Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 6481635, 14 pages
https://doi.org/10.1155/2018/6481635

Research Article

Improved Ant Colony Optimization for Weapon-Target Assignment

Xinwu Hu,[1] **Pengcheng Luo**,[2] **Xiaonan Zhang**,[1] **and Jun Wang**[1]

[1] *College of Systems Engineering, National University of Defense Technology, Changsha, 410073, China*
[2] *AllSim Technology Inc., Changsha, 410013, China*

Correspondence should be addressed to Xinwu Hu; huxinwu12@nudt.edu.cn

Received 24 May 2018; Revised 8 September 2018; Accepted 26 September 2018; Published 11 October 2018

Academic Editor: Mengqi Hu

Weapon-target assignment (WTA) which is crucial in cooperative air combat explores assigning weapons to targets with the objective of minimizing the threats from those targets. Based on threat functions, there are four WTA models constrained by the payload and other tactical requirements established. The improvements of ant colony optimization are integrated with respect to the rules of path selection, pheromone update, and pheromone concentration interval, and algorithm AScomp is proposed based on the elite strategy of ant colony optimization (ASrank). We add garbage ants to ASrank; when the pheromone is updated, the elite ants are rewarded and the garbage ants are punished. A WTA algorithm is designed based on the improved ant colony optimization (WIACO). For the purpose of demonstration of WIACO in air combat, a real-time WTA simulation algorithm (RWSA) is proposed to provide the results of average damage, damage rate, and kill ratio. The following conclusions are drawn: (¡?cmd?¿1) the third WTA model, considering the threats of both sides and hit probabilities, is the most effective among the four; (¡?cmd?¿2) compared to the traditional ant colony algorithm, the WIACO requires fewer iterations and avoids local optima more effectively; and (¡?cmd?¿3) WTA is better conducted when any fighter is shot down or any fighter's missiles run out than along with the flight.

1. Introduction

Weapon-target assignment (WTA) is a dynamic multivariable and multiconstraint problem, which is characterized by antagonism, initiative, and uncertainty. So far, there are bulks of studies on the solution to WTA, such as the use of genetic algorithm (GA), simulated annealing (SA), and particle swarm optimization (PSO) algorithm. Additionally, many scholars use ant colony optimization.

In [1, 2], GA was used to solve the problem. In GA, a population of individuals, which encode the problem solutions, are manipulated according to their fitness values through genetic operators, such as reproduction, mutation, and crossover. GA has delightful global searching ability and can find all feasible solutions. However, its local searching ability is poor; to be exact, it is prone to premature convergence. Moreover, it is time consuming as well. Reference [3, 4] used SA in WTA. SA starts from a high initial temperature, and as the temperature falls down, the global optimal solution is found randomly; even when the searching falls into a local optimal solution, SA has a probability to jump out and eventually goes to the global optimum. But this algorithm converges slowly and takes time. PSO was applied to WTA in [5, 6]. PSO builds a swarm intelligence model, initializes a set of random solutions, and searches for the optimal solution by iterations. PSO is simple to implement and converges fast, but it is easy to fall into local optima.

In addition, reference [7, 8] developed a novel multi-objective optimization method based on the evolutionary game theory in real-time WTA. Darryl et al. [9] proved that the dynamic programming method could also solve WTA problem. Liang et al. [10] presented an objective optimization approach based on clonal selection algorithm to solve the problems of WTA in warship formation antiaircraft application. Based on the auction algorithm, Fei et al. [11] brought forward a new distributed multiaircraft cooperative

fire assignment method. Considering the complexity and strict time constraints, Sahin et al. [12] proposed a fuzzy decision method to aid commanders in making decisions for WTA.

As is known, ant colony optimization has the characteristics of distributed computing, self-organization, and positive feedback. The complicated WTA process in air combat can be mapped to ant foraging behavior.

The ant system was first proposed by Dorigo [13] in 1991 in his doctoral thesis. In 1994, Lumer and Faita [14] took the idea of ant colony clustering to data analysis and proposed the LF algorithm. Considering the good performance of ant colony optimization in solving discrete combinatorial optimization problems, Lee et al. [15] first applied it to WTA in 2002. The basic ant colony algorithm was applied to the target assignment problem of the air defense C^3I systems by Huang et al. [16] in 2005. In recent years, ant colony optimization was widely used. For example, in 2013, Olmo et al. [17] used it in the research on association task of data mining, and the results obtained were very exciting. In 2015, Ariyasingha et al. [18] analyzed the performance of multiobjective ant colony optimization for the traveling salesman problem and concluded that the algorithm performed better in problems with more than two objectives and its performance depended slightly on the number of objectives, iterations, and ants. Lately, in 2017, Li et al. [19] designed a biobjective WTA optimization model which maximized the expected damage of the enemy and minimized the cost of missiles; a modified Pareto ant colony optimization algorithm was used in the solution, which produced better results than two multiobjective optimization algorithms NSGA-II and SPEA-II.

Generally, WTA in air combat aims to minimize the threat from opponent fighters, in other words, to maximize the defused threat, constrained by the payload and other tactical requirements. This paper focuses on the WTA modeling, solution, and simulation in air combat scenario. Based on threat functions, four WTA models are established. Among them, Model 3 is proposed for the first time considering the threats of both sides and hit probabilities. For the solution to the WTA models, a WTA algorithm based on improved ant colony optimization (WIACO) is designed, integrating the improvements of traditional ant colony optimization with respect to the rules of path selection, pheromone update, and pheromone concentration interval, and proposes algorithm AS_{comp}. Through a comparative experiment, it is concluded that WIACO requires fewer iterations than traditional ant colony algorithm, and it avoids local optima more effectively. Furthermore, in order to demonstrate and exemplify the effectiveness of WIACO in air combat, a real-time WTA simulation algorithm (RWSA) is presented to simulate real-time WTA in air combat, with the results of average damage, damage rates, and kill ratios.

The paper is organized as follows: Section 2 discusses the air combat threat functions and four WTA models. In Section 3, the WIACO algorithm is introduced, and then in Section 4 a comparative analysis is offered. The RWSA algorithm, three experiments, and result analysis are given in Section 5. Finally, the conclusions of this paper are drawn in Section 6.

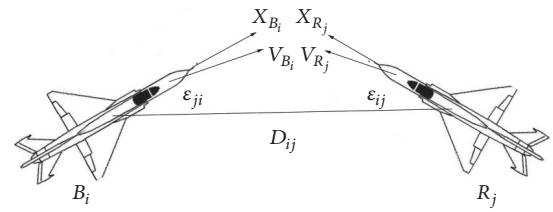

Figure 1: Air combat situation diagram.

2. Weapon-Target Assignment Model

The objective of WTA is to maximize the expected impact on the opponents and to minimize the risk we face [20] in terms of threat. In this case, we need to measure the threat in air combat first.

2.1. Air Combat Threat. It is assumed that the red side has N fighters and the blue side has K fighters and that the red side's early warning aircraft can accurately identify the model, speed, spatial position, and other basic information of the blue fighters. This section uses the air combat situation in reference [21] as shown in Figure 1 for threat modeling.

In Figure 1, R_i is the ith red fighter, B_j is the jth blue fighter, X_{Ri} is the direction of R_i, V_{Ri} is the speed of R_i, ε_{ji} is the off-axis angle of B_j relative to R_i, D_{ij} is the distance between R_i and B_j, and the other parameters are defined similarly.

2.1.1. Angle Threat Function. The angle threat function [22] is given as follows:

$$S_{ij}{}^{1} = \frac{\varepsilon_{ij} - \varepsilon_{ji}}{180^\circ} \tag{1}$$

where $S_{ij}{}^{1}$ is the angle threat of B_j to R_i, with $0^\circ \le \varepsilon_{ji}, \varepsilon_{ij}, \le 180^\circ$ and $-1 \le S_{ij}{}^{1} \le 1$. In particular, when the R_i off-axis angle ε_{ij} is 180° and the B_j off-axis angle ε_{ji} is 0°, that is, when B_j chases R_i from behind, the angle threat of B_j to R_i is 1.

2.1.2. Distance Threat Function. The distance threat function [23] is given as follows:

$$S_{ij}{}^{2} = \begin{cases} 1 & D_{ij} \le T_{a_b} \\ 1 - \dfrac{D_{ij} - T_{a_b}}{L_{r_b} - T_{a_b}} & T_{a_b} < D_{ij} \le L_{r_b} \\ 0 & L_{r_b} < D_{ij} \end{cases} \tag{2}$$

where T_{a_b} represents the missile range of the blue fighter and L_{r_b} is the maximum detection range of the blue radar. When the red fighter is within the blue attack range, the distance threat of B_j to R_i takes the maximum value of 1; when the red fighter is out of the detection range of blue, the distance threat takes the minimum value of 0.

2.1.3. Speed Threat Function. The speed threat function [22] is given as follows:

$$S_{ij}{}^{3} = \begin{cases} 1 & V_{R_i} \le 0.5 \cdot V_{B_j} \\ 1.5 - \dfrac{V_{R_i}}{V_{B_j}} & 0.5 \cdot V_{B_j} < V_{R_i} \le 1.4 \cdot V_{B_j} \\ 0 & 1.4 \cdot V_{B_j} < V_{R_i} \end{cases} \tag{3}$$

where V_{R_i} and V_{B_j} are the speeds of the red and blue fighters, respectively. The greater the V_{R_i} than the V_{B_j}, the smaller the threat of B_j to R_i.

2.1.4. Ability Threat Function. In this paper, we embrace the air combat capability formula in [24] given as follows:

$$C = \left[\ln B + \ln\left(\sum A_1 + 1\right) + \ln\left(\sum A_2\right)\right] \varepsilon_1 \varepsilon_2 \varepsilon_3 \varepsilon_4 \tag{4}$$

where B is the maneuverability parameter of the fighter; A_1 is the fire attack capability parameter; A_2 is the radar detection capability parameter; ε_1 is the pilot's control capability coefficient; ε_2 is the fighter survivability coefficient; ε_3 is the fighter range coefficient; and ε_4 is the Electronic Counter Measures (ECM) capability coefficient. The ability threat function is defined as [22]:

$$S_{ij}{}^{4} = \begin{cases} 1 & \dfrac{C_j}{C_i} \ge 1.5 \\ 0.75 & 1.5 \ge \dfrac{C_j}{C_i} > 1 \\ 0.5 & \dfrac{C_j}{C_i} = 1 \\ 0.25 & 1 > \dfrac{C_j}{C_i} \ge 0.3 \\ 0 & \dfrac{C_j}{C_i} < 0.3 \end{cases} \tag{5}$$

For example, supposing the air combat capability of R_1 is 17.9, if the capabilities of B_1, B_2, and B_3 are 15.8, 18.8, and 17.9, the ability threat degrees of blue to red are 0.25, 0.75, and 0.5, respectively.

As a combination of the above threat functions, the threat degree of B_j to R_i is

$$S_{ij} = \omega_1 S_{ij}{}^{1} + \omega_2 S_{ij}{}^{2} + \omega_3 S_{ij}{}^{3} + \omega_4 S_{ij}{}^{4} \tag{6}$$

where $0 < \omega_i < 1$ $(i = 1,2,3,4)$ is the weight, $\Sigma\omega_i = 1$.

The overall threat degree of B_j to all the red fighters is

$$S_j = \sum_i S_{ij} \tag{7}$$

2.2. Weapon-Target Assignment. Let R_i carry M_i missiles and E_j the number of missiles assigned to B_j:

$$E_j = \begin{cases} 1 & S_j \le 1 \\ 2 & 1 < S_j \le 3 \\ 3 & S_j > 3 \end{cases} \tag{8}$$

Four WTA models are presented in this subsection.

Model 1.

$$\begin{aligned} \max \quad & \sum_i \sum_j x_{ij} S_{ij} \\ \text{s.t} \quad & \sum_j x_{ij} \le M_i, \quad (i = 1, 2, \ldots N) \\ & \sum_i x_{ij} \le E_j, \quad (i = 1, 2, \ldots K) \\ & k = \sum_i \sum_j x_{ij} = \min\left\{\sum_i M_i, \sum_j E_j\right\} \end{aligned} \tag{9}$$

where x_{ij}, an integer from 0 to 3, represents the missile number of R_i assigned to attack B_j; S_{ij} is the threat of B_j to R_i; k is the number of red missiles actually used to attack the target; N is the number of red fighters; and K is the number of blue fighters. The first constraint indicates that the number of missiles launched by each red fighter cannot exceed its carrying capacity; the second constraint implies that the number of missiles assigned to each blue fighter cannot exceed the value obtained in formula (8); the third constraint indicates that the number of fired missiles takes the smaller value between the total number of missiles carried by the red side and the total number of missiles assigned to the blue side.

Model 2.

$$\max \sum_j S_j \left[1 - \prod_i \left(1 - p_{ij}\right)^{x_{ij}}\right] \tag{10}$$

Model 4 was first proposed in reference [25]. The constraints are the same as in Model 1, and p_{ij} is the probability that R_i hits B_j, which is calculated with the two-step adjudication model [26] in this paper. Compared to Model 1, the objective here takes the hit probability of the blue fighters into account; the larger the p_{ij}, the greater the threat.

Model 3.

$$\max \sum_i \sum_j \left[\left(S_{ij} + S_{ji}\right) \times p_{ij} \times x_{ij}\right] \tag{11}$$

The constraints are the same as in Model 1, and S_{ji} is the threat of R_i to B_j. Model 3 considers the blue side threat while also considering the advantages of the red side for the blue side. The objective signifies that red missiles will be preferentially assigned to the blue fighters which pose high threat to red, get high threat from red, and have high hit probability.

Model 4.

$$\max \left[\sum_j S_j \frac{P_j}{\sum_i x_{ij}} + M \sum_j \min\left(0, P_j - P_{dj}\right)\right] \tag{12}$$

$$P_j = 1 - \prod_i \left(1 - p_{ij}\right)^{x_{ij}} \tag{13}$$

Model 4 was first proposed in reference [27], where P_j is the overall hit probability of the red fighter on B_j; P_{dj} is the threshold of hit probability of B_j; and M is the penalty factor. Model 4 calculates the average hit probability for each missile, maximizing the threat of each missile resolution. At the same time, the minimum hit probability requirement is set for each blue fighter, and if P_j is lower than P_{dj}, it will be punished. The objective here is to maximize the average hit probability, considering the blue fighter's hit probability threshold.

3. Improvements of the Ant Colony Optimization for Weapon-Target Assignment

To solve the problem of WTA, some researches tried to make improvements on traditional ant colony optimization with respect to the rules of path selection, pheromone update, and pheromone concentration interval. We consider basic ant colony algorithm (ACO), ant system (AS), elitist-rank ACO algorithm (AS_{rank}), and max-min ant system (MMAS) and improved the elite strategy in AS_{rank}, named AS_{comp}. We add garbage ants to the algorithm of AS_{rank}; when the pheromone is updated, the elite ants are rewarded and the garbage ants are punished, as seen in the following sections (Sections 3.1–3.3). Section 3.4 integrates these improvements in the application of air combat WTA and proposes a WTA algorithm based on the integration.

3.1. Path Selection Rule. (1) In the basic ACO, the path is selected according to the probability P_{ij}, calculated as follows:

$$p_{is} = \begin{cases} \dfrac{[\tau(i,s)]^{\alpha} \cdot [\eta(i,s)]^{\beta}}{\sum_{s \in J_k(i)} [\tau(i,s)]^{\alpha} \cdot [\eta(i,s)]^{\beta}}, & s \in allowed_k \\ 0, & s \notin allowed_k \end{cases} \tag{14}$$

(2) To avoid search stagnation, the path selection in AS [28–30] uses a combination of deterministic and random selections and dynamically adjusts the state transition probability in searching. The specific path selection rules are as follows:

$$j = \begin{cases} \arg\max\limits_{s \in J_k(i)} \left\{ [\tau(i,s)]^{\alpha} \cdot [\eta(i,s)]^{\beta} \right\}, & q \le q_0 \\ J & q > q_0 \end{cases} \tag{15}$$

where i represents the ith red fighter and j, s, J are indices of blue fighters, respectively; J is determined by P_{ij}, the same as the ACO,

where

$\tau(i, s)$ is the pheromone concentration on the path between the current missile i and the assigned position s.

$\eta(i, s)$ is a heuristic function; the greater the threat of the opponent, the greater the probability of launching missiles.

α is the information heuristic factor used to measure the influence of pheromones on the path.

β is the expected heuristic factor used to measure the influence of the threat degree.

$allowed_k$ represents the set of available targets. As the search progresses, $allowed_k$ is getting smaller.

$J_k(i)$ is the set of nodes that the kth ant needs to access after node i has been accessed.

q is a uniform distributed random number in $[0, 1]$, and q_0 is a constant, $0 \le q_0 \le 1$.

p_{is} is the probability of selecting J.

When q is less than or is equal to q_0, the path with the highest pheromone concentration is selected. When q is greater than q_0, the selection probability of each node is obtained by formula (14).

3.2. Pheromone Update Rule. (1) In the ASO, the pheromone is updated after all ants have iterated, and the concentration of all passing paths is updated, which is called partial update:

$$\tau(i,j) = (1-\rho) \cdot \tau(i,j) + \rho \cdot \sum_{k=1}^{m} \Delta\tau(i,j)^k \tag{16}$$

$$\Delta\tau(i,j)^k = \begin{cases} \dfrac{Q}{L_k} & \text{if ant k passes path } (i,j) \\ 0 & \text{others} \end{cases} \tag{17}$$

where $\triangle\tau(i, j)$ is the pheromone concentration increment on the optimal path (i, j); ρ is pheromone volatilization rate, $\rho \in (0,1)$; Q is a constant used to regulate the pheromone concentration; and L_k is the path length of the ant k.

(2) In AS, both partial update and global update are performed; global update is the same as formula (16). Only the pheromone on the optimal path is updated in the global update [31, 32] to enhance the effect of positive feedback. The update rules are as follows:

$$\tau(i,j) = (1-\rho) \cdot \tau(i,j) + \rho \cdot \Delta\tau(i,j)^* \tag{18}$$

$$\Delta\tau(i,j) * = \begin{cases} \dfrac{1}{L_{gb}} & \text{if } (i,j)^* \text{ is the optimal path} \\ 0 & \text{others} \end{cases} \tag{19}$$

where $\triangle\tau(i, j)^*$ is the pheromone concentration increment on the optimal path (i, j) and L_{gb} is the shortest path length of the current cycle.

(3) Traditional algorithms may lead to the elimination of the most adapted ants, and AS_{rank} is proposed to preserve them [33]. AS_{rank} can find better solutions and find these solutions for a shorter period. Set σ elite ants; the global update rules for AS_{rank} are as follows:

$$\tau(i,j) = (1-\rho) \cdot \tau(i,j) + \rho \cdot \sum_{l=1}^{\sigma} (\sigma - l + 1) \cdot \Delta\tau(i,j)^k \tag{20}$$

$$\Delta\tau(i,j)^k = \begin{cases} \dfrac{1}{L_{gb} + L_k} & \text{if ant k is elite ant} \\ 0 & \text{others} \end{cases} \tag{21}$$

(4) Next, the AS_{rank} is improved to make it easier to find the optimal path. Garbage ants are added to the algorithm AS_{rank}. When the pheromone is updated, the elite ants are rewarded while punishing the garbage ants. To prevent the

penalty from causing the pheromone concentration to be too low, the penalty factor ω is set. The improved AS formula is as follows (named AS_{comp}):

$$\tau(i,j) = (1-\rho)\cdot\tau(i,j) + \rho\cdot\sum_{l=1}^{\sigma}(\sigma - l + 1) \cdot \Delta\tau(i,j)^k - \sum_{l=1}^{\gamma}(\gamma - l + 1)\cdot\Delta\tau(i,j)^s \tag{22}$$

$$\Delta\tau(i,j)^s = \begin{cases} \dfrac{\omega}{L_{gb} - L_s} & \text{if ant s is garbage ant} \\ 0 & \text{others} \end{cases} \tag{23}$$

where $\triangle\tau(i,j)^k$ is the same as formula (21) and γ is the number of ant garbage ants.

3.3. Pheromone Concentration Interval Rule. (1) In MMAS, to avoid local stagnation in searching, the pheromone concentration interval is set as follows [34]:

$$\tau_{\max} = \frac{1}{\lambda_1(1-\rho)}\cdot L_{gb} \tag{24}$$

$$\tau_{\min} = \frac{\tau_{\max}(t)}{\lambda_2} \tag{25}$$

where λ_1, λ_2 are two constants used to regulate the pheromone concentration.

(2) Another way to adjust pheromone is to smooth the concentration. By increasing the probability of selecting low pheromone paths, the ability to explore new solutions can be improved:

$$\tau^*(i,j) = \tau(i,j) + \delta(\tau_{\max} - \tau(i,j)) \tag{26}$$

where $\tau^*(i,j)$ is the amount of pheromone after smoothing; δ is a constant used to regulate the concentration; and τ_{max} is the same as formula (24).

3.4. Algorithm Performance Comparison. Considering the rules of path selection, pheromone update, and pheromone concentration interval, 24 sets of algorithms are obtained, as shown in Table 1. These algorithms are brought into the WTA problem for performance comparison analysis. The average optimal solution and convergence of the 100 trials are counted.

Algorithm 1 selects P_{ij} of basic ACO, partial update of ACO, and none. Algorithm 2 selects P_{ij} of basic ACO, partial update of ACO, and MMAS's interval, and other algorithms are selected according to the same method. The test results are as shown in Table 2.

As can be seen from Table 2, the average optimal solution of algorithm 24 is the largest, which is 20.15. So, it can resolve the largest threat, and the average convergence is 7.13. Algorithm 11 has the worst effect, and the optimal solution is only 13.17, which is the easiest to fall the local optimal solution.

Algorithm 24 selects rules of random selections of AS, punishes the rule of AS_{comp}, smooths the concentration, and achieves the best results. Algorithm 24 is used as the research algorithm of this paper.

3.5. WIACO Algorithm. When the ant colony optimization is used to solve the WTA problem, the assignment process needs to be modeled with an ant colony network. For example, in Figure 2, each red missile is represented by a small node and each red and blue fighter is represented by a big node; the two sides both dispatch two fighters; red fighter 1 carries four missiles and red fighter 2 carries three; blue fighter 1 is assigned two missiles and blue fighter 2 is assigned one missile.

The ants follow the path from the red nodes to the blue nodes, and then, according to the same strategy, take a virtual path back to the red nodes until the assignment is completed.

The number of ants in the population is set to [35]:

$$m = N_{rm} + \sum E_j \tag{27}$$

where N_{rm} is the total number of missiles carried by red fighters and E_j is the number of missiles assigned to B_j. In the beginning of the iteration, m ants are placed randomly on the red missiles, and the initial pheromone concentration on each path is set to 1.

The ants move according to the following rules:

Rule 1. An ant can only move to a blue fighter whose missile assignment is insufficient; red fighters launch the remaining number of missiles at most.

Rule 2. Each ant can only reach one node at a time; that is, each missile can only attack a single target.

Rule 3. The ants do not interfere with each other. Ants return to red fighter nodes with the same pseudorandom probability, and the targets are the red fighters who still have missiles.

Rule 4. All ants need to update the pheromone by the end of a cycle and generate new pheromones only on the optimal path; pheromone is partially volatile.

Pseudocode of the WIACO algorithm is shown as in algorithm 1.

4. Examples of Comparative Analysis

4.1. Model Parameter Analysis. The parameters of the WTA model called by WIACO algorithm are discussed with a control variable experiment in this subsection, including the α, β, ρ, Q, and q_0. Set δ to 0.001 and set ω to 0.1. The number of iterations is 200, and the means of the results are taken.

Taking Model 3 in Section 2.2 for example, the parameter variation curves of Model 3 are shown in Figure 3. The horizontal axis represents the parameter; the vertical axis represents the maximum defused threat. As we can see in Figure 3, when $\alpha = 1.5$, $\beta = 3$, $\rho = 0.7$, $Q = 0.4$, and $q_0 = 0.7$, the curves reach their respective maximum defused threat. Therefore, Model 3 takes the final parameters as $\alpha = 1.5$, $\beta = 3$, $\rho = 0.7$, $Q = 0.4$, and $q_0 = 0.7$. Similarly, the final parameters of the other three models are set as in Table 3.

The obtained parameters are substituted into the models for the experimental analysis, and the results of one of these experiments are as in Figure 4. The horizontal axis represents

Table 1: Algorithm set.

Path Selection Rule	Pheromone Update Rule	Pheromone Interval
P_{ij} of basic ACO	Partial update of ACO	None
Random selections of AS	Partial and global update of AS	MMAS's interval
	Elite ant strategy of AS_{Rank}	Smooth the concentration
	Punish rule of AS_{comp}	

Table 2: Test results of Algorithm set.

Algorithm	**1**	**2**	**3**	**4**	**5**	**6**
convergence	21.36	19.60	19.03	14.43	4.43	16
optimal	15.27	15.45	15.43	19.77	15.96	20.05
Algorithm	**7**	**8**	**9**	**10**	**11**	**12**
convergence	23	4.30	18.53	12.90	3.03	12.56
optimal	19.68	14.35	19.983	20.10	13.17	20.03
Algorithm	**13**	**14**	**15**	**16**	**17**	**18**
convergence	5.66	6.10	6.23	17.36	16.86	15.76
optimal	14.23	14.22	14.21	18.12	19.52	17.98
Algorithm	**19**	**20**	**21**	**22**	**23**	**24**
convergence	23.10	15	22.33	16.23	9.76	7.13
optimal	18.61	18.40	18.49	19.75	17.99	20.15

Table 3: Model parameter values.

Parameter	Model 1	Model 2	Model 3	Model 4
α	1.5	1.5	1.5	1.5
β	3	2	3	2
ρ_1	0.4	0.7	0.7	0.6
ρ_2	0.7	0.5	0.4	0.6
q_0	0.6	0.7	0.7	0.5

the iterations; the vertical axis represents the maximum defused threat.

Repeat the experiment 100 times and take the average for analysis. The results are presented in Table 4. As seen in Table 4, the variances of the four models are not large and the iterations converge at the 20th iteration, earlier or later, and that is acceptable.

4.2. Comparison with Traditional Algorithm. It is assumed that the red side adopts WIACO algorithm with WTA Model 3.

According to Table 5 and the air combat capability in formula (4), the air combat capability of each red fighter is $C_r = 20.79$ and that of each blue fighter is $C_b = 21.97$.

Based on reference [25], assuming the fighter performance has the greatest impact on air combat, we let $\omega_1 = 0.2$, $\omega_2 = 0.2$, $\omega_3 = 0.2$, and $\omega_4 = 0.4$ and take the value of λ_1=3, λ_2=40.

The traditional ant colony optimization and the WIACO proposed in this paper are both simulated 100 times, and the best convergence results of the two algorithms are obtained.

The convergence of the traditional algorithm is shown in Figure 5(a); thereinto, the results fluctuate. After the algorithm goes through the maximum defused threat degree of 26.06, it falls into the local optimum and gets a stable defused threat degree of 25.9. In light of this, the traditional algorithm cannot jump out of local optimum, and it does not produce the global optimal WTA solution.

The WIACO convergence is shown in Figure 5(b). Figure 5(b) shows that at the 3rd iteration the improved algorithm finds a locally optimal assignment with the defused threat degree of 26.19; then it jumps out of the local optimum quickly and converges to a stable optimal assignment with a maximum defused threat degree of 26.29 at the 12th iteration. Table 6 shows the WTA solution in detail. For example, as we can see on the first row of Table 6, the red fighter 1 launches two missiles to attack the blue fighter 1, one missile to attack blue fighter 2, and one missile to attack the blue fighter 6, and the missiles all run out; on the seventh row, none of the red fighter 7's missiles are assigned.

The above analysis indicates the advantages of the WIACO algorithm, which can provide better solution than traditional algorithm for the WTA. Comparatively speaking, it can be considered that the traditional algorithm takes longer time to convergence, and it is harder to jump out of local optimum.

```
Call the threat model;
Initialize parameters:
While NC - 1<NC_max (The largest iteration number)
  For i=1:m (The number of ants)
      Put m ants on red missiles randomly;
  End
  For k=1:count (The total number of ants' movements)
      For j=1:m
          For i=1:K_b (Number of attackable blue fighters)
              Generate a random number q;
              Select path with pseudorandom probability using formula ((14), (15));
              Add the target to selected set;
              Assignable target number minus 1;
              The number of remaining missiles in red fighters minus 1;
              allowed_k minus the assigned target;
          End
      End
      For j=1:m
          For i=1:N_rl (The remaining amount of red missiles)
              Generate a random number q;
              Select path with pseudorandom probability using formula ((14), (15));
              Add the target to selected set;
              allowed_k minus the assigned target;
          End
      End
      Determine the path of the maximum threat in this cycle using WTA model;
      Global update of pheromone using formula ((22), (23));
      Update the pheromone interval on the paths using formula (26);
      NC=NC+1;
  End
End
Get the optimal WTA solution.
```

Algorithm 1: WIACO (WTA algorithm based on improved ant colony optimization).

Table 4: Statistical analysis of the model results.

	Max	Min	Mean	Var
Model 1	17.2896	17.2831	17.2859	0.0021
Model 2	12.2186	11.8368	12.1267	0.0109
Model 3	22.1207	18.6858	20.4573	0.5852
Model 4	6.08764	5.8862	6.0533	0.0024

Table 5: Red and blue parameter sets.

Parameter	Red	Blue
Number of fighters	8	8
Number of missiles	4	4
Flight speed *(m/s)*	340	340
Missile range *(km)*	80	90
Radar detection distance *(km)*	120	120
Maneuvering parameter	25.5	25.65
Fire parameter	2761	2832
Detection capability parameter	1514	1553
Manipulation efficiency factor	0.9	0.95
Viability coefficient	0.915	0.995
Voyage coefficient	1.14	1.14
Electronic countermeasure coefficient	1.05	1.1

```
For time=1:T
    Call the WIACO algorithm;
    For i=1:N_r
        For j=1:N_b
            If D_ij<T_i (D_ij is the distance between two sides, T_i is the range of the red fighters)
                Call the two-step adjudication model to get the probability of damage P^r, P^b;
            End
        End
    End

    For k_r=1:N_r
        Get random number P^r_k;
        If P^r_k<P^r
            Red fighter k_r is shot down;
        End
    End
    For k_b=1:N_b
        Get random number P^b_k;
        If P^b_k<P^b
            Blue fighter k_b is shot down;
        End
    End
    Update the air combat situation;
    If one termination condition is met, end the loop;
    time=time+1;
End
Output simulation results.
```

Algorithm 2: RWSA (real-time WTA simulation algorithm).

Table 6: Improved algorithm assignment.

Red fighter	Assignment (missiles)	Remaining missiles
Red1	Blue1 (2); Blue2 (1); Blue6 (1)	0
Red2	Blue3 (3); Blue6 (1)	0
Red3	Blue5 (2); Blue4 (1)	1
Red4	Blue5 (1); Blue4 (2)	1
Red5	Blue4 (1); Blue7(2)	1
Red6	Blue4 (1)	3
Red7	—	4
Red8	Blue6 (1); Blue8 (3)	0

5. Simulation Analysis of WTA in Air Combat

During air combat, the combat situation is constantly changing, and the WTA needs to be continuously updated. This section simulates the complete air combat process, identifies the final effectiveness, and makes relevant experimental analysis.

5.1. Simulation Strategy. Assuming the red side performs WTA using the WIACO algorithm, the air combat simulation in this section is based on the following rules:

Rule 1. Due to the early warning aircraft, air combat situation is accessed in real time. During the simulation, both sides constantly approach each other. To reflect the randomness and flexibility of the formation and the air combat situation, the fighters move with a random and constant speed under their respective limits. Assuming that both sides are constantly adjusting their direction of flight, the off-axis angles of two side fighters are decreasing [36].

Rule 2. Once the blue fighters move into range, the red fighters launch their missiles.

Rule 3. The data are put into the two-step adjudication model [26] to calculate the effectiveness.

Rule 4. The termination conditions of the simulation are as follows: (1) one side runs out of missiles; (2) all fighters on either side are shot down; and (3) the air combat move beyond the horizon (within 20 km).

The simulation steps are as follows:

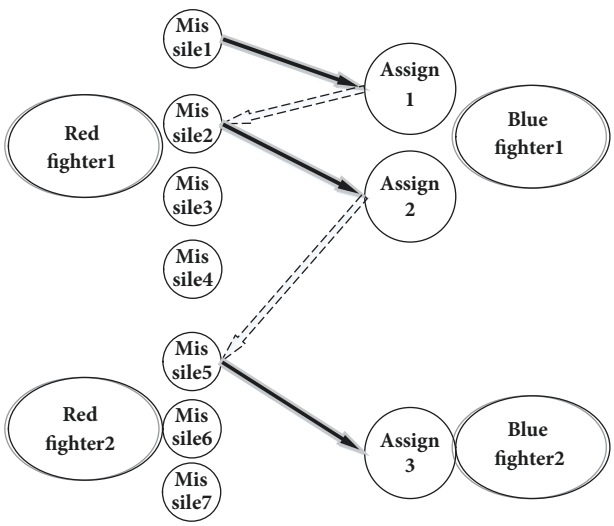

Figure 2: Ant colony network of air combat.

Step 1. Obtain the initial data of the red and blue sides and initialize the parameters.

Step 2. Call the WIACO algorithm.

Step 3. Determine whether or not the blue fighters are within range of red side. Once they are within range, launch the red missiles and call the two-step adjudication model to assess the effectiveness.

Step 4. Update the air combat situation.

Step 5. Repeat Steps 2–4 and output the final effectiveness when one of the termination conditions is reached.

In the simulations, the number of red fighters is N_r and that of blue fighters is N_b, and the number of simulations is T. The pseudocode of the RWSA algorithm is given as shown in Algorithm 2.

5.2. Experiment 1. In experiment 1, both sides dispatch 8 fighters. The parameters used in the two-step adjudication model are shown in Table 7.

Because of the randomness of fighter kill and air combat situation, each simulation result is different, which reflects the uncertainty of air combat. Table 8 lists the results of one simulation as an example. As seen in Table 8, by the end of the simulation, 4 red fighters (3,6,7, and 8) are shot down, 4 red fighters (1,2,4, and 5) run out of missiles, 3 blue fighters (1,3, and 7) are shot down, and 2 blue fighters (2 and 4) run out of missiles.

In this paper, we run a large number of simulations and statistical results with formula ((28)-(30)). Denote the amount of damage of red side in the ith simulation with L_i^r and that of blue side with L_i^b, and the damage rates of red and blue sides, P^r and P^b, are calculated with formula ((28), (29)):

$$P^r = \frac{\sum_{i=1}^{T} L_i^r}{T \times N_r} \tag{28}$$

$$P^b = \frac{\sum_{i=1}^{T} L_i^b}{T \times N_b} \tag{29}$$

The kill ratio W is calculated with formula (23):

$$W = \frac{\sum_{i=1}^{T} L_i^r}{\sum_{i=1}^{T} L_i^b} \tag{30}$$

The smaller the kill ratio is, the greater the advantage of the red side is in air combat.

First, we run 1000 times simulations using Model 1, and the relationship between the kill ratio and the number of iterations is shown in Figure 6. Figure 6 shows that, as the number of iterations increases, the kill ratio tends to stabilize. The results of the air combat are shown in Figure 7 and Table 9. From Figure 7 we can see that 4 to 6 red fighters are killed mostly and 2 to 4 blue fighters are killed in most cases. The kill ratio in Table 9 is 1.6921; that is, the blue side is ascendant.

And then additional experiments are conducted using Models 2, 3, and 4 ($M = 100$, $P_{dj} = 0.2$), respectively. Among the results of the four simulations seen in Tables 9–12, the kill ratio of Model 3 is the smallest, so Model 3 gets the best effectiveness for WTA in air combat.

5.3. Experiment 2. In the air combat simulations, there are two possible options of WTA timing. The first is to reassign at each time step of the simulation so that the WTA can be adjusted according to the real-time air combat situation. However, this increases the requirements of the pilots' target locating capability. The second option is to reassign when any fighter is shot down or any fighter's missiles are used up. This option allows the pilot to focus on the attacks on located targets but reduces the ability to adapt to the battlefield.

Experiment 2 is carried out using the above two options, where parameters are the same as in experiment 1. The results are shown in Tables 13 and 14. As seen in Tables 13 and 14, there is not much difference in the kill ratio between option 1 and option 2, but the red damage rate in option 2 is lower. Therefore, option 2 is more suitable for WTA regarding survivability and economics.

5.4. Experiment 3. Given the number of blue fighters, experiment 3 studies how many red fighters should be dispatched. The number of blue fighters is fixed at 8, while the number of red fighters increases from 3 to 18. The parameters are the same as in experiment 1. A total of 1000 simulations are performed to calculate the kill ratio, and the results are shown in Table 15 and Figure 8.

In Table 15 and Figure 8, as the number of red fighters' increases, the kill ratio declines; however, the downward trend weakens after 12. This implies that the red side achieves a satisfying combat effectiveness if it dispatches 12 fighters to cope with 8 blue fighters at a kill ratio of about 1.5159.

6. Conclusions and Future Directions

In the air combat, the battlefield situation is complex and changeable, and WTA plays a decisive role. In Section 2

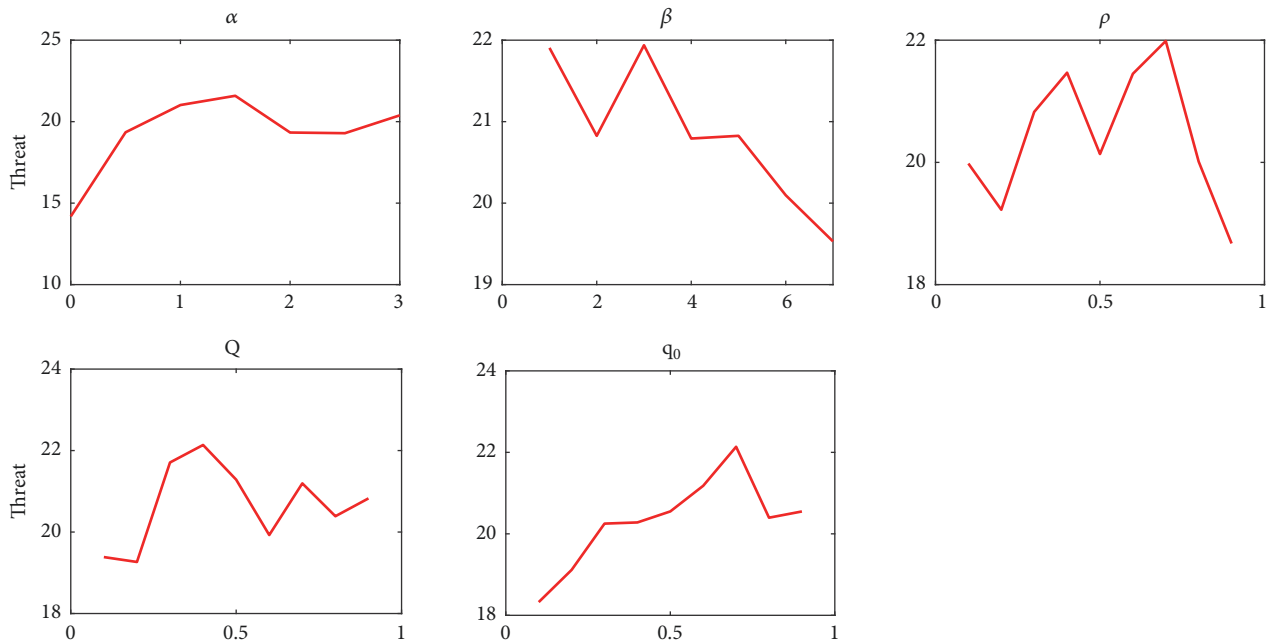

Figure 3: Parameter variation curves of Model 3.

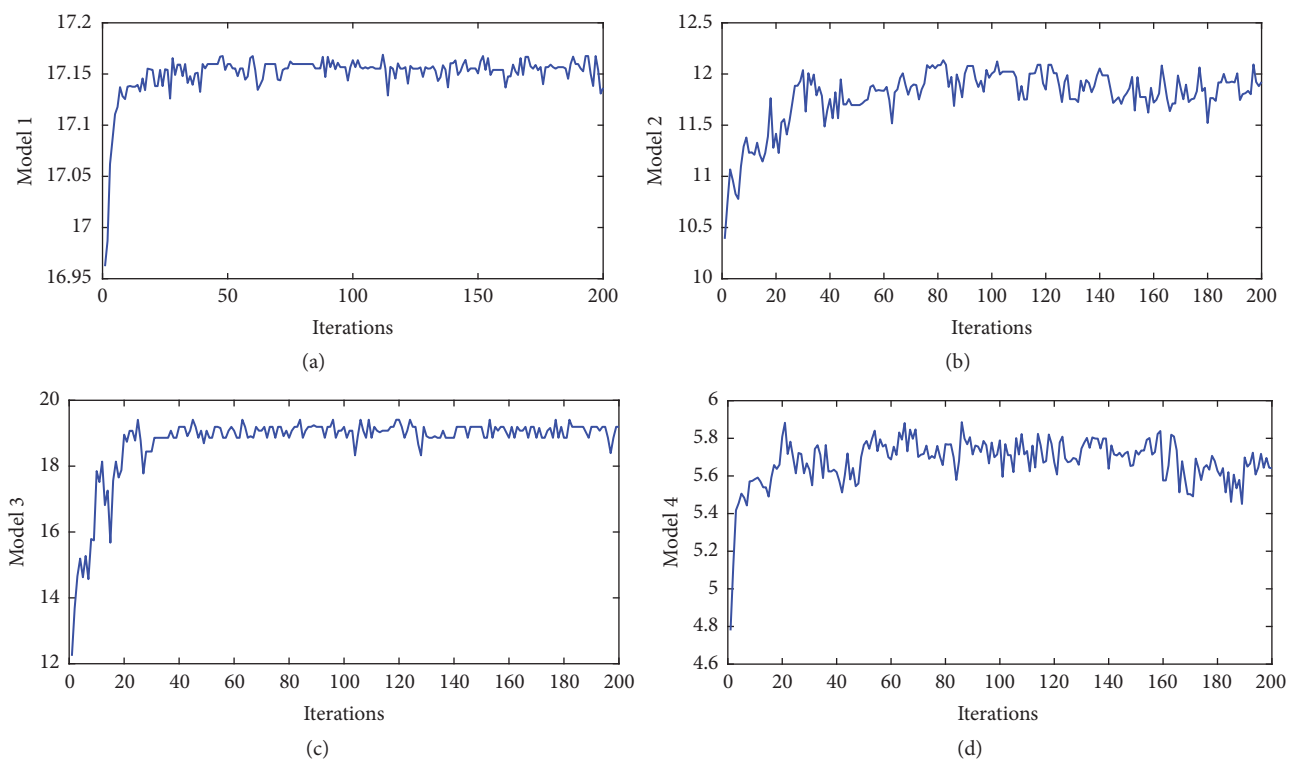

Figure 4: Model convergence curves.

of this paper, four threat functions are used to evaluate the threat. Meanwhile, four WTA models are set up with different objectives. Among them, Model 3 is proposed for the first time considering the threats of both sides and hit probabilities.

In order to solve the WTA problem, WIACO algorithm is presented in Section 3 with improvements of the traditional ant colony optimization in the aspects of path selection rule, pheromone updating rule, and pheromone concentration interval rule. The comparative experiment in Section 4 shows

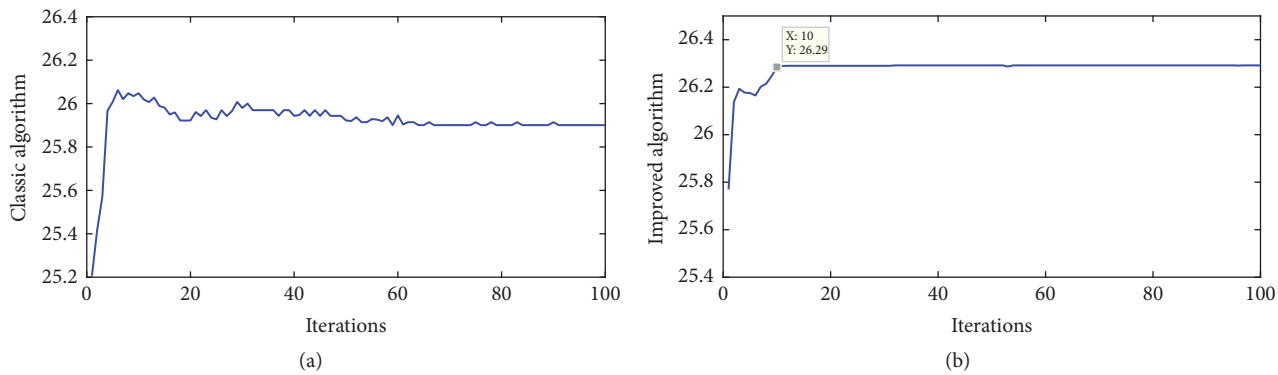

Figure 5: Traditional and improved algorithm convergence curves.

Table 7: Two-step adjudication model parameters.

Parameter	Red	Blue
Command factor	0.6	0.7
Fighter vulnerability	0.45	0.4
Multitarget attack capability	4	4
Sorting ratios	1	1
Allow war to match	0.8	0.8
Fighter radar reflection cross section (m^2)	3.2	2
Probability of target discovery	0.7	0.7
Missile score	0.8	0.8
Airborne radar maximum discovery distance (km)	120	120
Airborne missile maximum effective range (km)	80	90

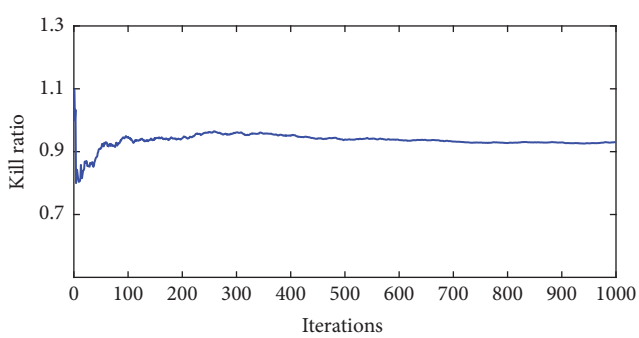

Figure 6: Kill ratio and iterations curve.

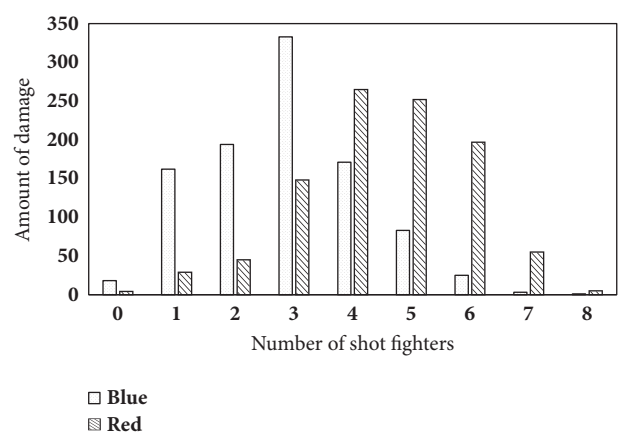

Figure 7: Damage distribution.

that WIACO algorithm which provides the optimal solution for WTA has the advantages of faster convergence and better avoidance from local optima.

For the sake of the demonstration and exemplification of WIACO in air combat, four combat simulation rules and RWSA algorithm are designed in Section 5. Based on this, this section carries out three simulation experiments. Through experiment 1, we can find that Model 3 gets the minimal kill ratio indicating the largest advantage of red side. Hence, we can conclude that Model 3 gets the best effectiveness for WTA in air combat. In experiment 2, we analyze different WTA timings with results showing that WTA is better conducted when air combat situation changes (i.e., any fighter is shot down or any fighter's missiles are used up) than along with the flight. Finally, experiment 3 shows that, when the blue dispatches 8 fighters, 12 red fighters shall be dispatched accordingly. When the number of red fighters exceeds 12, the decrease in kill ratio is not obvious, if it is not increasing.

In general, from the advantages exemplified by the simulation experiments, it can be concluded that the improved ant colony optimization proposed in this paper can be applied to WTA in air combat.

As future work, we intend to apply other intelligent algorithms to the WTA problem and compare it with the

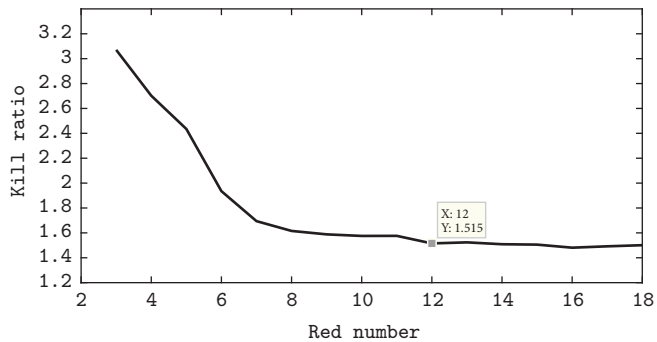

Figure 8: Red fighter number and kill ratio curve.

Table 8: Simulation results (example).

	Red	Blue
clock 6	Fighters 6 and 8 shot down; fighters 4 and 5 out of missiles	Fighters 1 and 7 shot down; Fighter 4 out of missiles
clock 8	Fighter 3 shot down; fighter 2 out of missiles	Fighter 3 shot down
clock 15	Fighter 7 shot down; 1 out of missiles	Fighter 2 out of missiles
total	Shot down: 4; Missiles exhausted: 4	Shot down: 3, Missiles exhausted: 2

Table 9: Model 1 simulation results.

	Red	Blue
Average damage	7.3945	4.1864
Damage rate	0.9243	0.5233
Kill ratio	1.7663	

improved ant colony algorithm to further explore the best solution for WTA. At the same time, the air combat simulation process needs to be refined. The current simulation hypothesis is relatively simple. The next step is to make the simulation process closer to actual combat and to make the simulation results more practical.

Notations

N: Number of red fighters
K: Number of blue fighters
R_i: The ith red fighter
B_j: The jth blue fighter
X_{Ri}: The direction of R_i
V_{Ri}: The speed of R_i
ε_{ji}: The off-axis angle of B_j relative to R_i
D_{ij}: The distance between R_i and B_j
S_{ij}: The threat of B_j to R_i
T_{ab}: The missile ranges of the blue fighter
L_{rb}: The maximum detection ranges of the blue radar
V_{Ri}: The speeds of the red fighter
V_{Bj}: The speeds of the blue fighter
A_1: The fire attack capability parameter
A_2: The radar detection capability parameter
B: The maneuverability parameter of the fighter
C: The air combat capability
ε_1: The pilot's control capability coefficient
ε_2: The fighter survivability coefficient
ε_3: The fighter range coefficient
ε_4: The Electronic Counter Measures capability coefficient
M_i: The number of missiles of R_i
E_j: The number of missiles assigned to B_j
x_{ij}: The missile number of R_i assigned to attack B_j
k: The number of red missiles actually used to attack
p_{ij}: The probability that R_i hits B_j
P_j: The overall hit probability of the red fighter on B_j
P_{dj}: The threshold of hit probability of B_j
P_{ij}: The path selection probability
$\tau(i, s)$: The pheromone concentration
$\eta(i, s)$: The heuristic function
α: The information heuristic factor
β: The expected heuristic factor
$allowed_k$: The set of available targets
$J_k(i)$: The set of nodes that the kth ant needs to access
q: A random number in $[0, 1]$
q_0: A constant, $0 \le q_0 \le 1$
ρ: The pheromone volatilization rate
Q: A constant used to regulate the pheromone concentration
L_k: The path length of the ant k
L_{gb}: The shortest path length of the current cycle
γ: The number of ant garbage ants
λ_i: A constant used to regulate the pheromone concentration

Table 10: Model 2 simulation results.

	Red	Blue
Average damage	7.2821	4.3632
Damage rate	0.9103	0.5454
Kill ratio	1.6690	

Table 11: Model 3 simulation results.

	Red	Blue
Average damage	7.2492	4. 5326
Damage rate	0.9062	0.5665
Kill ratio	1.5993	

Table 12: Model 4 simulation results (M=100, P_{dj}=0.2).

	Red	Blue
Average damage	7.3215	4. 0854
Damage rate	0.9151	0.5106
Kill ratio	1.7921	

Table 13: Option 1 simulation results.

	Red	Blue
Average damage	4.5212	2.8727
Damage rate	0.5652	0.3591
Kill ratio	1.5738	

Table 14: Option 2 simulation results.

	Red	Blue
Average damage	3.9122	2.4820
Damage rate	0.4890	0.3103
Kill ratio	1.5762	

Table 15: Red fighter number and kill ratio.

Number of red fighters	3	4	5	6
Kill ratio	3.0726	2.6845	2.4587	1.9752
Number of red fighters	7	8	9	10
Kill ratio	1.6872	1.6154	1.5886	1.5754
Number of red fighters	11	12	13	14
Kill ratio	1.5743	1.5159	1.5245	1.5088
Number of red fighters	15	16	17	18
Kill ratio	1.5051	1.4912	1.4921	1.5011

δ: A constant used to regulate the concentration
ω: The penalty factor
N_{rm}: The total number of missiles carried by red fighters
L_i^r: The amount of damage of red side in the ith simulation
L_i^b: The amount of damage of blue side in the ith simulation
P^r: The damage rate of red side
P^b: The damage rate of blue side
W: The kill ratio.

Data Availability

The [data.xlsx] data used to support the findings of this study are included within the supplementary information file(s). Link: https://figshare.com/s/433d02b1d101aa94301d

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Z.-J. Lee, S.-F. Su, and C.-Y. Lee, "Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 33, no. 1, pp. 113–121, 2003.

[2] M. A. Sahin and K. Leblebicioglu, "A genetic algorithm for weapon target assignment problem," in *Proceedings of the Summer Computer Simulation Conference Society for Modeling Simulation International*, pp. 61–65, 2009.

[3] Y. Li and Y. Dong, "Weapon-target assignment based on simulated annealing and discrete particle swarm optimization in cooperative air combat," *Acta Aeronautica et Astronautica Sinica*, vol. 31, no. 3, pp. 626–631, 2010.

[4] Z. Ding, D.-W. Ma, M.-D. Tang, and X.-F. Zhang, "TSAPSO: a hybrid search algorithm of tabu search and annealing particle swarm optimization for weapon-target assignment," *Xitong Fangzhen Xuebao / Journal of System Simulation*, vol. 18, no. 9, pp. 2480–2483, 2006.

[5] M. C. Su, "A PSO-based Decision Aid for Multi-Aircraft Combat Situations," in *Proceedings of the International Journal of Fuzzy Systems*, vol. 10, pp. 161–167, 2010.

[6] X. Liu, Z. Liu, W. S. Hou, and J. H. Xu, "Improved MOPSO algorithm for multi-objective programming model of weapon-target assignment," *Systems Engineering & Electronics*, vol. 35, no. 2, pp. 326–330, 2013.

[7] C. Leboucher, H.-S. Shin, S. Le Ménec et al., "Novel evolutionary game based multi-objective optimisation for dynamic weapon target assignment," in *Proceedings of the 19th IFAC World Congress on International Federation of Automatic Control, IFAC 2014*, pp. 3936–3941, South Africa, August 2014.

[8] Z. Yao, M. Li, Z. Chen, and R. Zhou, "Mission decision-making method of multi-aircraft cooperatively attacking multi-target based on game theoretic framework," *Chinese Journal of Aeronautics*, vol. 29, no. 6, pp. 1685–1694, 2016.

[9] D. K. Ahner and C. R. Parson, "Optimal multi-stage allocation of weapons to targets using adaptive dynamic programming," *Optimization Letters*, vol. 9, no. 8, pp. 1689–1701, 2015.

[10] H. Liang and F. Kang, "Adaptive chaos parallel clonal selection algorithm for objective optimization in WTA application," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 6, pp. 3459–3465, 2016.

[11] A. G. Fei, L. Y. Zhang, and Q. J. Ding, "Multi-aircraft cooperative fire assignment based on auction algorithm," *Systems Engineering & Electronics*, vol. 34, no. 9, pp. 1829–1833, 2012.

[12] M. A. Şahin and K. Leblebicioğlu, "Approximating the optimal mapping for weapon target assignment by fuzzy reasoning," *Information Sciences*, vol. 255, pp. 30–44, 2014.

[13] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed Optimization by Ant Colonies," in *Proceedings of the Ecal91 - European Conference on Artificial Life*, 1992.

[14] E. D. Lumer and B. Faiet, "Diversity and adaptation in populations of clustering ants," in *Proceedings of the third international conference on Simulation of adaptive behavior : from animals to animats 3: from animals to animats 3*, pp. 501–508, 1994.

[15] Z.-J. Lee, C.-Y. Lee, and S.-F. Su, "An immunity-based ant colony optimization algorithm for solving weapon-target assignment problem," *Applied Soft Computing*, vol. 2, no. 1, pp. 39–47, 2002.

[16] S. C. Huang and W. M. Li, "Research of Ant Colony Algorithm for Solving Target Assignment Problem," *Journal of Systems Engineering and Electronics*, vol. 27, no. 1, pp. 79-80, 2005.

[17] J. L. Olmo, J. M. Luna, J. R. Romero, and S. Ventura, "Mining association rules with single and multi-objective grammar guided ant programming," *Integrated Computer-Aided Engineering*, vol. 20, no. 3, pp. 217–234, 2013.

[18] I. D. I. D. Ariyasingha and T. G. I. Fernando, "Performance analysis of the multi-objective ant colony optimization algorithms for the traveling salesman problem," *Swarm and Evolutionary Computation*, vol. 23, pp. 11–26, 2015.

[19] Y. Li, Y. Kou, Z. Li, A. Xu, and Y. Chang, "A Modified Pareto Ant Colony Optimization Approach to Solve Biobjective Weapon-Target Assignment Problem," *International Journal of Aerospace Engineering*, vol. 2017, 2017.

[20] L.-M. Zhang, Y. Zhang, and W.-B. Liu, "The design of target assignment model based on the reverse mutation ant colony algorithm," in *Proceedings of the 2012 International Workshop on Information and Electronics Engineering, IWIEE 2012*, pp. 1554–1558, China, March 2012.

[21] M.-C. Su, S.-C. Lai, S.-C. Lin, and L.-F. You, "A new approach to multi-aircraft air combat assignments," *Swarm and Evolutionary Computation*, vol. 6, pp. 39–46, 2012.

[22] S. C. Huang and L. I. Wei-Min, "Ant colony algorithm for solving scheduling problem in multi-targets attacking," *Omr Ngnrng*, 2008.

[23] T. Zhang, L. Yu, X. Wei, and Z. L. Zhou, "Decision-making for cooperative multiple target attack based on adaptive pseudo-parallel genetic algorithm," *Fire Control & Command Control*, vol. 38, no. 5, pp. 137–140, 2013.

[24] B. L. Zhu, R. C. Zhu, and X. F. Xiong, *Effectiveness Evaluation for Combat Aircrafts*, Aviation Industrial Publishing House, Beijing, China, 2006.

[25] H. Cai, J. Liu, Y. Chen, and H. Wang, "Survey of the research on dynamic weapon-target assignment problem," *Journal of Systems Engineering and Electronics*, vol. 17, no. 3, pp. 559–565, 2006.

[26] D. A. Shlapak, D. T. Orletsky, and B. A. Wilson, "Dire strait? Military aspects of the China-Taiwan confrontation and options for US policy," *EMBO Journal*, vol. 17, no. 5, pp. 1228–1235, 2000.

[27] J. Yan, X. Li, L. Liu, and F. Zhang, "Weapon-target assignment based on Memetic optimization algorithm in beyond-visual-rang cooperative air combat," *Beijing Hangkong Hangtian Daxue Xuebao/Journal of Beijing University of Aeronautics and Astronautics*, vol. 40, no. 10, pp. 1424–1429, 2014.

[28] R. J. Kuo, B. S. Wibowo, and F. E. Zulvia, "Application of a fuzzy ant colony system to solve the dynamic vehicle routing problem with uncertain service time," *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 40, no. 23-24, pp. 9990–10001, 2016.

[29] C.-J. Ting and C.-H. Chen, "A multiple ant colony optimization algorithm for the capacitated location routing problem," *International Journal of Production Economics*, vol. 141, no. 1, pp. 34–44, 2013.

[30] J. Zhang, "ACGA Algorithm of Solving Weapon - Target Assignment Problem," *Open Journal of Applied Sciences*, vol. 02, no. 04, pp. 74–77, 2012.

[31] P. Guo and L. Zhu, "Ant colony optimization for continuous domains," in *Proceedings of the 8th International Conference on Natural Computation (ICNC '12)*, pp. 758–762, Chongqing, China, May 2012.

[32] M. Mahi, Ö. K. Baykan, and H. Kodaz, "A new hybrid method based on particle swarm optimization, ant colony optimization and 3-Opt algorithms for traveling salesman problem," *Applied Soft Computing*, vol. 30, pp. 484–490, 2015.

[33] A. C. Zecchin, A. R. Simpson, H. R. Maier, A. Marchi, and J. B. Nixon, "Improved understanding of the searching behavior of ant colony optimization algorithms applied to the water distribution design problem," *Water Resources Research*, vol. 48, no. 9, 2012.

[34] M. M. S. Abdulkader, Y. Gajpal, and T. Y. Elmekkawy, "Hybridized ant colony algorithm for the Multi Compartment Vehicle Routing Problem," *Applied Soft Computing*, vol. 37, pp. 196–203, 2015.

[35] J. Guan and G. Lin, "Hybridizing variable neighborhood search with ant colony optimization for solving the single row facility layout problem," *European Journal of Operational Research*, vol. 248, no. 3, pp. 899–909, 2016.

[36] J. Li, J. Chen, B. Xin, and L. Dou, "Solving multi-objective multi-stage weapon target assignment problem via adaptive NSGAII and adaptive MOEA/D: A Comparison Study," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '15)*, pp. 3132–3139, May 2015.