theta_1 \) and \( \theta_2 \) are set as 0.5. \( d_{\text{missile}} \) of the N side are 16 and 19, and \( d_{\text{missile}} \) of the M side are 15 and 19. \( d_{\text{rader}} \) of the N side are 500 and 800, and \( d_{\text{rader}} \) of the M side are 600 and 700. \( B, A_1, A_2, \epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5 \) of both sides are randomly generated between 0.6 and 0.9. It is assumed that the speed of all fighters can be divided into two kinds of constant speed and acceleration in the simulation. The termination condition of the algorithm was set as follows: reaching the maximum number of decision iterations 31 times, or the difference between the two payoff function values reaching the threshold value of 0.5. In a simulation, it is assumed that the basic manoeuvre library of both sides is the same and the same decision algorithm is adopted. The game solution space in the experiment is the solution of all three-dimensional space satisfying the constraints of manoeuvre and speed.

The simulation results adopting the two-layer game decision algorithm in this paper are shown in Figures 4–7.
and Table 2. Table 2 lists the results of manoeuvre decisions for all fighters. The air combat flight trajectories of four fighters are shown in Figure 4. Figure 4(a) and (b) are the three-dimensional and two-dimensional plane projection graphs of the flight trajectories, respectively. Each curve represents the combat flight trajectory of a fighter using this algorithm. The total payoff function change curves of the N side and the M side corresponding to the decision in the first layer are shown in Figure 5. Figure 5(a) represents the total payoff function of the N-side decision, and Figure 5(b) represents the total payoff function of the M-side decision. The payoff function change curves of the four fighters corresponding to the second layer decision are shown in Figure 6 and Figure 7. Figure 6(a) and (b) represent the changes in the payoff function of fighter 1 and fighter 2 of N in the whole decision-making process, respectively, while Figure 7(a) and (b) represent the changes of payoff function of fighter 3 and fighter 4 of M in the whole decision-making process, respectively.

From the experimental results, it can be seen that the algorithm designed in this paper is correct and effective. Since there are a large number of sampling and continuous simulations in the game in the two steps of extension node and simulation of the MCTS algorithm, the MCTS algorithm can more and more accurately predict the battlefield situation, more and more accurately predict the enemy’s manoeuvring intention and strategic trends and finally search out the best manoeuvring strategy scheme. It can be seen from Figure 4(a) and (b) that the two sides struggle with each other through continuous target allocation and manoeuvring decisions and the flight trajectories show a highly staggered pattern. In close-range air combat, the situation changes rapidly and is complicated, the two sides struggle fiercely, and the decision-making time is urgent. Among them, the target allocation of the No. 1 fighter of N and the No. 2 fighter of M changed at $t = 17$, while the target allocation of other fighters remained unchanged during the air combat game. From Figure 5(a) and (b), it can be seen that in the air combat game, the total payoff values of both sides and the payoff values of each fighter fluctuate constantly, indicating that both sides alternate frequently conversion from an advantage turn to a disadvantage in the process of air combat, and there are situations from an advantage turn to a balance or a disadvantage, or from a disadvantage turn to an advantage or a balance. As can be seen from the result figures, when $t = 31$, the total payoff values of the N side and the M side are 0.3785 and 0.3189 at the target allocation stage, respectively; the payoff values of fighter 1 and fighter 2 of N are 0.06162 and 0.2756, respectively; and the payoff values of fighter 1 and fighter 2 of M are 0.2368 and $-0.06162$, respectively (The negative value of the fighter’s payoff is due to the angle disadvantage). In the end, the N side ended the air combat with a slight advantage.

In order to evaluate the performance of the algorithm, we perform two groups of experiments to compare the final total payoff function values of both sides. In the two groups of experiments, the N side adopts the method presented in this paper. But the M side adopts the method presented in this paper in the first group of experiments and adopts the distributed MCTS method with double game trees of single-layer game decision-making in the second group of experiments. The final total payoff function values of the two groups of experiments are shown in Figure 8. Figure 8(a) shows the final total payoff function value of the N side in the two groups of experiments, and Figure 8(b) shows the final total payoff function value of the M side in the two groups of experiments. In Figure 8(a) and (b), blue shows the results of the first group of experiments, and red shows the results of the second group of experiments. In the first experiment, the average of the final total payoff of N is 0.7095 and the average of the final total payoff of M is 0.7040. In the second experiment, the average of the final total payoff of N is 0.7992 and the average of the final total payoff of M is 0.6395. As can be seen from the figures and experimental results, the final

| Iterations | No.1 fighter’s manoeuvre for N | No.2 fighter’s manoeuvre for N | No.1 fighter’s manoeuvre for M | No.2 fighter’s manoeuvre for M |
|------------|--------------------------------|--------------------------------|--------------------------------|--------------------------------|
| 0          | 1                              | 1                              | 1                              | 1                              |
| 1          | 1                              | 1                              | 4                              | 2                              |
| 2          | 3                              | 1                              | 4                              | 5                              |
| 3          | 1                              | 1                              | 1                              | 5                              |
| 4          | 7                              | 7                              | 7                              | 5                              |
| 5          | 8                              | 9                              | 7                              | 4                              |
| 6          | 8                              | 9                              | 7                              | 6                              |
| 7          | 8                              | 7                              | 8                              | 4                              |
| 8          | 7                              | 7                              | 7                              | 6                              |
| 9          | 9                              | 8                              | 7                              | 6                              |
| 10         | 3                              | 8                              | 9                              | 3                              |
| 11         | 6                              | 7                              | 9                              | 1                              |
| 12         | 3                              | 1                              | 9                              | 3                              |
| 13         | 1                              | 3                              | 9                              | 5                              |
| 14         | 2                              | 9                              | 7                              | 4                              |
| 15         | 8                              | 7                              | 8                              | 4                              |
| 16         | 1                              | 1                              | 1                              | 1                              |
| 17         | 2                              | 7                              | 1                              | 3                              |
| 18         | 8                              | 7                              | 1                              | 9                              |
| 19         | 8                              | 7                              | 4                              | 9                              |
| 20         | 7                              | 7                              | 5                              | 9                              |
| 21         | 9                              | 7                              | 2                              | 7                              |
| 22         | 9                              | 9                              | 5                              | 8                              |
| 23         | 7                              | 9                              | 4                              | 8                              |
| 24         | 7                              | 7                              | 6                              | 8                              |
| 25         | 1                              | 8                              | 6                              | 2                              |
| 26         | 10                             | 8                              | 6                              | 2                              |
| 27         | 3                              | 7                              | 6                              | 1                              |
| 28         | 9                              | 7                              | 4                              | 10                             |
| 29         | 7                              | 7                              | 5                              | 1                              |
| 30         | 8                              | 9                              | 2                              | 3                              |
| 31         | 8                              | 3                              | 2                              | 6                              |

Table 2. The results of manoeuvre decision.
Figure 4. The air combat flight trajectories of four fighters. (a) Three-dimensional flight trajectory (b) Two-dimensional flight trajectory.

Figure 5. The total payoff function change curves of the N side and M side in the first layer decision. (a) Total payoff of N (b) Total payoff of M

total payoff function value of the second group of experiments is better than that of the first group of experiments for the N side, and the final total payoff function value of the first group of experiments is better than that of the second group of experiments for the M side. They all explain that the method of the first group experiment on the M side is better than the method of the second group experiment. That is, the distributed MCTS method with double game trees of two-layer game decision-making is more efficient than the distributed MCTS method with double game trees of single-layer game decision-making.

In addition, the decision-making time is compared in the experiments between the distributed MCTS method with double game trees of two-layer game decision-making and the distributed MCTS method with double game trees of single-layer game decision-making.

Figure 6. The payoff function change curves of the N side in the second layer decision. (a) No.1 fighter’s payoff for N (b) No.2 fighter’s payoff for N
In 31 decisions, the average decision-making time of a two-layer game decision-making algorithm is 9 ms, and the average decision-making time of a single-layer game decision-making algorithm is 14 ms. It can be seen that the two-layer game decision-making and distributed MCTS method with the double game trees algorithm proposed in this paper are helpful to improve decision efficiency.

5. Conclusions

Aiming at the manoeuvre decision-making problem of multi-fighter air combat, this paper establishes the two-layer game decision algorithm model and proposes the distributed Monte Carlo search method based on a double game tree to solve the optimal game strategy scheme. Therefore, the algorithm can grasp and predict the air combat game situation more and more accurately in the continuous simulation game process, to accurately grasp and predict the enemy’s strategic trend in air combat as much as possible, and finally can search out the best manoeuvring strategy scheme in the current and future situations. The experiment results show that the model and algorithm designed in this paper are correct and effective and the fighters in air combat can accurately predict the opponent’s manoeuvring intention and battlefield situation and can quickly search out the optimal game decision scheme by adapting the opponent’s strategy. The MCTS is pruned in advance by the two-layer game decision-making method, which greatly reduces the time of subsequent decision searching. The distributed Monte Carlo search method with double game trees can better meet the high real-time requirement of air combat missions.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the National Natural Science Foundation of China [grant number 62106284]; Xi’an Youth Talent Promotion Plan [grant number 095920201309]; Nature Science Foundation of Shaanxi Province of China [grant number 2021JQ-370].

References

Barriga, N. A., Stanescu, M., & Buro, M. (2017). Combining strategic learning with tactical search in real-time strategy games. In B. Magerko & J. P. Rowe (Eds.), Proceedings of the Thirteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (pp. 9–15).

Browne, C., Powley, E. J., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., &
Colton, S. (2012). A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1), 1–43. https://doi.org/10.1109/TCIAG.2012.2186810

Ding, D., Wei, Z., Tang, S., & Huang, Z. (2020). Robust maneuvering decision-making method for air combat using adaptive prediction weight. *Journal of Systems Engineering and Electronics*, 42(10), 2275–2284. https://doi.org/10.3936/j.issn.1001-506X.2020.10.16

Du, H., Cui, M., Han, T., Wei, Z., Tang, C., & Tian, Y. (2018). Maneuvering decision in air combat based on multi-objective optimization and reinforcement learning. *Journal of Beijing University of Aeronautics and Astronautics*, 44(11), 2247–2256. https://doi.org/10.13700/j.issn.1001-5965.2018.0132

Gameron, B. (2012). Survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4, 1–49.

Han, T., Cui, M. L., & Zhang, W. (2020). Multi-UCAV cooperative air combat maneuvering decision. *Journal of Ordnance Equipment Engineering*, 41(4), 117–123. https://doi.org/10.11809/bqzbgcxb2020.04.023

Huang, C., Dong, K., Huang, H., & Tang, S. (2018). Autonomous air combat maneuver decision using Bayesian inference and moving horizon optimization. *Journal of Systems Engineering and Electronics*, 29(1), 86–97. https://doi.org/10.21629/JSEE.2018.01.09

Ji, H., Yu, M., Qiao, X., & Zhang, S. (2020). Application of the improved BAS-TIMS algorithm in air combat maneuver decision. *Journal of National University of Defense Technology*, 42(4), 123–133. https://doi.org/10.11887/j.cn.202004019

Li, Q. N., Yang, R. N., Haoliang, L. I., Huan, Z. H., & Chao, F. E. (2018). Modeling and game strategy analysis of suppressing IADS for multiple fighters’ cooperation. *Journal of Systems Engineering and Electronics*, 29(2), 78–86. https://doi.org/10.21629/JSEE.2018.02.10

Li, Q. N., Yang, R. N., & Liu, Z. C. (2018). Research on the multi-UAV attacking maneuvering method according to optimal potential. *Flight Dynamics*, 36(3), 28–33. https://doi.org/10.13645/j.cnki.f.d.20180208.017

Li, Q. N., Yang, R. N., & Liu, Z. C. (2019). Approach for air-to-air confrontment based on uncertain interval information conditions. *Journal of Systems Engineering and Electronics*, 30(1), 100–109. https://doi.org/10.21629/JSEE.2019.01.10

Li, S. H., Ding, Y., & Gao, Z. L. (2019). UAV air combat maneuvering decision based on intuitionistic fuzzy game theory. *Systems Engineering and Electronics*, 41(2), 1063–1070. https://doi.org/10.3969/j.issn.1001-506X.2019.05.19

Luo, Y. Q., & Meng, G. L. (2017). Research on UAV Maneuver decision - making method based on Markov network. *Journal of System Simulation*, 29(1), 110–116. https://doi.org/10.16182/jiss1004731x.joss.2017S1015

Ma, W., Li, H., Wang, Z., Huang, Z., Wu, Z., & Chen, X. (2020). Close air combat maneuver decision based on deep stochastic game. *Systems Engineering and Electronics*, 43(2), 443–451. https://doi.org/10.12305/j.issn.1001-506X.2021.02.19. http://kns.cnki.net/kcms/detail/11.2422.TN.20200911.1450.006.html

Mei, D., Liu, J. T., & Gao, L. (2017). Maneuver decision of air combat based on approximate dynamic programming and zero-sum game. *Ordnance Industry Automation*, 36(3), 35–39. https://doi.org/10.7690/bgzh.2017.03.010

Mylvaganam, T., Sassano, M., & Astolfi, A. (2017). A differential game approach to multi-agent collision avoidance. *IEEE Trans on Automatic Control*, 62(8), 4229–4235. https://doi.org/10.1109/TAC.2017.2680602

Shin, H., Lee, J., Kim, H., & Hyunchul Shim, D. (2018). An autonomous aerial combat framework for two-on-two engagements based on basic fighter maneuvers. *Aerospace Science and Technology*, 72(3), 305–315. https://doi.org/10.1016/j.ast.2017.11.014

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., & Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676), 354–359. https://doi.org/10.1038/nature24270

Wang, X., Wang, W. J., Song, K. P., & Wang, M. W. (2019). UAV air combat decision based on evolutionary expert system tree. *Ordnance Industry Automation*, 38(1), 48–53. https://doi.org/10.7690/bgzh.2019.01.010

Xu, Z. F., Xu, A., Kou, Y. X., Zhan-Wu, L., & Ai-Wu, Y. (2020). Decision process of multi-aircraft cooperative air combat maneuver. *Systems Engineering and Electronics*, 42(2), 381–389. https://doi.org/10.3969/j.issn.1001-506X.2020.02.1

Xu, A., Gao, C. Q., Kou, Y. X., et al. (2020). Autonomous air combat decision of 1v1 based on BFM method. *Systems Engineering and Electronics*, 43(11), 2513–2519. https://doi.org/10.3969/j.issn.1001-506X.2020.11.13

Zhang, B. C., Kou, Y. N., Wu, M., & Zuo J. (2017). Close-range air combat situation assessment using deep belief network. *Journal of Beijing University of Aeronautics and Astronautics*, 43(7), 1450–1459. https://doi.org/10.13700/j.bh.1001-5965.2016.0956

Zhang, H., Huang, C., Xuan, Y., & Tang, S. (2020). Maneuver decision of autonomous air combat of unmanned combat aerial vehicle based on deep neural network. *Acta Armamentarii*, 41(8), 1613–1622. https://doi.org/10.3969/j.issn.1000-1093.2020.08.01

Zhang, X., Liu, G., Yang, C., & Wu, J. (2018). Research on air confrontation maneuver decision-making method based on reinforcement learning. *Electronics*, 7(11), 279–289. https://doi.org/10.3390/electronics7110279