



**ARTICLE**

# An Air Defense Weapon Target Assignment Method Based on Multi-Objective Artificial Bee Colony Algorithm

Huaxi Xing\* and Qinghua Xing

Air and Missile Defense College, Air Force Engineering University, Xi'an, 710051, China

\*Corresponding Author: Huaxi Xing. Email: huaxi\_xing@163.com

Received: 21 September 2022 Accepted: 08 December 2022 Published: 08 October 2023

## ABSTRACT

With the advancement of combat equipment technology and combat concepts, new requirements have been put forward for air defense operations during a group target attack. To achieve high-efficiency and low-loss defensive operations, a reasonable air defense weapon assignment strategy is a key step. In this paper, a multi-objective and multi-constraints weapon target assignment (WTA) model is established that aims to minimize the defensive resource loss, minimize total weapon consumption, and minimize the target residual effectiveness. An optimization framework of air defense weapon mission scheduling based on the multi-objective artificial bee colony (MOABC) algorithm is proposed. The solution for point-to-point saturated attack targets at different operational scales is achieved by encoding the nectar with real numbers. Simulations are performed for an imagined air defense scenario, where air defense weapons are saturated. The non-dominated solution sets are obtained by the MOABC algorithm to meet the operational demand. In the case where there are more weapons than targets, more diverse assignment schemes can be selected. According to the inverse generation distance (IGD) index, the convergence and diversity for the solutions of the non-dominated sorting genetic algorithm III (NSGA-III) algorithm and the MOABC algorithm are compared and analyzed. The results prove that the MOABC algorithm has better convergence and the solutions are more evenly distributed among the solution space.

## KEYWORDS

Weapon target assignment; multi-objective artificial bee colony; air defense; defensive resource loss; total weapon consumption; target residual effectiveness

## 1 Introduction

In the modern complex battlefield environment, with the emergence of high-tech and high-speed aerial weapon platforms, intelligent decision-making and scheduling of air defense weapon resources are particularly important. In air defense and anti-missile operations, weapon target assignment (WTA) is an important issue that needs to be resolved in the battlefield fire attack decision [1–4]. Our maximum effectiveness is the target of air defense operations. Developing a rational air defense WTA planning by determining the threat of target attack is an important topic in the field of military combat research. The design of WTA models and the selection of accurate and efficient intelligent



optimization algorithms in a given situation is the focus and difficulty. Based on the destruction area of multiple target passage numbers for ground-to-air missile defense systems, Liu et al. [3] studied the WTA problem of maximizing the interception of enemy targets and satisfying the time constraint, space constraint, and resource constraint. Davis et al. [4] studied the ballistic missile defense problem based on a dynamic perspective with an uncertainty of multiple rounds of flanking fire to maximize the protection of own resources. The single-objective WTA model suffers from the obvious shortcomings of resource wastage and inconsistency, and the current mainstream studies are generally multi-objective WTA models. Given the new characteristics of defense strategies during group target attacks, Nie et al. [5] established a WTA model with principles that aim at the maximum attack effectiveness, the maximum own remaining effectiveness, and the minimum total weapon consumption. Based on the above research, we design the air defense weapon target assignment as a multi-objective optimization model, taking defensive resource loss, total weapon consumption, and target residual effectiveness as optimization objectives.

The intelligent optimization algorithm is an effective method to solve the NP-hard problems, such as the immune-based ant colony optimization (ACO) algorithm [6,7], genetic algorithm (GA) [8], discrete particle swarm optimization (DPSO) [9], etc. All belong to the meta-heuristic algorithm, which can simulate the foraging and reproduction of biological populations. However, these methods can only deal with single-objective WTA issues and cannot be applied to multi-objective optimization models. Nie et al. [5] proposed an optimization framework and specific process based on the NSGA-III algorithm and conducted simulation experiments under specific combat missions. Zhou et al. [10] proposed a D-NSGA-III algorithm to achieve high-dimensional multi-target task assignment for UAV swarms. Long et al. [11] used the data set to train a decision-making neural network to realize air defense weapons intelligent scheduling. As mentioned above, an important issue in air defense weapon target assignment is the selection of an appropriate performance metric. In the modeling that follows, we should optimize the performance metric for the mission WTA scenario under a set of operational constraints.

The targets move fast and can only be destroyed in a short time window. For air defense operations, the requirement of timeliness is high. Considering the short time window of the weapon's striking, it is still necessary to further improve the real-time performance. The computational efficiency and global convergence of NSGA-III algorithm for WTA solving are insufficient. In addition, the actual data samples for combat are small, and it is difficult to obtain an effective decision model through neural network training. The rationality of input simulated data training cannot be verified.

The artificial bee colony (ABC) algorithm, proposed by Turkey scholar Karaboga in 2005, is a swarm intelligence optimization method simulating bees' foraging behaviors [12]. It completes a series of optimization processes by simulating bees foraging. Compared with the above algorithms, it has fewer parameters and is easy to implement. It is widely used in production scheduling, path planning, system optimization, and other fields, and its algorithm research is relatively mature. The multi-objective bee colony algorithm first proposed in the reference [13] is a milestone. It further extends the ability of the algorithm to achieve multi-objective optimization by constructing an external file on the standard ABC algorithm and using better individuals to guide the evolution. The MOABC algorithm has great adaptability to high-dimensional multi-objective optimization, and has significant advantages in improving operating efficiency, increasing the diversity of solution spaces, and reducing computational complexity, effectively compensating for many shortcomings.

To summarize, the contributions of this work are presented in the following list:

The effectiveness of air defense operations is evaluated through indicators including defensive resource loss, total weapon consumption, and target residual effectiveness based on the characteristics of the battlefield environment and operational tasks. A 0–1 integer-constrained multi-objective WTA optimization model is developed.

The MOABC algorithm for air defense weapon target assignment is presented. One-to-one and many-to-one scenarios weapon and target mapping relationships are established by cleverly designed coding rules, respectively. The effectiveness of the algorithm is verified by establishing Archive update non-dominated solutions.

## 2 Related Work

Recently, as one type of the most effective algorithms for solving non-deterministic polynomial hard (NP-hard) problems like MOPs, multi-objective artificial bee colony (MOABC) algorithm have been proved successfully in practice and demonstrated powerful competitiveness. due to its ease of implementation, it has been successfully applied to many real-world problems, such as industrial systems, image processing, and so on. Compared with other evolutionary algorithms, such as particle swarm optimization (PSO) and NSGA, ABC shows strong global search ability. Zhao et al. [14] used the MOABC algorithm based on the limit search strategy. With the bees searching according to individual limits, the search radius is dynamically adjusted to improve the search accuracy and convergence speed, which improves the search accuracy and convergence speed. The proposed algorithm performs well in convergence and distribution. To reduce the number of casualties and damages in disaster management, Niyomubyeyi et al. [15] used the MOABC algorithm for evacuation planning and provided a better evacuation solution. In addition, they compared and studied four meta-heuristic algorithms, AMOSA, MOABC, MSPSO, and NSGA-II, and found that the MOABC algorithm achieves higher quality solutions and a higher level of repeatability [16]. Zhang et al. [17] proposed a multi-objective artificial bee colony optimization (MOABCO) algorithm and applied it to solve the Multi-Objective Test Case Prioritization (MOTCP) problem to improve the efficiency of regression testing.

The MOABC algorithm has unique advantages for solving mission planning and intelligent decision-making in the military field, but its related research in the air defense combat WTA problem is rare. Mao et al. [18] used the ABC algorithm to solve the weapon target assignment problem, but their optimization objective considered only one optimization objective, i.e., the maximum mathematical expectation of all weapon system destruction targets. Gu et al. [19] used the MOABC algorithm to optimize the UAV formation topology based on the formation communication chain length, considering the network delay influence factor and the residual energy imbalance degree, etc. The study showed that the optimization result of MOABC algorithm can improve the formation range. In order to solve the problem of troop deployment and firepower coordination of tank detachments in offensive combat, Chu et al. [20] first put forward the model of tank position deployment and the model of tank firepower allocation. Then, the two-layer iteration strategy was adopted and the bottom iteration was used to solve the firepower allocation model. Finally, the ABC algorithm was used to optimize the tank position deployment scheme. The experimental results show that the convergence speed and convergence accuracy of the ABC algorithm on a specific model are significantly improved. Under the same fitness evaluation, it can significantly improve the timeliness and scientificity of command decision-making in combat.

At present, there is no research on solving multi-objective air defense WTA with MOABC. Inspired by the above research, we design multi-objective-driven WTA models and use the MOABC algorithm for the optimization of air defense WTA schemes to meet the requirements of rapidity and accuracy.

### 3 System Model and Problem Description

#### 3.1 Descriptions of Multi-Objective Programming

The multi-objective programming (MOP) is composed of multiple objective functions, decision variables, and constraints, and there are generally conflicting relationships between objective functions. The model can be described as Eq. (1).

$$\begin{aligned} \min F(x) &= [f_1(x), f_2(x), \dots, f_m(x)]^T \\ \text{s.t. } g_i(x) &\leq 0, i \in I \\ h_j(x) &= 0, j \in I \end{aligned} \quad (1)$$

In Eq. (1),  $x$  is a feasible solution,  $f(x)$  is an objective function,  $g(x)$  and  $h(x)$  are inequality constraints and equality constraints, respectively.

The solution of multi-objective optimization is not unique, but a set of solutions. Pareto dominance is the most common multi-objective solving mechanism. In the case of multiple objective functions, there may be conflicts and incomparable phenomena between objective functions. For example, a solution that is best for one objective function may be the worst for another. While improving any objective function, these solutions called the Pareto solutions will inevitably weaken at least one other objective function, which is also called non-dominated solutions. The set of optimal solutions for multi-objective functions is called the Pareto optimal set. The optimal set forms a Pareto front in the solution space. Pareto proposed the concept of a non-dominated set of multi-objective solutions in 1986, whose basic definition is as follows.

For a multi-objective programming problem, its variable feasible region is denoted by  $\mathcal{S}$ , and the corresponding objective function feasible region is  $\mathcal{Z} = f(\mathcal{S})$ . Given a feasible point  $\mathbf{x}^* \in \mathcal{S}$ , for  $\forall \mathbf{x} \in \mathcal{S}$ , if  $f(\mathbf{x}^*) < f(\mathbf{x})$ ,  $\mathbf{x}^*$  is called the absolute optimal solution. If there is no  $\mathbf{x} \in \mathcal{S}$ , resulting in  $f(\mathbf{x}^*) < f(\mathbf{x})$ ,  $\mathbf{x}^*$  is called the effective solution, which is the Pareto optimal solution of the multi-objective programming problem.

#### 3.2 A Multi-Objective Programming WTA Model

##### 3.2.1 Model Assumptions

In this paper, the optimization goal is to minimize the target residual effectiveness, minimize the defensive resource loss, and minimize the weapon resource consumption, which is also in line with the experience of battlefield commanders. Then the air defense WTA model is constructed to quickly form a strike decision plan. The specific combat scenario is an air defense combat mission. Suppose we have  $K$  important ground defensive resources ( $D = \{d_1, d_2, \dots, d_n\}$ ), and the enemy has a task formation composed of  $N$  potential aerial targets ( $T = \{t_1, t_2, \dots, t_n\}$ ). We deploy  $M$  available weapons to strike the targets ( $W = \{w_1, w_2, \dots, w_m\}$ ). The cost of firing a weapon to strike a target is  $C = \{c_1, c_2, \dots, c_m\}$ , the target unit value vector is  $V = \{v_1, v_2, \dots, v_n\}$ , and the value vector of our defense resources is  $L = \{l_1, l_2, \dots, l_k\}$ .

The deployed firepower resources of  $M$  defense weapons to  $N$  air targets are allocated by setting a Boolean variable  $x_{ij}$ , which  $x_{ij} = 1$  means that the weapon  $w_i$  strikes the target  $t_j$ , and  $x_{ij} = 0$  means

that the weapon  $w_i$  does not strike the target  $t_j$ . Therefore, it can be represented by the “0–1” integer programming model.

### 3.2.2 Objective Functions

#### (1) Defensive resource loss

The purpose of air defense weapons is to destroy aerial targets to secure defense resources from attack. However, in actual air defense operations, the enemy air combat units will attack or even destroy important defense resources, so combat losses are inevitable. Assuming that each aerial target has penetration capability, the target’s ability to strike and destroy is positively correlated to the weapon’s damage probability and the target’s threat level.

$$f_1 = \sum_{k=1}^K l_k \sum_{j=1}^N q_{kj} \prod_{i=1}^M (1 - p_{ij})^{x_{ij}} \tag{2}$$

In Eq. (2):  $p_{ij}$  ( $0 \leq p_{ij} \leq 1$ ) is the damage probability of the  $i$ -th weapon against the  $j$ -th target,  $q_{jk}$  is the threat of the  $j$ -th target to the  $k$ -th one’s defense resource, and  $l_k$  is the value of the  $k$ -th defense resource (task contribution).

#### (2) Total weapon consumption

Assuming that each surface-to-air weapon can only hit one target when weapons attack the opponent’s targets, the total weapon consumption is shown in Eq. (3).

$$f_2 = \sum_{i=1}^M c_i \sum_{j=1}^N x_{ij} \tag{3}$$

#### (3) Target residual effectiveness

The target residual effectiveness is an important indicator to measure the strike and destroy the capability of surface-to-air weapons. The higher the damage probability of the surface-to-air weapon, the lower the target’s residual effectiveness. In the adversarial battlefield scenario, the damage probability of air defense weapons is used to characterize the damage capability, and the target residual effectiveness function is designed as Eq. (4).

$$f_3 = \sum_{j=1}^N v_j \prod_{i=1}^M (1 - p_{ij})^{x_{ij}} \tag{4}$$

To efficiently destroy enemy targets, protect our battlefield defensive resources from attacks, and reduce weapon consumption, the MOP model of WTA shown in Eq. (5) is obtained.

$$\min F = (f_1, f_2, f_3) \tag{5}$$

$$\begin{cases} M \geq N \\ \sum_{j=1}^N x_{ij} = 1, i = 1, 2, \dots, M \\ 1 \leq \sum_{i=1}^M x_{ij} \leq M - N + 1, j = 1, 2, \dots, N \\ x_{ij} \in \{0, 1\} \end{cases} \tag{6}$$

Constraints in Eq. (6): ① The number of weapon resources  $M$  is more than that of aerial targets  $N$ ; ② Each weapon can strike at most one target when performing fire strike missions; ③ The number of weapons allocated to the same target is no more than  $M - N + 1$ .

## 4 A Multi-Objective Artificial Bee Colony Algorithm for WTA

### 4.1 The Mechanism of Seeking Nectar

The multi-objective artificial bee colony algorithm (ABC) is a swarm intelligence optimization algorithm based on the nectar-collecting mechanism of bees. The ABC algorithm has the advantages of few parameters, fast convergence, and simple operation. The description of the ABC algorithm is as follows: The candidate solution of the problem is regarded as a nectar source, and bees are divided into three types according to the division of labor, including employed bees, onlooker bees, and scout bees. Employed bees are responsible for the initial search for nectar source and sharing information. Onlooker bees are responsible for staying in the hive to collect nectar according to the information provided by the employed bees. After the original nectar source is abandoned, the scout bees randomly search for a new nectar source to replace the original one. After initializing the bee colony and nectar source, the three operations, including the phase of employed bees, onlooker bees, and scout bees, are performed repeatedly to find the optimal solution.

#### (1) Initialize the bee colony

Initialize the algorithm parameters and randomly generate a certain amount of nectar source  $X = (x_1, x_2, \dots, x_N)$  as feasible solutions within the feasible region, where  $x_i = (x_i^1, x_i^2, \dots, x_i^D)^T$ .  $D$  is the dimension of the solution vector, and Eq. (7) represents randomly generated nectar source.

$$x_i^j = x_{\min}^j + rand[0, 1] \times (x_{\max}^j - x_{\min}^j) \quad i = \{1, 2, \dots, N\}, j = \{1, 2, \dots, D\} \quad (7)$$

In Eq. (7),  $x_i^j$  is the  $j$ -th dimension of the  $i$ -th nectar source.  $rand[0, 1]$  is a random number of  $[0, 1]$ .  $x_{\max}^j$  and  $x_{\min}^j$  are the maximum and minimum values of the  $j$ -th dimension of all nectar source, respectively.

#### (2) Employed bees phase

Employed bees find new nectar source according to Eq. (8). The value of the nectar source is evaluated by comparing the nectar quality (fitness value) of each nectar source. The larger the value, the better the quality of the feasible solution. The greedy mechanism is used to select the nectar source and retain the better nectar source.

$$v_i^j = x_i^j + \phi_i^j(x_i^j - x_k^j) \quad (8)$$

In Eq. (8),  $x_k$  ( $k = \{1, 2, \dots, N\}$ ) is the nectar in the neighborhood of  $x_i$ .  $\phi_i^j$  is a random number from  $-1$  to  $1$ , which affects the convergence speed of the algorithm.

#### (3) Onlooker bees phase

After the employed bees find a new source of nectar, the employed bees select onlooker bees to follow based on a roulette strategy and then track and develop the nectar source. Like employed bees, a new nectar source location is locally searched according to Eq. (9), and the better nectar source is retained following the principle of the greedy mechanism. If the quality of the nectar source is better than the original nectar source, the existing nectar source is replaced with a count of the  $trial(i) = 0$ , otherwise  $trial(i) = trial(i) + 1$ . The selection probability of onlooker bees is shown in Eq. (10) and  $F_i$



is the fitness value of the  $i$ -th nectar source.

$$p_i = F_i / \sum_{n=1}^N F_n, n = \{1, 2, \dots, N\} \quad (9)$$

$$F_i = \begin{cases} \frac{1}{1 + f_i}, f_i \geq 0 \\ 1 + |f|, f_i < 0 \end{cases} \quad (10)$$

#### (4) Scout bees phase

If the nectar source has not been updated when the number of searches reaches the upper limit of  $G_{\max}$ , the current nectar source is discarded. The corresponding employed bees and onlooker bees act as scout bees. In the scout bees phase, the scout bees randomly search for new nectar source to replace the current nectar source according to Eq. (7). The scout bees phase effectively avoids falling into the local optimum prematurely and improves the global search capability of the algorithm.

### 4.2 Archive Update

In order to apply the artificial bee colony algorithm to multi-objective optimization, it is necessary to improve the original algorithm and set up an external file-handling mechanism. The high-quality solutions generated in each iteration are stored in the Archive [15], and the internal nectar source is constantly updated during the iteration. The detailed steps are as follows:

1. The solution generated during the iteration is discarded if it is dominated by the solution in the Archive.
2. If the new solution dominates one or more solutions in the Archive, the solutions dominated in the Archive are replaced by the new solution.
3. All the solutions in the Archive are independent of each other, and a new solution is added to the Archive.

The addition of new non-dominated solutions to the Archive can effectively maintain group diversity. When the number of solutions in the Archive reaches the limit, similar individuals need to be eliminated according to the dominance level and crowdedness distance to satisfy the Archive space.

1. Calculate the objective function  $f$  ( $f_1, f_2, \dots, f_n$ ) of all solutions, and find the maximum  $f_{i\max}$  and minimum  $f_{i\min}$  of each objective function.
2. Calculate the crowdedness distance  $D_c(i)$  of each solution according to Eq. (11).

$$D_c(i) = \sum_{j=1}^k \frac{f_i(j+1) - f_i(j-1)}{f_{i\max} - f_{i\min}} \quad (11)$$

In Eq. (11) and Fig. 1,  $f_i(j+1)$  and  $f_i(j-1)$  are the values of the  $i$ -th objective function corresponding to the two adjacent nectar source of the  $j$ -th nectar source.

3. If the number of solutions in the Archive exceeds the limit, the solution with a higher degree of congestion is preferred, that is, the solution with a smaller  $D_c$  is removed from the Archive.

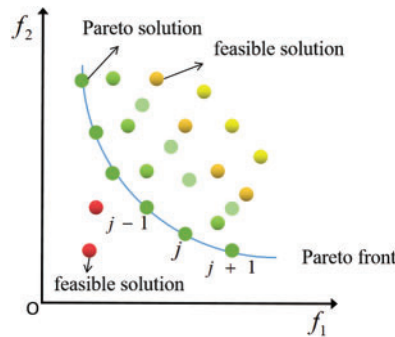


Figure 1: Congestion diagram

### 4.3 Encoding and Constraint Processing

In this paper, real number vectors are used to encode individuals in the population. For a scenario where  $M$  air defense weapons strike  $N$  aerial targets, the dimension  $D = M$ . Each dimension is represented by a random real number in the interval  $(0, N + 1)$ . The integer represents the target number (except when the integer part is 0), and the order of decimals represents the weapon serial number of the corresponding target.  $M$  air defense weapons are numbered in descending order. Each weapon strike at most one opponent's target, so not all weapons are assigned. When the integral part of a dimension is 0, the corresponding air defense weapon is not assigned. For example, it is supposed that 5 air defense weapons strike 4 aerial targets in a combat scenario, and the individual positions of the population are represented by random real numbers in  $[0, 5]$ . In Fig. 2,  $x_1 = 2.91$ ,  $x_3 = 2.46$ , the integer part is 2, and the decimal part in descending order is 1st and 5th, which means that  $w_1$  and  $w_5$  jointly attack the second target. Compared with binary encoding, real number encoding effectively utilizes the characteristics of the bee colony and greatly reduces the dimensionality of the computation.

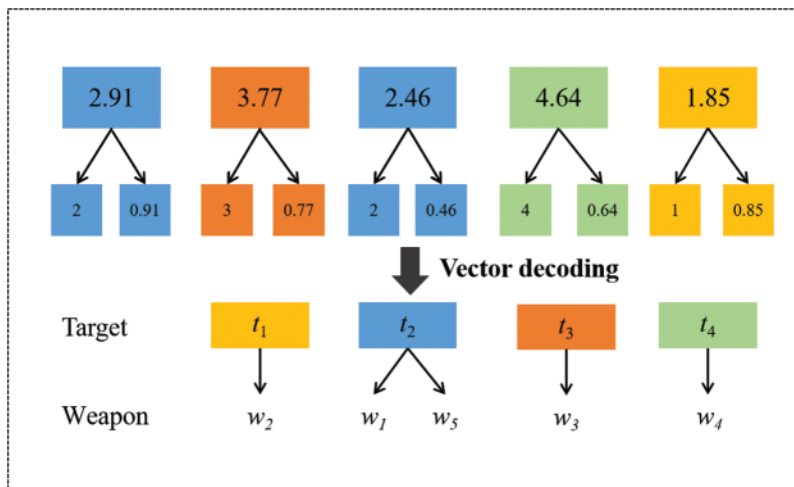


Figure 2: An example diagram of correspondence between coding and weapon target assignment scheme

Generally, multi-objective optimization algorithms only focus on the domain of variables, in other words, all dimensions of the solution are within the boundary. However, in the process of searching for new nectar source, infeasible solutions that do not meet the constraints may be generated. In this



regard, it is necessary to adopt a repair strategy for individuals who do not meet the constraints to ensure the feasibility of the solution. There are two principles of repair. 1. The decimals of each dimension are different. 2. The integers contain the number of all targets.

For example, the position of the nectar source after one update is represented by the vector in Eq. (12).

$$J = [x_1, x_2, \dots, x_d] \quad (12)$$

If the integers of two or more elements in  $J$  are equal, there must be some integers in the interval  $(0, N+1)$  that do not exist, which means that these solutions do not satisfy the second constraint. Because there are unassigned targets, these solutions are infeasible and must be fixed.

1. If there are as many weapons as targets, select elements with the equivalent integer, and change the integers of these elements to the serial numbers of unassigned targets.
2. If there are more weapons than targets and there are unassigned targets, the repairing strategy is the same as 1, otherwise, no repair is required.

#### 4.4 The Pseudo-Code of MOABC Algorithm

Based on the above principle, the pseudo-code of the MOABC algorithm is as follows.

---

##### Algorithm 1: MOABC algorithm for solving the WTA problem

---

```

01:      Initialization: Generate  $N$  solutions  $X$  based on the initialization method
02:      FoodNumber = Population Size/2;
03:      Define the archive size  $I_m$ ,
04:      Maximum number of objective function evaluations  $G_{max}$ 
05:      for  $i = 1: FoodNumber$ 
06:      Evaluate the initial solution  $f(x_i)$ ;
07:       $trial(i) = 0$ ;
08:      end
09:      repeat
10:      //Send employed bees
11:      for  $i = 1: FoodNumber$ 
12:      Select a parameter  $d \in \{1, 2, \dots, D\}$  randomly;
13:      Select a neighbor  $x_k$ ,  $k \in \{1, 2, \dots, FoodNumber\}$  from population randomly;
14:      Generate a new solution  $v_i$  with Eq. (8) then round the solution;
15:      Handle the constraint by the method in Section 4.3;
16:      Evaluate the new solution  $v_i$ ;
17:      if  $v_i$  dominates  $x_i$ 
18:      Update the solution,  $trial(i) = 0$ ;
19:      else
20:       $trial(i) = trial(i) + 1$ ;
21:      end
22:       $G = G + 1$ ;
23:      if  $G == G_{max}$ 
24:      Memorize the best solution achieved so far and exit the main repeat;
25:      end

```

---

(Continued)

**Algorithm 1:** (continued)

---

```

26:           //end employed bees phase
27:           //Send onlooker bees
28:           Calculate probabilities for each nectar source;
29:           for  $i = 1:FoodNumber$ 
30:             Select a parameter  $d \in \{1, 2, \dots, D\}$  randomly;
31:             Select a neighbor  $x_k$  according to the probability;
32:             Generate a new solution  $v_i$  with Eq. (8) then round the solution;
33:             Handle the constraint by the method in Section 4.3;
34:             if  $v_i$  dominates  $x_i$ 
35:               Update the solution,  $trial(i) = 0$ ;
36:             else
37:                $trial(i) = trial(i) + 1$ ;
38:             end
39:            $G = G + 1$ ;
40:           if  $G == G_{max}$ 
41:             Memorize the best solution achieved so far and exit the main repeat;
42:           end
43:           //end onlooker bees phase
44:           //update the archive
45:           Check for archive and remove dominated members;
46:           Save the current Pareto optimal solutions to the archive;
47:           if the number of solutions exceeds  $I_m$ ;
48:             Remove redundant solutions based on congestion  $Dc(i)$ ;
49:           //Send scout bees
50:           Find the index  $i$  that  $trail(i) == \max(trial)$ ;
51:           if  $trial(i) > Limit$ 
52:             Generate a new solution by Eq. (7) and round the new solution;
53:             Handle the constraint by the method in Section 4.3;
54:             Replace  $x_i$  with a new solution;
55:              $trail(i) = 0$ ,  $G = G + 1$ ;
56:           if  $G == G_{max}$ 
57:             Memorize the best solution achieved so far and exit the main repeat;
58:           end
59:           //end scout bees phase
60:           Memorize the best solution achieved so far;
61:           until  $G == G_{max}$ 
62:           Return Archive

```

---

**5 Simulation and Discussion**

The experiment in this paper is conducted on the Windows 10 operating system with the main frequency of 2.9 GHz, and 8 GB of memory. Suppose that several enemy targets attack our important ground defensive resources. To implement effective defensive measures, we deploy several air defense weapons to strike and destroy opponents' targets. Target types include bombers, fighter planes, electronic jammers, unmanned jammers, air-to-surface missiles, etc. Due to the confidentiality of real

battlefield data, we simulated the data based on operational scenarios. For the construction of the WTA model, two operational scenarios are designed in the experiment. ① Weapon-target one-to-one assignment. ② Weapon-target many-to-one assignment.

Algorithm parameter settings: Population size  $2N = 100$ , Archive size  $I_m = 100$ , maximum repetition number  $Limit = N * D$ , and maximum fitness evaluations  $G_{max} = 200D$ .

**5.1 Operational Scenario 1: Weapon-Target One-to-One Assignment**

It is supposed that there are 10 enemy assault targets, 10 air defense weapons, and 6 important ground resources on the battlefield. The target damage probability, target threat level, defensive resource value, target value, and weapon consumption are shown in [Tables 1–5](#).

**Table 1:**  $10 \times 10$  damage probability matrix

ID	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	t <sub>4</sub>	t <sub>5</sub>	t <sub>6</sub>	t <sub>7</sub>	t <sub>8</sub>	t <sub>9</sub>	t <sub>10</sub>
w <sub>1</sub>	0.72	0.13	0.70	0.02	0.15	0.42	0.80	0.28	0.75	0.80
w <sub>2</sub>	0.11	0.54	0.36	0.15	0.80	0.49	0.59	0.14	0.80	0.75
w <sub>3</sub>	0.40	0.64	0.87	0.99	0.16	0.56	0.37	0.77	0.20	0.76
w <sub>4</sub>	0.82	0.50	0.06	0.76	0.96	0.21	0.84	0.15	0.71	0.75
w <sub>5</sub>	0.83	0.93	0.08	0.02	0.15	0.55	0.33	0.32	0.55	0.59
w <sub>6</sub>	0.21	0.39	0.47	0.22	0.09	0.06	0.93	0.33	0.09	0.15
w <sub>7</sub>	0.80	0.32	0.58	0.53	0.44	0.35	0.85	0.69	0.16	0.66
w <sub>8</sub>	0.99	0.05	0.02	0.83	0.78	0.02	0.98	0.03	0.65	0.95
w <sub>9</sub>	0.37	0.52	0.30	0.03	0.18	0.30	0.90	0.68	0.84	0.23
w <sub>10</sub>	0.71	0.81	0.15	0.60	0.06	0.61	0.04	0.32	0.92	0.13

**Table 2:**  $10 \times 10$  target-to-defensive resource threat degree matrix

ID	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	t <sub>4</sub>	t <sub>5</sub>	t <sub>6</sub>	t <sub>7</sub>	t <sub>8</sub>	t <sub>9</sub>	t <sub>10</sub>
d <sub>1</sub>	0.64	0.91	0.88	0.48	0.47	0.22	0.43	0.29	0.66	0.74
d <sub>2</sub>	0.27	0.38	0.21	1.00	0.54	0.27	0.23	0.63	0.13	0.69
d <sub>3</sub>	0.43	0.45	0.85	0.45	0.84	0.61	0.63	0.83	0.65	0.21
d <sub>4</sub>	0.83	0.63	0.72	0.70	0.42	0.26	0.11	0.29	0.28	0.66
d <sub>5</sub>	0.77	0.64	0.67	0.87	0.04	0.27	0.36	0.45	0.33	0.33
d <sub>6</sub>	0.52	0.93	0.83	0.63	0.81	0.21	0.96	0.20	0.11	0.95

**Table 3:** Defensive resource value

ID	l <sub>1</sub>	l <sub>2</sub>	l <sub>3</sub>	l <sub>4</sub>	l <sub>5</sub>	l <sub>6</sub>
	8	7	2	4	5	8

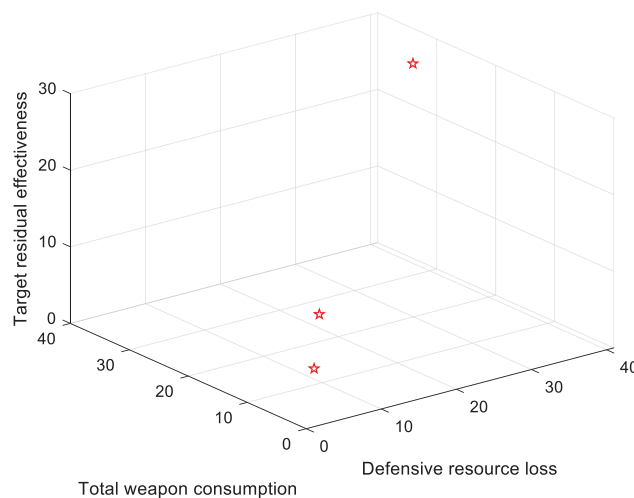
**Table 4:** Target value

ID	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$	$v_7$	$v_8$	$v_9$	$v_{10}$
	4	1	5	1	7	7	10	5	7	6

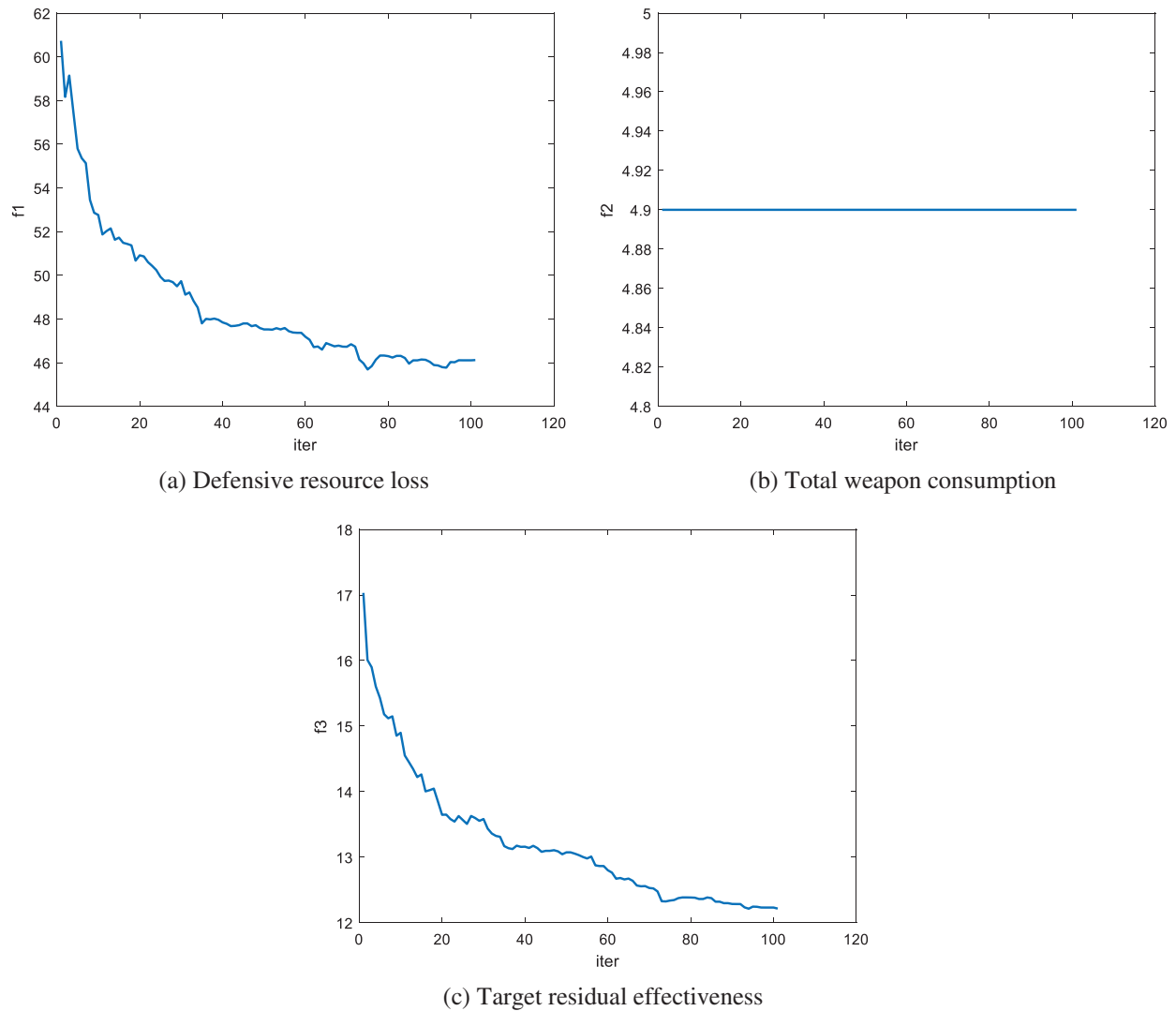
**Table 5:** Weapon consumption

ID	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$	$c_8$	$c_9$	$c_{10}$
	0.5	0.1	0.8	0.6	0.4	0.2	0.4	0.7	0.9	0.3

The distribution plan is solved by the MOABC algorithm. In Fig. 3, the three coordinate axes represent three objective function values. The scattered marks are non-dominated solutions. It can be seen that the distribution of the entire non-dominated solution set is relatively scattered. Because, during the iteration, similar solutions with smaller crowding distances will be continuously eliminated, making the solution set more diverse. It is for decision-makers to consider and choose an assignment plan from different requirements.

**Figure 3:** Weapon-target one-to-one distribution of non-dominated solutions

The MOABC algorithm has good convergence. Since the non-dominated sorting algorithm has a strict selection principle, the Archive has a forced ability to reserve better solutions. In Fig. 4, the defensive resource loss converges to about 46, the target residual effectiveness converges to approximately 12.3, and the total weapon consumption is always the same at 4.90. Since the number of targets is the same as the weapons, all weapons are involved in the strike.



**Figure 4:** Mean values of the objective function value obtained by solving scenario 1 over 20 independent runs in 1 to 100 generations

Table 6 shows each objective function value corresponding to the non-dominated solution. It can be seen that the non-dominated solutions sought are independent of each other. The decision-maker further selects the best plan from the alternatives based on the operational needs. Table 7 shows the WTA plan optimized corresponding to each non-dominated solution.

**Table 6:**  $10 \times 10$  the value of the objective function of the non-dominated solutions

	Defensive resource loss	Total weapon consumption	Target residual effectiveness
1	40.94	4.90	9.07
2	34.03	4.90	9.32
3	25.41	4.90	9.39

**Table 7:**  $10 \times 10$  assignment plan

	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$w_7$	$w_8$	$w_9$	$w_{10}$
1	$t_{10}$	$t_2$	$t_3$	$t_5$	$t_6$	$t_7$	$t_4$	$t_1$	$t_8$	$t_9$
2	$t_1$	$t_6$	$t_3$	$t_5$	$t_2$	$t_7$	$t_4$	$t_{10}$	$t_8$	$t_9$
3	$t_3$	$t_6$	$t_4$	$t_5$	$t_2$	$t_7$	$t_1$	$t_{10}$	$t_8$	$t_9$

**5.2 Operational Scenario 2: Weapon-Target Many-to-One Assignment**

It is supposed that there are 7 enemy assault targets. Other settings are consistent with operational scenario 1. The enemy’s target damage probability, target threat degree, and target value are shown in Tables 8–10.

**Table 8:**  $10 \times 7$  damage probability matrix

ID	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$
$w_1$	0.01	0.08	0.03	0.97	0.84	0.58	0.82
$w_2$	0.97	0.53	0.92	0.70	0.07	0.34	0.25
$w_3$	0.64	0.38	0.33	0.73	0.02	0.97	0.28
$w_4$	1.00	0.34	0.10	0.20	0.15	0.77	0.39
$w_5$	0.82	0.01	0.70	0.35	0.49	0.14	0.62
$w_6$	0.09	0.21	0.03	0.46	0.78	0.10	0.94
$w_7$	0.87	0.40	0.38	0.19	0.30	0.61	0.27
$w_8$	0.63	0.23	0.75	0.04	0.18	0.32	0.72
$w_9$	0.51	0.86	0.88	0.68	0.09	0.77	0.41
$w_{10}$	0.15	0.74	0.04	0.01	0.39	0.79	0.70

**Table 9:**  $10 \times 7$  target-to-defensive resource threat degree matrix

ID	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$
$d_1$	0.17	0.69	0.46	0.87	0.42	0.46	0.48
$d_2$	0.23	0.36	0.49	0.70	0.28	0.07	0.48

(Continued)

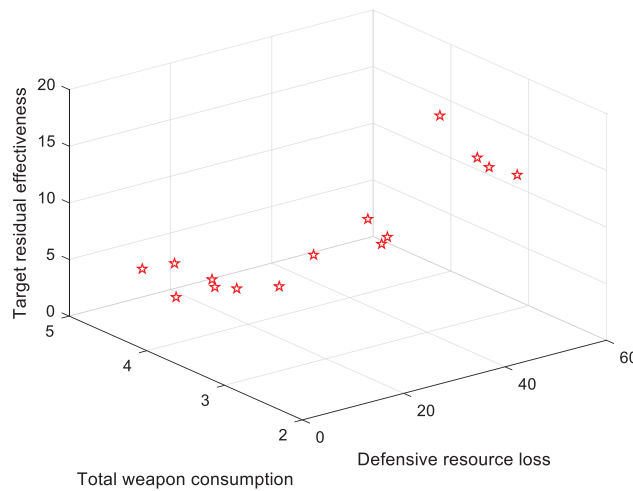
**Table 9 (continued)**

ID	t <sub>1</sub>	t <sub>2</sub>	t <sub>3</sub>	t <sub>4</sub>	t <sub>5</sub>	t <sub>6</sub>	t <sub>7</sub>
d <sub>3</sub>	0.41	0.44	0.95	0.15	0.80	0.04	0.72
d <sub>4</sub>	0.20	0.09	0.37	0.22	0.12	0.02	0.50
d <sub>5</sub>	0.54	0.20	0.14	0.26	0.61	0.94	0.58
d <sub>6</sub>	0.32	0.60	0.72	0.91	0.76	0.59	0.24

**Table 10: Target value**

ID	v <sub>1</sub>	v <sub>2</sub>	v <sub>3</sub>	v <sub>4</sub>	v <sub>5</sub>	v <sub>6</sub>	v <sub>7</sub>
	8	5	7	1	9	2	3

Fig. 5 shows the 15 non-dominated solutions obtained after running. Table 11 shows the objective function values corresponding to the solutions. Compared with operation scenario 1, more non-dominated solutions are obtained, demonstrating that the Pareto solution space is larger and the solution set is more diverse when the weapons are more than enemy targets. Table 12 shows the WTA plan optimized corresponding to each non-dominated solution.



**Figure 5: Weapon-target many-to-one distribution of non-dominated solutions**

**Table 11: The objective function value corresponding to the non-dominated solutions**

	Defensive resource loss	Total weapon consumption	Target residual effectiveness
1	14.39	4.10	4.30
2	19.27	4.90	2.70
3	44.53	2.50	15.56

(Continued)



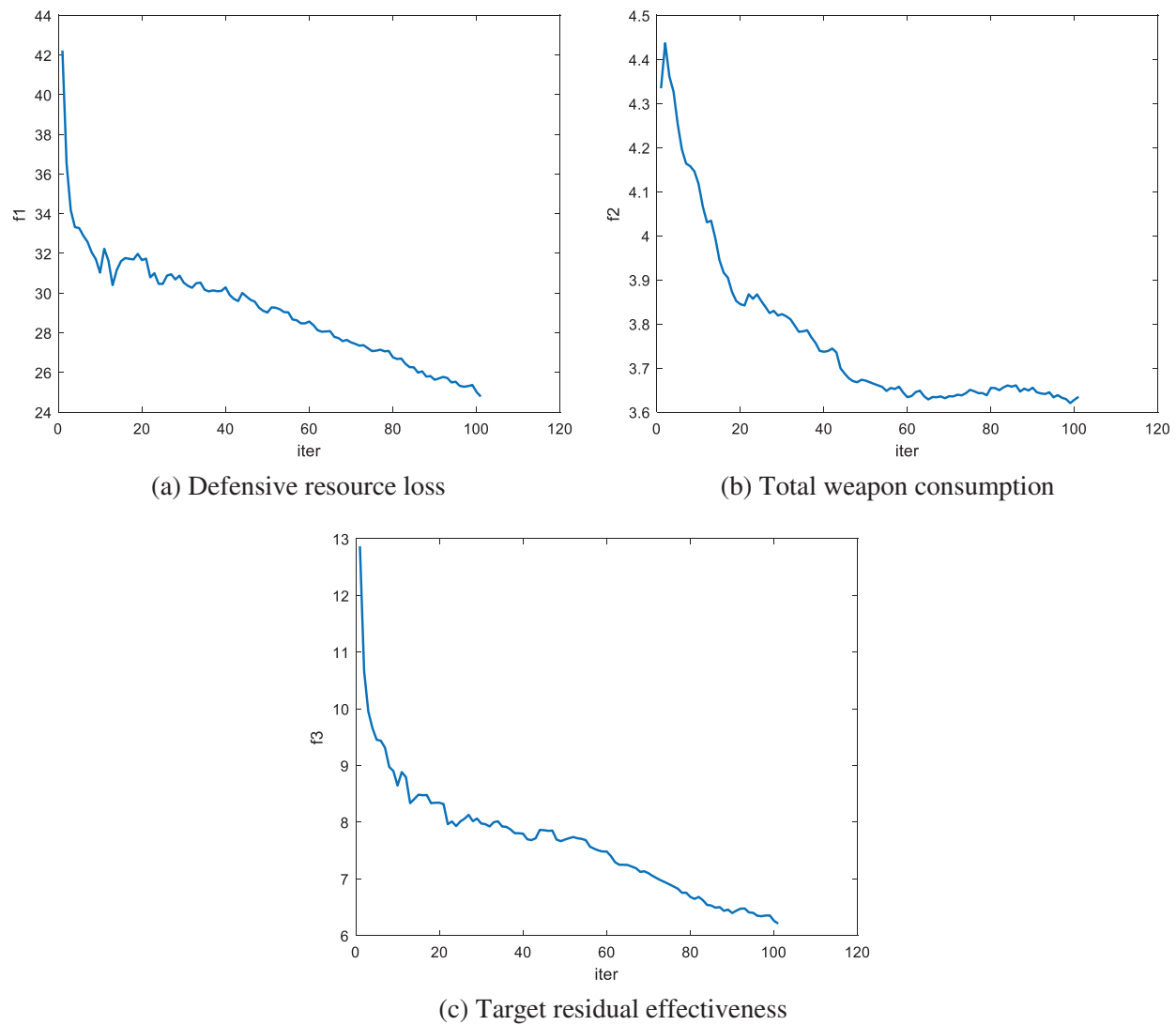
**Table 11 (continued)**

	Defensive resource loss	Total weapon consumption	Target residual effectiveness
4	34.05	3.20	7.87
5	8.84	4.20	3.08
6	12.90	4.90	2.96
7	18.44	3.50	5.04
8	38.26	3.40	7.38
9	23.69	3.40	7.51
10	51.62	2.60	13.74
11	31.33	3.20	10.39
12	14.95	4.10	3.56
13	17.74	4.00	3.40
14	49.87	3.00	14.25
15	44.02	3.10	18.34

**Table 12:**  $10 \times 7$  assignment plan

	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$w_7$	$w_8$	$w_9$	$w_{10}$
1	$t_4$	$t_3$	-	$t_1$	$t_2$	$t_5$	$t_6$	$t_7$	$t_2$	$t_7$
2	$t_5$	$t_3$	$t_6$	$t_1$	$t_5$	$t_4$	$t_5$	$t_7$	$t_2$	$t_2$
3	$t_4$	$t_1$	-	$t_7$	$t_5$	$t_3$	$t_6$	-	-	$t_2$
4	$t_5$	$t_7$	$t_6$	-	$t_1$	$t_4$	-	-	$t_3$	$t_2$
5	$t_4$	$t_3$	$t_6$	$t_1$	$t_7$	$t_5$	$t_5$	-	$t_2$	$t_7$
6	$t_5$	$t_3$	$t_6$	$t_1$	$t_5$	$t_7$	$t_4$	$t_7$	$t_4$	$t_2$
7	$t_4$	$t_1$	-	-	$t_5$	$t_5$	$t_6$	$t_7$	$t_3$	$t_2$
8	$t_5$	$t_2$	-	$t_6$	$t_1$	$t_7$	$t_3$	-	$t_3$	$t_4$
9	$t_4$	$t_1$	-	$t_2$	$t_7$	$t_5$	$t_6$	-	$t_3$	$t_7$
10	$t_5$	$t_2$	-	-	$t_4$	$t_7$	$t_1$	$t_6$	-	$t_3$
11	$t_4$	$t_3$	-	$t_1$	$t_5$	$t_2$	$t_6$	$t_3$	-	$t_7$
12	$t_4$	$t_3$	-	$t_1$	$t_5$	$t_5$	$t_6$	$t_3$	$t_2$	$t_7$
13	$t_5$	$t_3$	$t_4$	$t_1$	$t_5$	$t_7$	$t_6$	$t_3$	-	$t_2$
14	-	$t_1$	-	-	$t_4$	$t_2$	$t_5$	$t_3$	$t_6$	$t_7$
15	$t_4$	$t_2$	$t_5$	$t_6$	$t_1$	-	$t_3$	-	-	$t_7$

In Fig. 6, each objective function value of scenario 2 is converged, the defensive resource loss converges to about 25, and the target residual effectiveness converges to about 6.2. Unlike scenario 1, some weapons are not assigned targets due to the surplus of weapons. The total weapon consumption decreases while searching for a better solution, eventually converging to about 3.62. Overall, the convergence values of all three metrics are better than the results of scenario 1, indicating that the assignment plan not only improves operational effectiveness but also reduces total weapon consumption when increasing the number of air defense weapons.



**Figure 6:** Mean values of the objective function value obtained by solving scenario 2 over 20 independent runs in 1 to 100 generations

### 5.3 Algorithm Performance Comparison

Nie et al. [5] proposed firepower strike assignment for cluster UAV targets based on the NSGA-III algorithm. As a typical representative of multi-objective optimization algorithms, NSGA-III has a wide range of applications and strong practicability. In reference [5], WTA optimization model and framework are close to the current research solution. To better verify the effectiveness of the MOABC algorithm, the global optimization capabilities of the NSGA-III algorithm and the MOABC algorithm are compared. The initial parameter settings of both algorithms are as follows.

NSGA-III algorithm: population size  $2N = 100$ , Archive size  $I_m = 50$ , maximum repetition number  $Limit = N * D$ , and maximum fitness evaluations  $G_{max} = 200D$ , the crossover probability is 0.8, and the mutation probability is 0.05.

MOABC algorithm: population size  $2N = 100$ , Archive size  $I_m = 50$ , maximum repetition number  $Limit = N * D$ , and maximum fitness evaluations  $G_{max} = 200D$ .

The performance is evaluated by inverse generation distance (*IGD*) [21] and running time. *IGD* is a comprehensive performance evaluation index that assesses the convergence and diversity of the solutions. *IGD* can calculate the sum of the minimum distance between each solution on the real Pareto frontier and the solution set obtained. *IGD* is defined as Eq. (13).

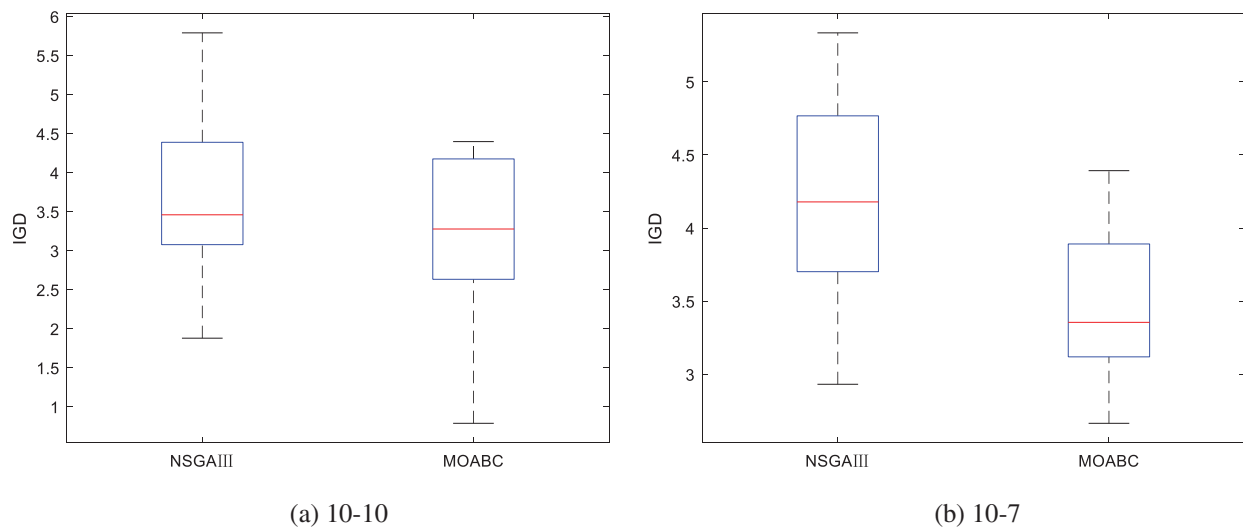
$$IGD(P, Q) = \frac{\sum_{v \in P} d(v, Q)}{|P|} \quad (13)$$

In Eq. (13),  $P$  is the solution set uniformly distributed on the real Pareto frontier.  $|P|$  is the number of solutions distributed on the real Pareto frontier.  $Q$  is the obtained Pareto optimal solution set.  $d(v, Q)$  is the minimum Euclidean distance from the solution  $v$  to the solution set  $Q$ . *IGD* evaluates the convergence and distribution uniformity of the algorithm by calculating the average minimum distance from the solution set on the real Pareto frontier to the solution set obtained. From Eq. (13), the smaller  $d(v, Q)$  indicates that the closer the solution  $v$  is to solution set  $Q$ , the better the convergence performance of the algorithm. However, the larger  $d(v, Q)$  indicates that most of the solutions obtained are concentrated in a narrow region, and the worse the distribution performance of the algorithm. In this paper, some non-dominated solutions with uniform distribution are selected from the solution sets obtained by multiple experiments, and their objective function values are approximated to the real Pareto frontier. The real Pareto frontier is filtered. Next, the solution set obtained by the algorithm is compared with the real Pareto frontier. The results of *IGD* values and average running time after 30 runs are summarized in Table 13. NSGA-III algorithm can not obtain the feedback information of the population in time, so its search speed is relatively slow, and more time is required to obtain more accurate solutions. For the two scenarios, the search efficiency of MOABC algorithm is better than that of NSGA-III algorithm, which reflects the advantages of the algorithm for solving the model in this paper.

**Table 13:** Algorithms' performance comparison

Scale index algorithm	10–10		10–7	
	<i>IGD</i>	running time/s	<i>IGD</i>	running time/s
NSGA-III	3.71	6.54	4.25	8.77
MOABC	2.67	3.92	3.47	8.15

Figs. 7a and 7b are the *IGD* box diagrams of the two operational scenarios, respectively. Comparing the results of both algorithms, the *IGD* value of the MOABC algorithm is superior to that of the NSGA-III algorithm, which shows that the approximate Pareto optimal solution obtained by the MOABC algorithm has better convergence and the solutions are more uniformly distributed in the solution space.



**Figure 7:** IGD box diagrams of NSGA-III and MOABC algorithm

## 6 Conclusion

It is of great military significance to study the issue of WTA in air defense operations. From the perspective of both opposing sides, we evaluated the results of air defense WTA with three indicators, including defensive resource loss, total weapon consumption, and target residual effectiveness. A multi-objective and multi-constrained air defense WTA model is constructed. The MOABC algorithm is used to realize the optimal solution of air defense weapon scheduling at different countermeasure scales, and the efficiency of the algorithm is verified. The performance of MOABC algorithm and NSGA-III algorithm is compared. The former has a lower IGD value of the Pareto solution set than the latter, and the overall quality of the solutions is higher and more uniformly distributed. The running time of the former is less than that of the latter, and the search efficiency of the algorithm is higher. This work provides certain ideas and means for air defense for group targets. Firstly, the applicability of the model has been improved. Multi-objective programming is established to objectively select the optimal assignment scheme, avoiding the commander's personal preference affecting the decision-making results. Secondly, the scientific and reasonable evaluation indicators are designed to compare and verify the optimization results, and closed-loop research is realized. Finally, the Pareto optimal solution set can be regarded as an important reference to assist decision-making, which may also provide the best plan for decision-making commanders.

**Acknowledgement:** The authors are very grateful to the referees for their valuable remarks, which improved the presentation of the paper.

**Funding Statement:** This work was supported by the National Natural Science Foundation of China (71771216).

**Author Contributions:** The authors confirm contribution to the paper as follows: study conception and design: H. Xing, Q. Xing; data collection: H. Xing; analysis and interpretation of results: H. Xing, Q. Xing; draft manuscript preparation: H. Xing. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The authors confirm that the data supporting the findings of this study are available within the paper.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] Y. Lu and D. Z. Chen, "A new exact algorithm for the weapon-target assignment problem," *Omega*, vol. 98, no. 1, pp. 102–138, 2021.
- [2] A. Shojaeifard, A. N. Amroudi, A. Mansoori, A. Mansoori and M. Erfanian, "Projection recurrent neural network model: A new strategy to solve weapon-target assignment problem," *Neural Processing Letters*, vol. 50, no. 3, pp. 3045–3057, 2019.
- [3] F. X. Liu, Y. L. Wang and Q. H. Xing, "Study on problems of optimized target assignment in ground to air defense," *Fire Control & Command Control*, vol. 28, no. 4, pp. 45–48, 2003.
- [4] M. T. Davis, M. J. Robbins and B. J. Lunday, "Approximate dynamic programming for missile defense interceptor fire control," *European Journal of Operational Research*, vol. 259, no. 3, pp. 873–886, 2017.
- [5] J. F. Nie, X. J. Chen and Q. Su, "Modeling and optimization of weapon target assignment for the group targets attacking defense based on NSGA-III algorithm," *Acta Armamentarii*, vol. 42, no. 8, pp. 1771–1779, 2021.
- [6] Y. F. Zhang, D. H. Yan, J. Y. Wang and Z. Cao, "A quasi dynamic defense weapon assignment algorithm based on ant-colony optimization," *Fire Control & Command Control*, vol. 41, no. 9, pp. 1614–1619, 2016.
- [7] 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. 33, no. 2, pp. 1–14, 2017.
- [8] 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.
- [9] C. L. Fan, Q. H. Xing, M. F. Zheng and Z. J. Wang, "Weapon-target allocation optimization algorithm based on IDPSO," *Systems Engineering and Electronics*, vol. 37, no. 2, pp. 336–342, 2015.
- [10] J. Zhou, X. Z. Zhao, Z. Xu, Z. Lin and X. P. Zhang, "Many-objective task allocation method based on D-NSGA-III algorithm for multi-UAVs," *Systems Engineering and Electronics*, vol. 43, no. 5, pp. 1240–1247, 2021.
- [11] T. Long, Z. Y. Liu, R. H. Shi and S. Y. Wang, "Neural network based air defense weapon target intelligent assignment method," *Air & Space Defense*, vol. 4, no. 1, pp. 1–7, 2021.
- [12] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [13] R. Hedayatzadeh, B. Hasanizadeh and R. Akbari, "A multi-objective artificial bee colony for optimizing multi-objective problems," in *Proc. ICACTE*, Chengdu, China, pp. 277–281, 2010.
- [14] X. Q. Zhao, S. Y. Duan and X. M. Ma, "A multi-objective artificial bee colony based on limit search strategy," *Control and Decision*, vol. 35, no. 8, pp. 1793–1802, 2020.
- [15] O. Niyomubeyei, P. Pilesjö and A. Mansourian, "Evacuation planning optimization based on a multi-objective artificial Bee colony algorithm," *International Journal of Geo-Information*, vol. 8, no. 3, pp. 110, 2019.
- [16] O. Niyomubeyei, T. E. Sicutio, J. I. Díaz González, P. Pilesjö and A. Mansourian, "A comparative study of four metaheuristic algorithms, AMOSA, MOABC, MSPSO, and NSGA-II for evacuation planning," *Algorithms*, vol. 13, no. 1, pp. 1–16, 2020.
- [17] N. Zhang, W. Zhang, B. Wu and X. A. Bao, "Multi-objective test case prioritization based on MOABCO," *Journal of Test and Measurement Technology*, vol. 33, no. 2, pp. 93–98, 2019.

- [18] Y. F. Mao and D. L. Zhang, "Improved artificial bee colony algorithm for solving weapon target assignment problem," *Military Operations Research and Systems Engineering*, vol. 29, no. 1, pp. 30–33, 2015.
- [19] X. Y. Gu, L. Chen and X. P. Deng, "Multi-objective optimization of UAV formation information interaction topology," *Electronics Optics & Control*, vol. 29, no. 9, pp. 27–31+52, 2022.
- [20] K. X. Chu, T. Q. Chang, D. P. Kong, L. Zhang and H. Z. Sun, "Bee colony algorithm based model of tank troop deployment and firepower allocation," *Systems Engineering and Electronics*, vol. 44, no. 2, pp. 546–556, 2022.
- [21] H. X. Xing, H. Wu, Y. Chen and X. Zhang, "Multi-efficiency optimization method of jamming resource based on multi-objective grey wolf optimizer," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 46, no. 10, pp. 1990–1998, 2020.