Research Article

Improved Ant Colony Optimization for Weapon-Target Assignment

Xinwu Hu, Pengcheng Luo, Xiaonan Zhang, and Jun Wang

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

Copyright © 2018 Xinwu Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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 averaged damage, damage rate, and kill ratio. The following conclusions are drawn: (1) the third WTA model, considering the threats of both sides and hit probabilities, is the most effective among the four; (2) compared to the traditional ant colony algorithm, the WIACO requires fewer iterations and avoids local optima more effectively; and (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 multiobjective 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 C²I 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, Arijasingha 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 AScomp. 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 simplify 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.

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 \( i \)th red fighter, \( B_j \) is the \( j \)th blue fighter, \( X_{R_i} \) is the direction of \( R_i \), \( V_{R_i} \) is the speed of \( R_i \), and \( \epsilon_{ij} \) 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:

\[ S_{ij}^1 = \frac{\epsilon_{ij} - \epsilon_{ij}^0}{180^\circ} \]  

where \( S_{ij}^1 \) is the angle threat of \( B_j \) to \( R_i \), with \( 0^\circ \leq \epsilon_{ij} \leq 180^\circ \) and \(-1 \leq S_{ij}^1 \leq 1\). In particular, when the \( R_i \) off-axis angle \( \epsilon_{ij} \) is \( 180^\circ \) and the \( B_j \) off-axis angle \( \epsilon_{ij} \) is \( 0^\circ \), 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:

\[ S_{ij}^2 = \begin{cases} 1 & D_{ij} \leq T_{ab} \\ 1 - \frac{D_{ij} - T_{ab}}{L_{rb} - T_{ab}} & T_{ab} < D_{ij} \leq L_{rb} \\ 0 & L_{rb} < D_{ij} \end{cases} \]  

where \( T_{ab} \) represents the missile range of the blue fighter and \( L_{rb} \) 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.

![Image](image-url)
2.1.3. Speed Threat Function. The speed threat function [22] is given as follows:

\[
S_{ij}^3 = \begin{cases} 
1 & V_{R_i} \leq 0.5 \cdot V_{B_j} \\
1.5 - \frac{V_{R_i}}{V_{B_j}} & 0.5 \cdot V_{R_i} < V_{R_i} \leq 1.4 \cdot V_{B_j} \\
0 & 1.4 \cdot V_{R_i} < V_{R_i}
\end{cases}
\]  

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:

\[
C = \ln B + \ln \left( \sum A_1 + 1 \right) + \ln \left( \sum A_2 \right) \epsilon_1 \epsilon_2 \epsilon_3 \epsilon_4 
\]  

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; \(\epsilon_1\) is the pilot’s control capability coefficient; \(\epsilon_2\) is the fighter survivability coefficient; \(\epsilon_3\) is the fighter range coefficient; and \(\epsilon_4\) is the Electronic Counter Measures (ECM) capability coefficient. The ability threat function is defined as [22]:

\[
S_{ij}^4 = \begin{cases} 
1 & C_j \geq 1.5 \frac{C_j}{C_i} \\
0.75 & 1.5 \geq C_j > 1 \frac{C_j}{C_i} \\
0.5 & 1 \geq C_j \frac{C_j}{C_i} \\
0.25 & 1 > C_j \frac{C_j}{C_i} > 0.3 \\
0 & C_j \frac{C_j}{C_i} < 0.3
\end{cases}
\]  

For example, supposing the air combat capability of \(R_i\) is 17.9, if the capabilities of \(B_j\), \(B_2\), 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

\[
S_{ij} = \omega_1 S_{ij}^1 + \omega_2 S_{ij}^2 + \omega_3 S_{ij}^3 + \omega_4 S_{ij}^4 
\]  

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

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

\[
S_j = \sum_i S_{ij}
\]

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

\[
E_j = \begin{cases} 
1 & S_j \leq 1 \\
2 & 1 < S_j \leq 3 \\
3 & S_j > 3
\end{cases}
\]

Four WTA models are presented in this subsection.

Model 1.

\[
\begin{align*}
\max & \quad \sum_i x_{ij} S_{ij} \\
n & \quad \sum_j x_{ij} \leq M_i, \quad (i = 1, 2, \ldots, N) \\
n & \quad \sum_i x_{ij} \leq E_j, \quad (i = 1, 2, \ldots, K) \\
\end{align*}
\]

\[
k = \sum_i \sum_j x_{ij} = \min \left\{ \sum_i M_i, \sum_j E_j \right\}
\]

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.

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

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.

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

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.

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

\[
P_j = 1 - \prod_i \left( 1 - p_{ij} \right)^{x_{ij}}
\]
Model 4 was first proposed in reference [27], where \( P_i \) 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. At the same time, the minimum hit probability requirement is set for each blue fighter; if \( P_i \) 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\textsubscript{rank}), and max-min ant system (MMAS) and improved the elite strategy in AS\textsubscript{rank}, named AS\textsubscript{comp}. We add garbage ants to the algorithm of AS\textsubscript{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:

\[
P_{ij} = \begin{cases} \frac{[\tau (i, s)]^\alpha \cdot [\eta (i, s)]^\beta}{\sum_{s \in \text{allowed}_k} [\tau (i, s)]^\alpha \cdot [\eta (i, s)]^\beta}, & s \in \text{allowed}_k \\ 0, & s \notin \text{allowed}_k \end{cases}
\]  

(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:

\[
j = \arg \max_{s \in \text{allowed}_k} \left[ \tau (i, s)^\alpha \cdot [\eta (i, s)]^\beta \right], \quad q \leq q_0
\]

\[
j = \arg \max_{s \in \text{allowed}_k} \left[ \tau (i, s)^\alpha \cdot [\eta (i, s)]^\beta \right], \quad q > q_0
\]  

(15)

where \( i \) represents the \( i \)th red fighter and \( j, s, f \) 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. \( \alpha \) is the information heuristic factor used to measure the influence of pheromones on the path.

\( \beta \) is the expected heuristic factor used to measure the influence of the threat degree.

\( \text{allowed}_k \) represents the set of available targets. As the search progresses, \( \text{allowed}_k \) is getting smaller.

\( J_k (i) \) is the set of nodes that the \( k \)th 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, \( 0 \leq q_0 \leq 1 \).
\( p_n \) 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:

\[
\tau (i, j) = (1 - \rho) \cdot \tau (i, j) + \rho \cdot \Delta \tau (i, j)^k
\]  

(16)

\[
\Delta \tau (i, j)^k = \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ passes path } (i, j) \\ 0 & \text{others} \end{cases}
\]  

(17)

where \( \Delta \tau (i, j) \) is the pheromone concentration increment on the optimal path \( (i, j) \); \( \rho \) is the pheromone volatilization rate, \( \rho \in (0, 1) \); \( Q \) is a constant used to regulate the pheromone concentration; and \( L_k \) is the 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:

\[
\tau (i, j) = (1 - \rho) \cdot \tau (i, j) + \rho \cdot \Delta \tau (i, j)^* 
\]  

(18)

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

(19)

where \( \Delta \tau (i, j)^* \) 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\textsubscript{rank} is proposed to preserve them [33]. AS\textsubscript{rank} can find better solutions and find these solutions for a shorter period. Set \( \sigma \) elite ants; the global update rules for AS\textsubscript{rank} are as follows:

\[
\tau (i, j) = (1 - \rho) \cdot \tau (i, j) + \rho \cdot \sum_{l=1}^{\sigma} (\sigma - l + 1) 
\]

\[
\cdot \Delta \tau (i, j)^k
\]  

(20)

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

(21)

(4) Next, the AS\textsubscript{rank} is improved to make it easier to find the optimal path. Garbage ants are added to the algorithm AS\textsubscript{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 $\omega$ is set. The improved AS formula is as follows (named AS$_{\text{comp}}$):

$$
\tau(i, j) = (1 - \rho) \cdot \tau(i, j) + \rho \cdot \sum_{l=1}^{\sigma} (\sigma - l + 1) \Delta \tau(i, j) - \frac{\omega}{L_{gb} - L_s} \tau(i, j)^k
$$

wher $\Delta \tau(i, j)^k$ is the same as formula (21) and $\gamma$ 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]:

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

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

where $\lambda_1$, $\lambda_2$ 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:

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

where $\tau^* (i, j)$ is the amount of pheromone after smoothing; $\delta$ is a constant used to regulate the concentration; and $\tau_{\text{max}}$ 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 defines three parameters in random selection of AS, punishes the rule of AS$_{\text{comp}}$, 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 $[35]$: $m = N_m + \sum E_j$

where $N_m$ 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 $\alpha$, $\beta$, $\rho$, $Q$, and $q_0$. Set $\delta$ to 0.001 and set $\omega$ 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.

<table>
<thead>
<tr>
<th>Path Selection Rule</th>
<th>Pheromone Update Rule</th>
<th>Pheromone Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{ij}$ of basic ACO</td>
<td>Partial update of ACO</td>
<td>None</td>
</tr>
<tr>
<td>Random selections of AS</td>
<td>Partial and global update of AS</td>
<td>MMAS's interval</td>
</tr>
<tr>
<td>Elite ant strategy of $AS_{rank}$</td>
<td>Punish rule of $AS_{comp}$</td>
<td>Smooth the concentration</td>
</tr>
</tbody>
</table>

Table 2: Test results of Algorithm set.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>convergence</th>
<th>optimal</th>
<th>convergence</th>
<th>optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21.36</td>
<td>14.43</td>
<td>4.43</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>19.60</td>
<td>19.77</td>
<td>15.96</td>
<td>20.05</td>
</tr>
<tr>
<td>3</td>
<td>19.03</td>
<td>20.10</td>
<td>13.17</td>
<td>20.03</td>
</tr>
<tr>
<td>4</td>
<td>14.43</td>
<td>15.96</td>
<td>15.64</td>
<td>15.76</td>
</tr>
<tr>
<td>5</td>
<td>4.43</td>
<td>13.17</td>
<td>20.03</td>
<td>20.03</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>20.03</td>
<td>15.76</td>
<td>15.76</td>
</tr>
</tbody>
</table>

Table 3: Model parameter values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>$\beta$</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>$\rho_1$</td>
<td>0.4</td>
<td>0.7</td>
<td>0.7</td>
<td>0.6</td>
</tr>
<tr>
<td>$\rho_2$</td>
<td>0.7</td>
<td>0.5</td>
<td>0.4</td>
<td>0.6</td>
</tr>
<tr>
<td>$q_0$</td>
<td>0.6</td>
<td>0.7</td>
<td>0.7</td>
<td>0.5</td>
</tr>
</tbody>
</table>

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 $\lambda_1 = 3$, $\lambda_2 = 40$.

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:
1. While \( NC - 1 < NC_{\text{max}} \) (The largest iteration number)
2. For \( i = 1:m \) (The number of ants)
3. Put \( m \) ants on red missiles randomly;
4. End
5. For \( k=1:\text{count} \) (The total number of ants’ movements)
6. For \( j=1:m \)
7. For \( i=1:K_b \) (Number of attackable blue fighters)
8. Generate a random number \( q_i \);
9. Select path with pseudorandom probability using formula ((14), (15));
10. Add the target to selected set;
11. Assignable target number minus 1;
12. The number of remaining missiles in red fighters minus 1;
13. \( allowed_i \) minus the assigned target;
14. End
15. End
16. For \( j=1:m \)
17. For \( i=1:N_r \) (The remaining amount of red missiles)
18. Generate a random number \( q_i \);
19. Select path with pseudorandom probability using formula ((14), (15));
20. Add the target to selected set;
21. \( allowed_i \) minus the assigned target;
22. End
23. End
24. Determine the path of the maximum threat in this cycle using WTA model;
25. Global update of pheromone using formula ((22), (23));
26. Update the pheromone interval on the paths using formula (26);
27. \( NC = NC + 1 \);
28. End
29. End
30. Get the optimal WTA solution.

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

### Table 4: Statistical analysis of the model results.

<table>
<thead>
<tr>
<th></th>
<th>Max</th>
<th>Min</th>
<th>Mean</th>
<th>Var</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>17.2896</td>
<td>17.2831</td>
<td>17.2859</td>
<td>0.0021</td>
</tr>
<tr>
<td>Model 2</td>
<td>12.2186</td>
<td>11.8368</td>
<td>12.1267</td>
<td>0.0109</td>
</tr>
<tr>
<td>Model 3</td>
<td>22.1207</td>
<td>18.6858</td>
<td>20.4573</td>
<td>0.5852</td>
</tr>
<tr>
<td>Model 4</td>
<td>6.08764</td>
<td>5.8862</td>
<td>6.0533</td>
<td>0.0024</td>
</tr>
</tbody>
</table>

### Table 5: Red and blue parameter sets.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Red</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of fighters</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Number of missiles</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Flight speed (m/s)</td>
<td>340</td>
<td>340</td>
</tr>
<tr>
<td>Missile range (km)</td>
<td>80</td>
<td>90</td>
</tr>
<tr>
<td>Radar detection distance (km)</td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>Maneuvering parameter (km)</td>
<td>25.5</td>
<td>25.65</td>
</tr>
<tr>
<td>Fire parameter</td>
<td>2761</td>
<td>2832</td>
</tr>
<tr>
<td>Detection capability parameter</td>
<td>1514</td>
<td>1553</td>
</tr>
<tr>
<td>Manipulation efficiency factor</td>
<td>0.9</td>
<td>0.95</td>
</tr>
<tr>
<td>Viability coefficient</td>
<td>0.915</td>
<td>0.995</td>
</tr>
<tr>
<td>Voyage coefficient</td>
<td>1.14</td>
<td>1.14</td>
</tr>
<tr>
<td>Electronic countermeasure coefficient</td>
<td>1.05</td>
<td>1.1</td>
</tr>
</tbody>
</table>
1. For \( \text{time}=1:T \)
2. Call the WIACO algorithm;
3. For \( i=1:N_r \)
4. For \( j=1:N_b \)
5. 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 \( P_r^i \), \( P_b^i \);
6. End
7. End
8. End
9. For \( k_r=1:N_r \)
10. Get random number \( P_r^k \);
11. If \( P_r^k < P_r \)
12. Red fighter \( k_r \) is shot down;
13. End
14. End
15. For \( k_b=1:N_b \)
16. Get random number \( P_b^k \);
17. If \( P_b^k < P_b \)
18. Blue fighter \( k_b \) is shot down;
19. End
20. End
21. Update the air combat situation;
22. If one termination condition is met, end the loop;
23. \( \text{time}=\text{time}+1; \)
24. End
25. Output simulation results.

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

Table 6: Improved algorithm assignment.

<table>
<thead>
<tr>
<th>Red fighter</th>
<th>Assignment (missiles)</th>
<th>Remaining missiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red1</td>
<td>Blue1 (2); Blue2 (1); Blue6 (1)</td>
<td>0</td>
</tr>
<tr>
<td>Red2</td>
<td>Blue3 (3); Blue6 (1)</td>
<td>0</td>
</tr>
<tr>
<td>Red3</td>
<td>Blue5 (2); Blue4 (1)</td>
<td>1</td>
</tr>
<tr>
<td>Red4</td>
<td>Blue5 (1); Blue4 (2)</td>
<td>1</td>
</tr>
<tr>
<td>Red5</td>
<td>Blue4 (1); Blue7(2)</td>
<td>1</td>
</tr>
<tr>
<td>Red6</td>
<td>Blue4 (1)</td>
<td>3</td>
</tr>
<tr>
<td>Red7</td>
<td>—</td>
<td>4</td>
</tr>
<tr>
<td>Red8</td>
<td>Blue6 (1); Blue8 (3)</td>
<td>0</td>
</tr>
</tbody>
</table>

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:
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 $i$th simulation with $L_r^i$, and that of blue side with $L_b^i$, and the damage rates of red and blue sides, $p^r$ and $p^b$, are calculated with formula (28), (29):

$$p^r = \frac{\sum_{i=1}^{T} L_r^i}{T \times N_r}$$  \hspace{1cm} (28)

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

$$W = \frac{\sum_{i=1}^{T} L_r^i}{\sum_{i=1}^{T} L_b^i}$$  \hspace{1cm} (30)

The smaller the kill ratio is, the greater the advantage of the red side is 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
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
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
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 \( i \)th red fighter
- \( B_j \): The \( j \)th blue fighter
- \( X_{R_i} \): The direction of \( R_i \)
- \( V_{R_i} \): The speed of \( R_i \)
- \( \varepsilon_{ij} \): 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
- \( V_{R_i} \): The speeds of the red fighter
- \( V_{B_j} \): 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
- \( \varepsilon_1 \): The pilot's control capability coefficient
- \( \varepsilon_2 \): The fighter survivability coefficient
- \( \varepsilon_3 \): The fighter range coefficient
- \( \varepsilon_4 \): 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 \( B_j \)
- \( P_{ij} \): 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
- \( \alpha \): The information heuristic factor
- \( \beta \): The expected heuristic factor
- \( allowed_k \): The set of available targets
- \( I_k(i) \): The set of nodes that the \( k \)th ant needs to access
- \( q \): A random number in \([0, 1]\)
- \( q_0 \): A constant, \( 0 \leq q_0 \leq 1 \)
- \( \rho \): 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
- \( \gamma \): The number of ant garbage ants
- \( \lambda \): A constant used to regulate the pheromone concentration
Table 10: Model 2 simulation results.

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average damage</td>
<td>7.2821</td>
<td>4.3632</td>
</tr>
<tr>
<td>Damage rate</td>
<td>0.9103</td>
<td>0.5454</td>
</tr>
<tr>
<td>Kill ratio</td>
<td>1.6690</td>
<td></td>
</tr>
</tbody>
</table>

Table 11: Model 3 simulation results.

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average damage</td>
<td>7.2492</td>
<td>4.0854</td>
</tr>
<tr>
<td>Damage rate</td>
<td>0.9062</td>
<td>0.5665</td>
</tr>
<tr>
<td>Kill ratio</td>
<td>1.5993</td>
<td></td>
</tr>
</tbody>
</table>

Table 12: Model 4 simulation results ($M=100$, $P_d=0.2$).

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average damage</td>
<td>7.3215</td>
<td>4.0582</td>
</tr>
<tr>
<td>Damage rate</td>
<td>0.9151</td>
<td>0.5106</td>
</tr>
<tr>
<td>Kill ratio</td>
<td>1.7921</td>
<td></td>
</tr>
</tbody>
</table>

Table 13: Option 1 simulation results.

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average damage</td>
<td>4.5212</td>
<td>2.8727</td>
</tr>
<tr>
<td>Damage rate</td>
<td>0.5652</td>
<td>0.3591</td>
</tr>
<tr>
<td>Kill ratio</td>
<td>1.5762</td>
<td></td>
</tr>
</tbody>
</table>

Table 14: Option 2 simulation results.

<table>
<thead>
<tr>
<th></th>
<th>Red</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average damage</td>
<td>3.9122</td>
<td>2.4820</td>
</tr>
<tr>
<td>Damage rate</td>
<td>0.4890</td>
<td>0.3016</td>
</tr>
<tr>
<td>Kill ratio</td>
<td>1.5762</td>
<td></td>
</tr>
</tbody>
</table>

Table 15: Red fighter number and kill ratio.

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

$\delta$: A constant used to regulate the concentration

$\omega$: The penalty factor

$N_{rm}$: The total number of missiles carried by red fighters

$L_{ri}^a$: The amount of damage of red side in the $i$th simulation

$L_{ri}^b$: The amount of damage of blue side in the $i$th simulation

$P_r^a$: The damage rate of red side

$P_r^b$: 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


