IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 29, 2021, accepted May 7, 2021, date of publication May 11, 2021, date of current version May 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3079152

A Modified MOEA/D Algorithm for Solving Bi-Objective Multi-Stage Weapon-Target Assignment Problem

XIAOCHEN WU[1,2], CHEN CHEN[1,2], AND SHUXIN DING[3]
[1]School of Automation, Beijing Institute of Technology, Beijing 100081, China
[2]State Key Laboratory of Intelligent Control and Decision of Complex System, Beijing 100081, China
[3]Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Ltd., Beijing 100081, China

Corresponding author: Chen Chen (xiaofan@bit.edu.cn)

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61773066 and Grant 62022015.

ABSTRACT In command of modern intelligent operations, in addition to solving the problem of multi-unit coordinated task assignment, it is also necessary to obtain a suitable plan according to the needs of decision makers. Based on these requirements, we established a multi-stage bi-objective weapon-target assignment model, and designed a new algorithm with niche and region self-adaptive aggregation (named MOEA/D-NRSA) based on the decomposition-based multi-objective evolutionary algorithm (MOEA/D) to obtain richer solutions that meet the preferences of different decision makers. Compared with MOEA/D, MOEA/D-NRSA has advantages in improving the convergence and maintaining the distribution of the solution. On the one hand, it contains a population evolution method based on niche technology to obtain better offspring; on the other hand, it has a new neighborhood selection and update strategy. This strategy first clusters the individuals in the objective space to divide into different regions, in which the subproblems can independently select the appropriate aggregation mode according to the clustering density of the region and update its neighborhood. This strategy can improve the uneven distribution of individuals and maintain the diversity and distribution of the population. Numerical experiments selected state-of-the-art algorithms for comparison, which proved the superiority of MOEA/D-NRSA.

INDEX TERMS Multi-stage weapon target assignment (MWTA), decomposition-based multi-objective evolutionary algorithm (MOEA/D), niche, clustering, ideal-nadir Tchebycheff approach.

I. INTRODUCTION

The revolution of intelligent warfare has effectively organized various weapon platforms, thereby achieving a high level of ability matching and increasing the chance of more efficient operations [1]. However, how to solve the task allocation problem and obtain more diverse solutions is still an urgent problem to be solved. The weapon-target assignment (WTA) problem we studied is crucial in the multi-platform cooperative task allocation process. To obtain suitable solutions, we must start with this problem. Lloyd and Witsenhausen have shown that the WTA is NP-Complete [2], it involves how to obtain a set of weapon-target pairs, which can satisfy the decision maker's combat objectives in terms of combat effectiveness and loss [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Pavlos I. Lazaridis.

WTA is divided into two categories: static WTA (SWTA) and dynamic WTA (DWTA). Originally modeled by Manne [4], the SWTA defines a scenario wherein a known number of targets are detected and a finite number of weapons (interceptors), with known probabilities of successfully destroying the targets able for a single exchange. In SWTA, all weapons engage with targets in a single stage, and no subsequent actions are considered, since time is not considered in the problem. The DWTA problem is much more complicated than the SWTA, so the current research results mainly focus on SWTA. The original model was defined by Manne [4], based on this, researchers such as Ahuja *et al.* [1], Lee *et al. et al.* [5], Karasakal [6], and Kline *et al.* [7] proposed various new models with reference to change battlefield requirements. In terms of algorithms, genetic algorithms (GA) [6], [8], [9], ant colony algorithms (ACO) [10], [11], and other heuristic intelligent

optimization algorithms [12]–[14] are widely used in SWTA problems.

However, due to the continuous development of war modes, DWTA problems and related models are more suitable for describing the current war situation with a larger area, multiple combat platforms, and a larger time scale. DWTA contains a time dimension, which is a multi-stage global decision-making process. In solving the DWTA problem, all stages of the offense or defense must be considered, and its distribution results must be solved. Less attention has been given to the DWTA as compared to the SWTA. Thus, there are fewer heuristic algorithms shared among researchers. Currently, hybrid heuristics are often used to solve DWTA problems. Xin *et al*. [15], [16], solved this problem by using virtual permutation (VP) and tabu search heuristics (TSH); and Leboucher *et al*. [17] used Hungarian algorithm and GA-PSO hybrid algorithm in solving DWTA problems.

In the context of DWTA, this paper studies a kind of derivative problem of DWTA, that is, the multi-stage weapon target allocation problem (MWTA). This problem divides the entire combat process into different stages. Each stage needs to make decisions based on the previous stage results and the changes in the battlefield in the new stage and allocate weapons to the target. The decision-making process is the same at each stage, due to different problem scale (number of targets, number of weapons, etc.), there are also differences in computational complexity. The multi-stage weapon target assignment (MWTA) problem is intermediate between the SWTA one and the DWTA one. It also takes time windows into account, but does not have a fully dynamic process as the DWTA does.

At the level of problem models and algorithms, the objective of traditional WTA problems is focused on operational effects. It can also be called the target-based goal, which aims to maximize the expected damage of the targets, while the goal of the asset-based one is to minimize the expected loss. Therefore, many researchers transform this problem into a series of multi-objective optimization problems with constraints. In the process of solving such problems, various multi-objective evolutionary algorithms (MOEA), which are widely used as the primary method to approximate the true Pareto front (PF) of MOPs, have also become the first choice of experts [18]–[20].

As for the study of MOEAs, Schaffer [21] designed the first MOEA. Recently, proposed by Zhang and Li [22], a branch of MOEA based on decomposition has become increasingly popular. It decomposes a multi-objective problem into a set of scale subproblems and optimizes them simultaneously. Because of its advantages in solution efficiency, many researchers have conducted in-depth research on it and achieved certain results [23]–[27].

Unlike the general WTA model, the model studied in this paper considers the possible losses of the weapon platform itself. The task assignment process is to assign the appropriate weapons to the required targets. In the problem model design process, it transforms the WTA problem into a bi-objective optimization problem. We would like to maximize combat results while ensuring fewer combat losses and develop an optimization model with the above two conflicting optimization goals.

In terms of algorithms, after some researchers used MOEAs to solve the task assignment problem, we also used the MOEA/D framework and improved it to form a new algorithm to solve the problem, aiming to generate more differentiated and richer solutions.

The main contributions of this paper can be summarized as follows. Firstly, a bi-objective dynamic collaborative weapon-target assignment model is formulated. The objective used to describe the cost in the model includes ammunition consumption and considers the loss of the combat platform itself. Secondly, a population evolution strategy based on niche technology suitable for the MOEA/D framework is proposed. It selects parents based on the sharing degree of individuals and produces excellent offspring. Thirdly, a neighborhood update strategy based on ideal point and nadir point is proposed to maintain the diversity of evolutionary groups and improve the distribution of solutions.

The remainder of the paper is organized as follows. In Section II, we formulate the problem. The improved algorithm we designed is explained in Section III Some numerical experiment is carried in Section IV. The conclusion and future work are presented in Section V.

II. PROBLEM FORMULATION

Models of WTA problems depend on many factors, e.g., offense or defense strategies, features of targets and weapons, etc. The combat scenarios considered in this paper are as follows. A total of W weapon platforms with ve_t types attack an enemy area together. Our weapons need to explore, discover, and strike against enemy targets continuously. The entire offensive process of weapon-target assignment can be divided into multiple stages, where one stage is the minimum combat time unit. Suppose there are S stages in total.

The offense overview and timeline of each stage are shown in Figs. 1 and 2. In Fig. 1, the units of the enemy and ours are represented by three colors of red, blue, and gray. The red units are our offensive weapons, the blue and gray units together constitute the enemy forces in the area of engagement. Among them, the blue ones are targets that are known and are within our attack range, and the gray ones are unknown targets or targets that cannot be attacked yet. Throughout the offensive process, we stipulate that the length of each phase is equal. In stage t, our entities launch attacks on enemy forces within the range according to the current allocation plan.

In Fig. 2, S_t and S_{t+1} are the starting times of stage t and $t+1$, respectively. The start of the attack marks the beginning of stage t. In stage t, the tasks that need to be performed include offense, advancement, and action adjustment. Each stage contains dynamic events, which are the destruction of old targets (blue dot) and the appearance of new targets (red dot).

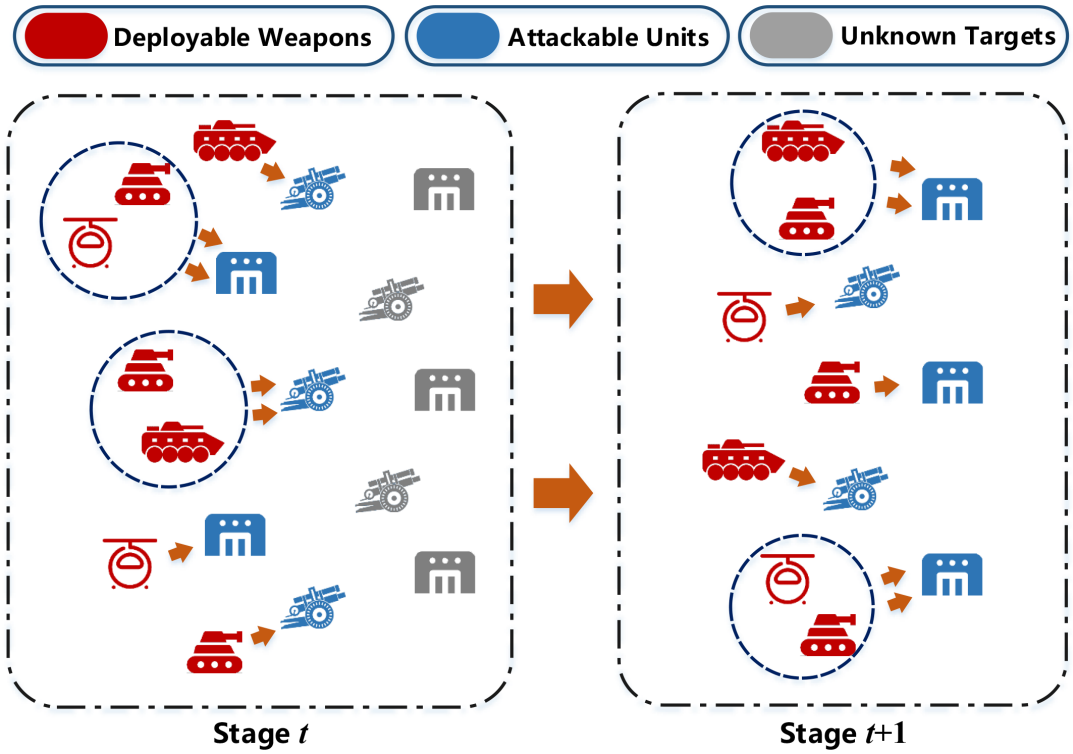

FIGURE 1. Multi-stage weapon-target assignment scenario.

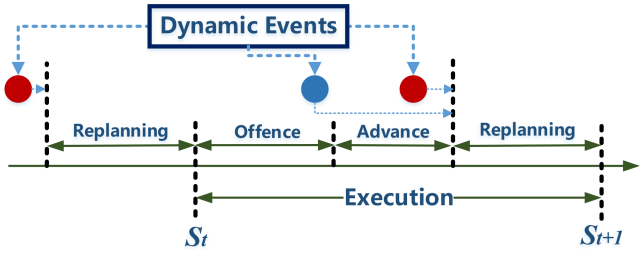

FIGURE 2. Timeline of task assignment.

Returning to Fig. 1, in stage $(t+1)$, the processing of dynamic events generated in the previous stage has been completed, the original blue targets were eliminated and disappeared, and the unknown gray targets were found to turn blue. Afterward, it is necessary to continue to attack existing targets according to the adjusted plan.

Given the set of targets and the set of available weapons, our goal is to find the best weapon-to-target pairing to maximize the probability of damage, while minimizing operations costs.

Through the above description, the MWTA problem is transformed into a bi-objective optimization model. The two optimization goals are the overall damage probability and the cost of operations. The objectives and related constraints will be described below.

A. EXPECTED DAMAGE OF ALL TARGETS

Based on the characteristics of the target-based model, the first objective is to maximize the total expected damage oncoming targets through all stages. The formulation of the expected damage at stage t is expressed as:

$$F_1(t) = \sum_{j=1}^{T} v_j \left(1 - \prod_{s=t}^{S} \prod_{i=1}^{W} \left(1 - p_{ij}(s)\right)^{x_{ij}(s)}\right) \quad (1)$$

where t and s are the indexes of offense stages, $X^t = [X_t, X_{t+1}, \ldots, X_S]$ with $X_t = \left[x_{ij}(t)\right]_{W \times T}$ is the decision matrix at stage t, and $x_{ij}(t)$ is a binary decision variable taking a value of one (i.e., $x_{ij}(t) = 1$) if weapon i is assigned to target i at stage t, or zero (i.e., $x_{ij}(t) = 0$) otherwise. $W(t)$ and $T(t)$ represent the remaining number of weapons and targets at stage t, respectively $(W(1) = W, T(1) = T)$. v_j means the threat value of target j. $p_{ij}(s)$ denotes the probability that weapon i destroys target j at stage s, which is also called kill probability. $p_{ij}(s)$ and v_j can be obtained in advance based on the theory of shooting and performances of weapons.

B. CONSUMPTION

Apart from satisfying the tactical requirement, a WTA decision should also cut down the operational costs. Therefore, when designing the second objective function, we would like to minimize ammunition consumption; besides, in the course of military operations, each side struggles to preserve itself and destroy the other. In the actual situation, considering that the enemy may attack our weapon platform, we need to minimize or even avoid enemy attacks on weapon platforms.

When the second objective was designed, we comprehensively considered the above two requirements and combined

the minimization of ammunition consumption with platform losses to form a new goal. It can be presented as follows:

$$F_2(t) = \sum_{s=1}^{S}\sum_{j=1}^{T}\sum_{i=1}^{W}\left[vw_i re_{ji}(s) + va_i(s)\right]x_{ij}(s) \tag{2}$$

where S represents the total stage of the attack, T is the number of targets, and W is the number of weapons. $va_i(s)$ denotes the ammunition consumption of weapon i at stage s, and vw_i is the value of weapons. $re_{ji}(s)$ represents the ability of the target to resist our attacks at stage s, and its value can be calculated by:

$$re_{ji}(s) = q_{ji}, \tag{3}$$

where q_{ji} represents the probability of target j hitting i.

Through the above analysis, we get two optimization objectives:

$$\max\ F_1(t), \min\ F_2(t).$$

C. CONSTRAINTS

$$\sum_{i=1}^{W} x_{ij}(t) \le Wm_j \quad \forall j \in I_j,\ \forall t \in I_t \tag{4}$$

$$\sum_{j=1}^{T} x_{ij}(t) \le Eu_i \quad \forall i \in I_i, \forall t \in I_t \tag{5}$$

$$\sum_{j=1}^{T}\sum_{t=1}^{S} x_{ij}(t) \le N_i \quad \forall i \in I_i \tag{6}$$

$$I_i = \{1, 2, \ldots, W\}; \quad I_j = \{1, 2, \ldots, T\};$$
$$I_t = \{1, 2, \ldots, S\}$$
$$x_{ij}(t) \le Us_{ij}(t) \quad \forall i \in I_i,\ \forall j \in I_j,\ \forall t \in I_t \tag{7}$$

Constraints (4) and (5) are the feasibility constraints of our weapon platform. Constraint (4) indicates that the number of weapons that can attack target j at each stage cannot exceed Wm_j. This constraint is linked to ammunition consumption for each target at each stage. Constraint (5) means that the maximum number of units attacked by each weapon in a single stage is Eu_i. In most cases, a single weapon can only attack one target, therefore, Eu_i is set to 1 in this paper. Constraint (6) indicates the amount of available ammunitions of weapon i. The last constraint limits the use of weapon i, where U_{si} is a binary variable like x_{ij}, but Us_{ij} is used to constrain the availability of i at stage t. When weapon i cannot attack j (e.g., beyond the range or exhausted ammunition), let Us_{ij} be 1; otherwise, $Us_{ij} = 0$.

The deterministic model is given as follows:

$$\begin{cases} \max & F_1(t) \\ \min & F_2(t) \end{cases}$$
$$\text{s.t.} \quad (4),\ (5),\ (6) \text{ and } (7).$$

The proposed model divides the offense time interval into several fixed stages. During the solution process, the length of each stage is flexible and can be determined according to the actual situation. A minimum length is required to ensure that the platforms can operate effectively at each stage. With the increase in the stage length, the computational complexity of the model decreases, and when the length of a stage is equal to the total time on offense, the model becomes a static version of the assignment model.

In the above model, constraint (6) is an important feature that distinguishes MWTA from traditional SWTA. It reflects the impact of the time window of each stage on the availability of weapons. At the beginning of each stage, the Us_{ij} corresponding to weapon i needs to be updated.

The above model holds a critical assumption: after the weapon is launched, its damage probability does not change. This is reasonable since missiles usually have active terminal guidance.

III. DESIGN OF MOEA/D-NRSA

Since the MWTA problem to be solved in this paper is NP-hard [2], [28], it is impossible to obtain an accurate solution through some polynomial time algorithm. Therefore, we design a new algorithm based on the MOEA/D framework proposed by Zhang and Li *et al*. [22]. This framework decomposes the multi-objective problem into multiple subproblems, optimizes them separately, and then integrates the solutions of the subproblems to form a complete solution. Our algorithm is called MOEA/D-NRSA. "N" means niche technology, which we used for the evolution process of populations, and "RSA" stands for region self-adaptive aggregation method. We divide the objective space into different regions, and independently select the appropriate aggregation function to guide the convergence direction of the population according to the region's characteristics. So MOEA/D-NRSA is mainly dedicated to improving the convergence to a certain extent while maintaining the distribution of the solution. The framework is given in Algorithm 1. We will discuss this in detail in the following subsections.

A. REGIONAL DIVISION AND CLASSIFICATION

When designing algorithms based on the MOEA/D framework to solve bi-objective optimization problems, two main factors need to be considered. The first is to maintain the diversity of the evolutionary group, which is one of the most important indicators to measure the pros and cons of the algorithm; the second is to improve the convergence speed of the algorithm, which is closely related to the efficiency of solving actual problems.

In maintaining the diversity of evolutionary populations, the clustering method is used by many experts and scholars [29], but generally, this method is directly used to generate MOEA. This paper also adopts the clustering algorithm, but it is only used to classify the solutions into different regions as the basis for subsequent operations to maintain population diversity.

1) CLUSTERING METHOD

In the iterative process of the evolutionary algorithm, as the mutation and replacement operations proceed,

Algorithm 1 The Framework of MOEA/D-NRSA

Input:

- F_1, F_2: two objectives;
- N: population number (number of subproblems);
- T_m: size of mating neighborhood;
- EP: an elite population used to preserve the Pareto solution;
- $MaxIter$: maximum iteration;
- $nReg$: the number of regions generated according to the clustering results;
- thd: threshold of cluster density;
- Stopping criterion;

Output:

- EP: final elite population;
- FV: fitness;

Step 1. Initialization

1: Let $EP = \varnothing$;
2: Create an initial population $Pop = \left\{x^1, \ldots, x^N\right\}$ by uniformly randomly sampling. Evaluate the fitness value FV^i of each solution x^i, $FV^i = \left(F_1\left(x^i\right), F_2\left(x^i\right)\right)$ and set $FV = \left\{FV^1, \ldots, FV^N\right\}$.
3: Generate N evenly distributed weight vectors $\lambda = \{\lambda_1, \ldots, \lambda_N\}$. Find the T_m closest weight vectors to each weight vector based on the Euclidean distances of any two weight vectors. Denote by $NS(i) = \left\{i_1, \ldots, i_{T_m}\right\}$ the neighbor set of the ith subproblem.
4: Initialize the ideal point $z = (z_1, z_2)^T$, required by the Tchebysheff approach, set $z_i = \min_j\left(FV_j^i\right)$, and nadir point $z_i^{nad} = \max_j\left(FV_j^i\right), i = 1, \ldots, m$.

Step 2. Update

1: **while** $MaxIter$ is not reached **do**
2: $nReg$ is the number of cluster centers. Use Algorithms 2 and 3 to cluster all the numbers in FV and generate $nReg$ regions.
3: **for** each region k **do**
4: The degree of aggregation in region k is calculated by Algorithm 4.
5: Update the flag fl^k of region k according to the threshold thd of the degree of aggregation.
6: **for** each subproblem i and FV^i in region k **do**
7: Two parents pa_1, pa_2 are selected in the neighborhood by Algorithm 5.
8: Generate offspring of^i through parents and Algorithms 6 and 7. And we evaluate its fitness fo_i.
9: $z_i = \min\left(z_i, fo_i\right), z_i^{nad} = \max\left(z_i^{nad}, fo_i\right), i = 1, \ldots, m$
10: Through Algorithm 8 and the value of fl^k, update the neighborhood of i.
11: **end for**
12: **end for**
13: Update EP.
14: **end while**

new offspring are produced in new positions. At this time, the solution distribution may be uneven. Most of the existing processing methods are oriented to the entire solution set. This paper uses clustering methods to divide individuals into regions with different aggregation degrees for processing separately, thereby maintaining the group's diversity. The specific clustering rules are given by Algorithm 2.

Algorithm 2 first randomly selects a center point for each class, and the remaining individuals are merged into adjacent classes according to the distance from each class. Then re-select the clustering points, form a new class according to the minimum distance principle, and finally repeat this process until the clustering process is stable.

In Algorithm 1, we define $nReg$ as the number of regions, so in Algorithm 2, we randomly generate $nReg$ cluster centers cc_i. $d(i, j)$ represents the degree of dissimilarity between i and j, expressed by the Euclidean distance between the two points.

$$d(i,j) = \sqrt{\left|x_{i1} - x_{j1}\right|^2 + \cdots + \left|x_{ip} - x_{jp}\right|^2} \tag{8}$$

where i and j are both p-dimensional vectors.

The evaluation value Ev can be calculated by:

$$Ev_i = \sum_{p \in cl_i} |cc_i - p|^2 \tag{9}$$

$$Ev = \sum_{i=1}^{nReg} Ev_i. \tag{10}$$

Algorithm 2 Clustering Rules

Input: $nReg, FV$.
Output: cc, cl.

Set cluster center cc, $cc = \{cc_1, cc_2, \ldots, cc_{nReg}\}$.
Set the corresponding class as cl_i, $i \in \{1, 2, \ldots, nReg\}$.
For $i \in \{1, 2, \ldots, N\}$, let $bo = \max\{\max\{F_1(x^i)\}, \max\{F_2(x^i)\}\}$;
On the corresponding coordinate axis, using bo as the boundary, the solution space is equally divided into $nReg$ intervals.
Let $Ev, Ev' = 0$.
while $i \leq nReg$ **do**
    Randomly select an individual in interval $\left[\frac{(i-1)bo}{nReg}, \frac{i \cdot bo}{nReg}\right]$ as cc_i;
    Calculate $d(cc_i, p)$; $cc_i, p \in FV, cc_i \neq p$;
    Let some p with the most remarkable similarity join cl_i;
    Calculate Ev, $Ev = Ev + Ev_i$;
    $i = i + 1$;
end while
while Ev_{min} is not reached **do**
    $Ev = Ev'$;
    for each $q_j \in cl_j$ **do**
        Randomly choose q_j, $q_j \neq cc_j$;
        Calculate $d(q_j, p)$, $q_j \in cl_i$, $p \in Pop$, $q_j \neq p$;
        Calculate Ev', $Ev' = Ev' + Ev'_j$;
    end for
    if $Ev' < Ev$ **then**
        for $k \leq nReg$ **do**
            $cc_k = q_j$;
        end for
    end if
end while

After obtaining the clustering results, we divide the operation area according to it. As shown in Fig. 3, we divide regions in the objective space, and each region has 4 boundaries (lb_1), (lb_2), (ub_1) and (ub_2). Two diagonals (lb_1, lb_2), (ub_1, ub_2) are generally used to identify a region.

Suppose the four boundaries of region i are (lb_{1i}), (lb_{2i}), (ub_{1i}) and (ub_{2i}), and the farthest points in the four directions are $pt_i\left(x^1_{ti}, x^2_{ti}\right)$, $pd_i\left(x^1_{di}, x^2_{di}\right)$, $pl_i\left(x^1_{li}, x^2_{li}\right)$, and $pr_i\left(x^1_{ri}, x^2_{ri}\right)$, respectively. The center point of the i-th cluster is $cc_i\left(x^1_{cci}, x^2_{cci}\right)$. The boundaries of each region can be obtained by Algorithm 3.

2) CLUSTER DENSITY CALCULATION

After completing the regional division, we introduce the concept of influence to calculate the aggregation density of each subspace.

Here, the influence of a certain solution j in the space on the i-th solution is described as:

$$\gamma\left(l_{j\to i}\right) : R \to R \tag{11}$$

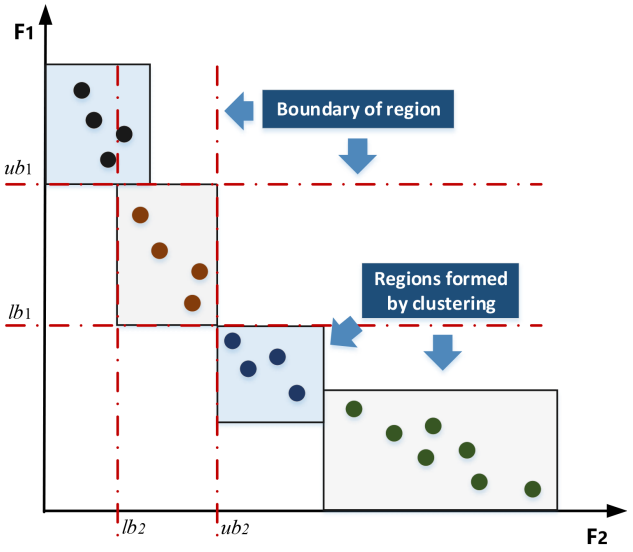

FIGURE 3. Clustering scenario of solution space.

Algorithm 3 Boundary Calculation

Input: $pt, pd, pl, pr, nReg, cc$.
Output: $lb_{1i}, lb_{2i}, ub_{1i}, ub_{2i}$.

if $i = 1$ **then**
    $x^1_{ri-1} = -x^1_{li}$;
end if
if $i = nReg$ **then**
    $x^1_{li+1} = x^1_{ri}$;
end if
if $x^1_{ri} \leq x^1_{li+1}$ **then**
    $ub_{2i} = \frac{(x^1_{li+1} - x^1_{ri})}{2}$;
    $ub_{1i} = x^2_{ti}$;
    if $x^1_{ri-1} \leq x1_{li}$ **then**
        $lb_{2i} = \frac{(x^1_{li} - x^1_{ri-1})}{2}$;
        $lb_{1i} = x^2_{di}$;
    else
        $lb_{2i} = x^1_{li}$;
        $lb_{1i} = \frac{(x^2_{di} - x^2_{ti-1})}{2}$
    end if
else
    $ub_{2i} = x^1_{ri}$;
    $lb_{1i} = \frac{(x2_{di+1} - x^2_{ti})}{2}$;
    if $x^1_{ri-1} \leq x^1_{li}$ **then**
        $lb_{2i} = \frac{(x^1_{li} - x^1_{ri-1})}{2}$;
        $lb_{1i} = x^2_{di}$;
    else
        $lb_{2i} = x^1_{li}$;
        $lb_{1i} = \frac{(x^2_{di} - x^2_{ti-1})}{2}$;
    end if
end if

Among them, $l_{j\to i}$ represents the Euclidean distance of individual j to i, and γ is the mapping function, which transforms $l_{j\to i}$ into the influence of j on i.

Therefore, the cluster density $D(a)$ of the region a can be defined as the mean value of all points in the region to influence the center point. After getting $D(a)$, we will classify the region a according to the density threshold thd.

$$D(a) = \frac{\sum_{i=1}^{m} \gamma(j, i)}{m} \tag{12}$$

The specific algorithm is as follows:

Algorithm 4 Cluster Density Calculation

Input: a, cc_a, thd.
Output: $D(a), fl^a$.

```
Count the number of solutions in a as m;
Let the solution set in a be S_a, S_a = {S_a1, ..., S_am};
Let SumD(a) = 0.
while i ≤ m do
    Calculate γ(l_{S_ai→cc_a});
    SumD(a) = SumD(a) + γ(l_{S_ai→cc_a});
    i = i + 1;
end while
D(a) = SumD(a)/m;
if D(a) ≤ thd then
    fl^a = 1;
else
    fl^a = 0;
end if
```

In Algorithm 4, $\gamma\left(l_{j\to i}\right)$ can be calculated by Gaussian influence function:

$$\gamma\left(l_{j\to i}\right) = \gamma(r) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{r^2}{2\sigma^2}} \tag{13}$$

where r represents Euclidean distance of individual j to i. The threshold thd is the concentration density of the true PF.

B. MODIFICATION OF MOEA/D FRAMEWORK

In this subsection, the Ideal-Nadir Tchebycheff approach that can modify the general MOEA/D framework is designed. In addition, a new neighborhood update and population evolution strategy has also been proposed to generate more good offspring, thereby improving the overall performance of the algorithm.

1) POPULATION EVOLUTION STRATEGY

In the basic MOEA/D framework, a subproblem and its neighbor have the exact equivalence when mating them to generate a new solution. This approach will make the process of selecting parent solutions more random and affect the convergence speed of the population. Therefore, when selecting the parent class, we hope that the weights of different individuals in the neighborhood are different, and the quality of the solution can be improved through appropriate weight settings.

Based on this idea, we propose a niche-guided matching method. This method refers to the niche technology based on the sharing mechanism [30], and introduces the group sharing degree as the weight of the individual. We use S_i to represent the sharing degree of individual i in the group, then:

$$S_i = \sum_{j \in NS_i} sh[d(i, j)], \tag{14}$$

where $d(i, j)$ can be obtained by (8), sh is the sharing function that measures the similarity level between individuals i and j, which is defined as:

$$sh[d(i,j)] = \begin{cases} 0, & d(i,j) > \sigma_{\text{share}} \\ 1 - \dfrac{d(i,j)}{\sigma_{\text{share}}}, & d(i,j) < \sigma_{\text{share}} \end{cases} \tag{15}$$

where σ_{share} is a predefined niche radius [31]. We set the value of σ_{share} to $T_m/2$, and T_m is the size of the neighborhood.

The higher the sharing degree S_i of an individual, the greater the similarity between it and all other individuals in the neighborhood, and the more hopeful it is to generate high-quality offspring for the current subproblem. The weight w_i of individual i in the neighborhood can be defined as:

$$w_i = \frac{S_i}{\sum_i S_i}, \quad i \in NS_i. \tag{16}$$

The specific algorithm is as follows:

Algorithm 5 Select Parents

Input: $\{S_j\}, j \in NS_i$.
Output: pa_1, pa_2.

```
for k = 1 : 2 do
    w_j = S_j / Σ_j S_j, j ∈ NS_i.
    Choose one subproblem pa_k ∈ NS_i based on the
    probability distribution {w_j}.
    Set w_{pa_k} = 0.
end for
```

After getting the parents, we need to generate offspring. The specific details are given by Algorithm 6.

Algorithm 6 Genetic Operator

Input: x_{pa_1} and x_{pa_2}, w_{pa_1} and w_{pa_2}.
Output: Offspring of

```
p_c = w_{pa1} / (w_{pa1} + w_{pa2})
for each genetic locus l do
    if rand < p_c then
        of(l) = x^l_{pa1}
    else
        of(l) = x^l_{pa2}
    end if
end for
```

The offsprings generated by Algorithm 6 may not satisfy the constraints. In this case, we need to repair the solution [32]. The random repair mechanism used is as follows:

Algorithm 7 Random Repair Mechanism

Input: An infeasible solution of, constraints.
Output: A feasible solution of'

1: Transform of into the corresponding 0-1 matrix X;
2: Find the rows or columns that violate this constraint;
3: Calculate the number of redundant 1s in these rows or columns, denote as num.
4: Randomly replace the num 1s in these rows or columns with 0 to obtain X'.
5: Retransform the 0-1 matrix X' into of'.
6: $Region_k$ is a region where the target value FV^i of subproblem i is located.
7: Calculate $FV^{of'}$.
8: **if** $FV^{of'}$ is not in $Region_k$ **then**
9: Regenerate offspring of.
10: **end if**

2) IDEAL-NADIR TCHEBYCHEFF APPROACH

For a long time in the past, many methods to decompose MOP into scalar optimization subproblems have been proposed, such as the weighted Tchebycheff approach [22], NPI-style Tchebycheff approach [33], ε-constraint approach [24], angle-based approach [34], etc. The most representative of the methods is the weighted Tchebycheff approach [22].

These approaches have their advantages and disadvantages, and they apply to different problems. This subsection mainly discusses the Tchebycheff approach and its improvement measures. A scalar optimization subproblem based on the weighted Tchebycheff approach with the ideal point is determined by:

$$\begin{aligned} &\min g^{te}(x \mid \lambda, z) \\ &\quad = \max_{1\le i\le m}\left\{\lambda_i \left| FV^i(x) - z_i \right|\right\} \\ &s.t. \quad x \in Pop \end{aligned} \tag{17}$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)^T$ is the weight vector of the scalar optimization subproblem, and $z = \min\left\{FV^i(x) \mid x \in Pop\right\}$, $i \in \{1, 2, \ldots, m\}$ is the ideal point.

The problems of the Ideal-based Tchebycheff approach are as follows:

(1) As the iterative process continues to advance, points close to z will be retained, which will cause the solutions of the subproblems to continue to gather in the direction of z. When the distance between the solutions of the subproblems corresponding to different λ gradually shrinks, it means that the neighborhood structure is destroyed.
(2) The subproblem solution continues to converge in the direction of z will bring another problem. For a convex Pareto front, this trend will destroy the diversity of the evolutionary population, thereby affecting the distribution of the solution and reducing the reliability of the solution [25].

To avoid the above problems, we considered the nadir-based Tchebycheff proposed by [35] in the process of population convergence. Unlike the ideal-based Tchebycheff, this method strives to make the optimization objective principle of each subproblem to the nadir point in the process of convergence. the scalar optimization subproblem i may also be formulated by:

$$\max\ g^{nte}\left(x \mid \lambda, z^{nad}\right) = \min_{1\le i\le m}\left\{\lambda^i \left| z_i^{nad} - FV^i(x)\right|\right\} \tag{18}$$

where FV is the objective function vector of a solution, z^{nad} is the nadir point. Under the guidance of the nadir point, the population will have better sparsity.

(17) and (18) are often only suitable for MOPs with normalized objective functions. Thus, when the ranges of the objectives are on very different scales, we normalize the objective value to avoid the influence of different dimensions. The subproblem is defined in the following form:

For ideal point:

$$\min g^{te}(x \mid \lambda, z) = \max_{1\le i\le m}\left\{\lambda_i \left| \frac{FV^i(x) - z_i}{z_i^{\text{nad}} - z_i}\right|\right\} \tag{19}$$

For nadir point:

$$\max g^{nte}\left(x \mid \lambda, z^{nad}\right) = \min_{1\le i\le m}\left\{\lambda_i \left| \frac{z_i^{nad} - FV^i(x)}{z_i^{nad} - z_i}\right|\right\} \tag{20}$$

Let

$$\begin{aligned} \tilde{FV}^i_{te} &= \frac{FV^i - Z_i}{Z_i^{nad} - Z_i}, \\ \tilde{FV}^i_{nte} &= \frac{Z_i^{nad} - FV^i}{Z_i^{nad} - Z_i}, \\ i &= 1, \ldots, m, \end{aligned} \tag{21}$$

the optimization goal becomes:

$$\min g^{te}(\tilde{FV}_{te} \mid \lambda, 0). \tag{22}$$

$$\max g^{nte}(\tilde{FV}_{nte} \mid \lambda, 0). \tag{23}$$

This section has designed an Ideal-Nadir Tchebycheff approach (INT), which uses both the ideal point and the nadir point to guide the convergence of the population. Its working principle is shown in Fig. 4. In the figure, according to the clustering rules designed in Section III-A, there are different regions in the objective space, and the individual aggregation density of each region is different. We can judge its aggregation density according to the signs fl^i of each region, and then choose the appropriate reference point and aggregation function to guide the convergence of the individual. For areas with high concentration density (RE_3 and RE_4), choose the ideal point as the reference point, otherwise (for RE_1 and RE_2) choose the nadir point as the reference point. In this way, it can be ensured that the solutions are distributed more widely and more uniformly in the objective space.

In the figure, λ_1 to λ_4 are the weights corresponding to the subproblems. In a two-dimensional space, the relationship between the weight λ_i and the essential direction λ_i' of subproblem i is: $(\lambda_{i1}, \lambda_{i2}) = \left(\lambda'_{i2}, \lambda'_{i1}\right)$.

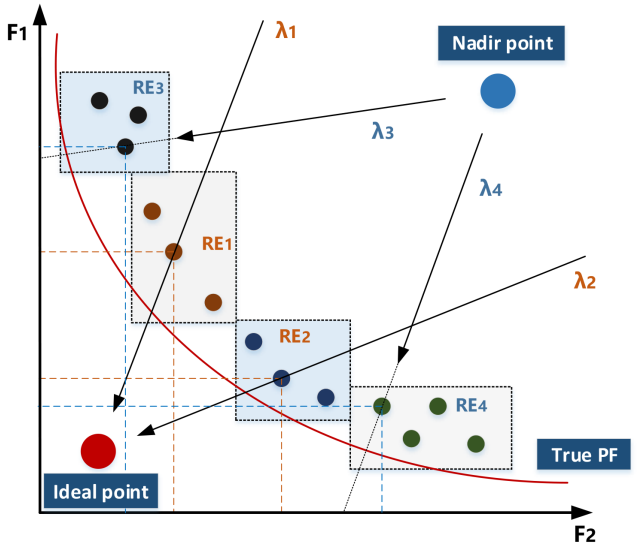

FIGURE 4. Ideal-Nadir Tchebycheff approach.

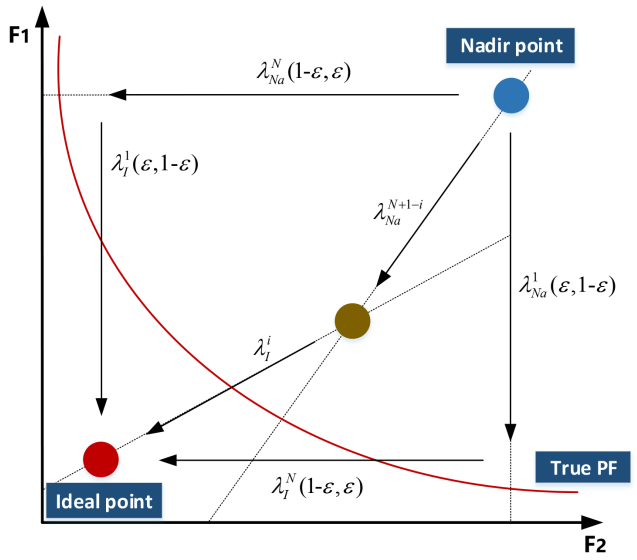

FIGURE 5. Correspondence between subproblems and weight vectors.

In the process of using INT, there will also be a problem-the unification of the weight vector. When the subproblem i updates the neighborhood based on the ideal point z or the nadir point z^{nad}, it corresponds to different weight vectors λ_i or λ_i'. What we need to do is to integrate λ_I^i and λ_{Na}^i into the same weight vector cluster λ.

Fig. 5 shows the objective space, the weight vectors λ_I^1, λ_I^N, λ_{Na}^1 and λ_{Na}^N are set to (ϵ,1-ϵ), (1-ϵ, ϵ), (ϵ,1-ϵ) and (1-ϵ, ϵ), where ϵ is a very small number to avoid abnormal selection. In the objective space, generate N uniformly distributed weight vectors along the clockwise direction to correspond to N subproblems, respectively. Because $\lambda_I^1 = \lambda_{Na}^1$ and $\lambda_I^N = \lambda_{Na}^N$, $\lambda_I^i = \lambda_{Na}^i$ can be obtained.

For the subproblem i in the figure, when the ideal point is selected, its corresponding weight vector is λ_I^i; when the nadir point is selected, its weight vector is λ_{Na}^{N+1-i}. Since

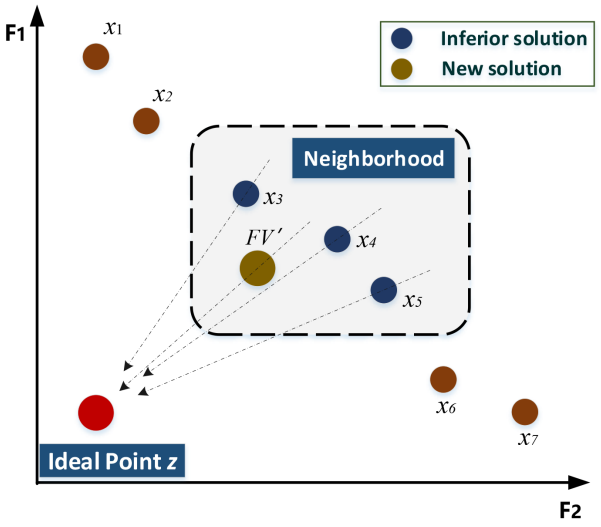

FIGURE 6. Neighborhood update strategy.

$\lambda_{Na}^{N+1-i} = \lambda_I^{N+1-i}$, the different weight vectors corresponding to subproblems i can all be represented by cluster λ_I, and (20) becomes:

$$\max g^{nte}\left(x \mid \lambda, z^{nad}\right) = \min_{1\le i\le m}\left\{\lambda_{N-i+1}\left|\frac{z_i^{nad} - FV^i(x)}{z_i^{nad} - z_i}\right|\right\} \quad (24)$$

3) NEIGHBORHOOD UPDATE RULES

The idea of the traditional MOEA/D neighborhood update strategy is as follows: optimize each subproblem by using the information of several adjacent subproblems, make these problems develop in a favorable direction, and finally converge to the Pareto frontier.

The neighborhood update strategy of MOEA/D generally takes the aggregate function value of the solution and the subproblem as the fitness value, and replaces all inferior solutions in the neighborhood by comparing the fitness value of the new solution with other solutions in its neighborhood.

As shown in Fig. 6, the aggregate function value of FV' is calculated by (17). Through comparison in the neighborhood, it can be seen that x_3, x_4, and x_5 are all inferior solutions compared to FV' and need to be replaced by FV'. Although this operation will speed up the convergence rate of the population, it will also destroy the diversity of the population. As the iteration progresses, the number of effective solutions in other neighborhoods will decrease sharply, which reduces the number of parent samples that can be selected during the evolution of some subproblems and affects the evolution efficiency of the population, making the algorithm fall into a local optimum.

Therefore, when we update the neighborhood, we need to selectively eliminate inferior solutions instead of replacing them all. In this way, the diversity of the population can be maintained to a certain extent.

As an important part of MOEA/D, the neighborhood structure plays an important role in iteration and population update. In the basic MOEA/D framework, the solutions to adjacent subproblems are considered similar to each other. A further consideration is that the closer the two subproblems are, the more similar their solutions will be. Based on this idea, a neighborhood screening strategy is proposed here. We define the relationship between subproblems i and j as [36]:

$$R_{ij} = e^{-\delta \|\lambda_i - \lambda_j\|^2}, \tag{25}$$

where λ_i and λ_j are the weight vectors of subproblems i and j, respectively. δ is a parameter used to adjust the size of R_{ij}. If we count all problems j related to subproblems i, and sum all R_{ij}. Then the proportion r_{ij} of any subproblem j in all relations of i can be defined as:

$$r_{ij} = \frac{R_{ij}}{\sum_j R_{ij}}, \quad j = 1, \ldots, IS. \tag{26}$$

Combining the contents of the previous subsection and this subsection, we can obtain a complete neighborhood update strategy, and the details are given in Algorithm 8.

Algorithm 8 combines the contents of the two subsections. First, select the individual's convergence mode (ideal point or nadir point) according to the flag fl^k of the region k. Then, compare the offspring of^i with the points in the neighborhood $NS(i)$, and count the number of inferior solutions. Finally, the probability distribution can be obtained according to (26). Choose the solution that needs to be eliminated.

According to (25) and (26), we can understand that the closer the inferior solution is to the offspring, the easier it is to be eliminated; otherwise, the corresponding subproblem is more likely to be retained. Through this rule, the diversity of the solution set can be maintained.

Algorithm 8 Neighborhood Update Strategy

Input: NS_i, of^i, λ_{ofi}, fo, fl^k.
Output: Pop, FV.

$z_i = \min\left(z_i, FV^i\right)$;
$z_i^{nad} = \max\left(z_i^{nad}, FV^i\right)$;
Let the set of inferior solutions be I_{so}, $I_{so} = \varnothing$;
if $fl^k = 1$ **then**
  for each subproblem $j \in NS(i)$ **do**
    if $g^{te}\left(\tilde{fo} \mid \lambda_j, 0\right) < g^{te}\left(\tilde{FV}_{te}^{j} \mid \lambda_j, 0\right)$ **then**
      Let x^j merge into I_{so};
    end if
  end for
else
  for each subproblem $j \in NS(i)$ **do**
    if $g^{nte}\left(\tilde{FV}_{nte}^{j} \mid \lambda_{N+1-j}, 0\right) < g^{nte}\left(\tilde{fo} \mid \lambda_{N+1-j}, 0\right)$ **then**
      Let x^j merge into I_{so};
    end if
  end for
end if
Count the number of elements contained in I_{so} and record it as m.
for each inferior solution q **do**
  Calculate R_{jq}.
end for
$r_{jq} = \frac{R_{jq}}{\sum_q R_{jq}}, q = 1, \ldots, m.$
for each inferior solution q **do**
  Choose inferior solution based on the probability distribution $\left\{r_{jq}\right\}$.
  if Inferior solution q is selected **then**
    $x^q = of^i$, $FV^q = fo$;
  end if
end for

C. COMPLEXITY ANALYSIS

In this subsection, We use N to denote the population size (Pop), m to denotes the number of objectives.

There are two parts in MOEA/D-NRSA, the clustering process and the process of generating non-dominated solutions. Suppose the number of iterations of the clustering process is Ir_c, then the time complexity of clustering algorithm is $O\left(m \cdot N \cdot nReg \cdot Ir_c\right)$. The process of generating non-dominated solutions is consistent with MOEA/D which is $O\left(m \cdot N \cdot T_m\right)$. So, the time complexity of MOEA/D-NRSA is $O\left[m \cdot N \cdot \left(nReg \cdot Ir_c + T_m\right)\right]$.

IV. EXPERIMENT AND RESULT ANALYSIS

A. COMPARISON ALGORITHM

In order to prove the effectiveness of the improvement measures, we compared MOEA/D-NRSA with five other algorithms, including multi-objective particle swarm optimization algorithm (MOPSO) [37], non-dominated sorting genetic algorithm II (NSGA-II), and two improved MOEA/D algorithms: MOEA/D with adaptive weight adjustment (MOEA/D-AWA) [38] and MOEA/D with dynamical resource allocation (MOEA/D-DRA) [39]. For the original versions of MOEA/D and NSGA-II [40], we apply the unified crossover and random mutation operators to the algorithm and compare them with MOEAD-NRSA.

B. ENCODING

This paper adopts decimal encoding. The length of the chromosome is the total number of different types of weapon platforms. Each weapon platform is regarded as a gene locus, and the gene value on it indicates the number of targets that the weapon is assigned to. The specific coding form and operation process are the same as those introduced in [41].

Different types of weapons are not coded separately but are integrated into the same chromosome. Their difference lies only in the probability of destruction, the value of the ammunition fired, and the value of the weapon itself.

Such an encoding method can guarantee that every solution satisfies the constraint (5) naturally. As to the other

constraints, each solution will be randomly repaired by Algorithm 7 to satisfy them.

C. PERFORMANCE METRICS

So far, there is no single performance indicator that can comprehensively measure the performance of an MOEA. Therefore, we have introduced several famous metrics to compare the performance of different algorithms.

1) INVERTED GENERATIONAL DISTANCE (IGD) [42]

The IGD metric can measure the diversity and convergence of solutions simultaneously, and it can give the average distance from a given set of non-dominated solutions to the true Pareto front. The smaller the value of $IGD\,(P, P^*)$, the better the performance of P. The formulation is as follows:

$$IGD\left(P, P^*\right) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \tag{27}$$

P^* denotes a set of uniformly distributed points in the objective space along the true Pareto front (PF) or nearly true PF when it is hard or impossible to get the true PF; The set of non-dominated solutions obtained by all comparison algorithms is used as the true Pareto front in this paper. $d(v, P)$ is the minimum Euclidean distance between v and elements in P.

2) GENERATIONAL DISTANCE (GD) [42]

The GD metric measures the average distance from an inverted perspective. This metric is more sensitive to the convergence of the solution. The smaller the value of $GD\,(P, P^*)$, the better the convergence of P. Its formulation is as follows:

$$GD\left(P, P^*\right) = \frac{\sum_{v \in P} \tilde{d}\,(v, P^*)}{|P|} \tag{28}$$

3) DIVERSIFICATION METRIC (DM) [43]

This metric measures the spread of non-dominated solutions on the Pareto front. The larger the value of DM, the wider the solution of this method is distributed in the target space, achieving a better approximation of the Pareto front. It is calculated as follows:

$$DM = \sqrt{\sum_{i=1}^{m} (\min f_i - \max f_i)^2} \tag{29}$$

where $\min f_i$ and $\max f_i$ are the minimum and the maximum value of each fitness function among all non-dominated solutions obtained by the algorithms.

D. PARAMETER SETTINGS AND TEST SCENARIOS

In the numerical experiment, we simulated six different offense scenarios, each of which contains an instance to test the algorithm's performance in different problem scales. Through simulation results and algorithm comparison, we can judge whether MOEA/D-NRSA can generate distribution plans that are highly differentiated and meet the needs of different decision makers.

TABLE 1. Public parameters of algorithms.

Scenarios	Scen1	Scen2	Scen3	Scen4	Scen5	Scen6
Pop	100	100	100	100	100	100
Gen	100	200	250	300	400	500

TABLE 2. Parameters of different scenarios.

Scenarios	Weapons	Stages	Targets
Scen 1	5	5	10
Scen 2	10	8	20
Scen 3	15	11	25
Scen 4	25	12	30
Scen 5	30	15	40
Scen 6	37	18	50

For all algorithms, we first set their public parameters of algorithms in Table 1:

Pop represents the population size, and *Gen* is the number of iterations.

The unique parameter settings in the comparison algorithm are as follows:

(1) Parameter settings in MOPSO adopted here are the same as those claimed in [44].

- The inertia weight: $w = 0.4$.
- The acceleration constants: $c_1 = c_2 = 1.4962$.

(2) Parameter settings in NSGA-II [40]:

- Probability for crossover: $c_r = 0.9$.
- Probability for mutation: $m_r = 1/n$. n represents the number of decision variables.

(3) Parameter Settings in MOEA/D, MOEA/D-AWA, and MOEA/D-DRA adopted here are the same as those claimed in [22], [38], and [39].

- Neighborhood size: $T_m = \lfloor 0.1\,N \rfloor$.
- Probability of selecting mate solutions: $\delta = 0.9$.
- Maximal number of replacement: $n_r = \lfloor 0.01\,N \rfloor$.

The Table 2 gives the relevant parameters of each scenario.

Table 3 shows the settings of the public simulation parameters required for the simulation, where the value expressed in the form of an interval represents that the value of the corresponding parameter is randomly generated in this interval.

E. RESULT ANALYSIS

In this subsection, we perform numerical experiments to compare six different algorithms under six different scenarios and analyze the results. Each scenario corresponds to an instance. Independent runs are performed on each instance 25 times for all algorithms. The program is implemented using MATLAB 2016b software and run on a desktop with 3.4 GHz Core i5-7500 CPU and 8.00 GB RAM. Fig. 7 shows the PF of

TABLE 3. Setting of public simulation parameters.

Parameter	Definition	Value
S	the number of stages.	Given by Table 2
W	the number of weapons.	Given by Table 2
T	the number of target.	Given by Table 2
$p_{ij}(s)$	the destroying probability of weapon i to target j at stage s.	[0.6,0.8]
$q_{ji}(s)$	the hitting probability of target j to weapon i at stage s.	[0.1,0.3]
v_j	the threat value of target j.	[1,100]
va_i	the value of weapon i.	[25,50]
vw_i	Ammunition consumption of weapon i in a single stage.	[5,20]
F_i	the amount of remaining ammunitions of weapon i.	[4,6]
$X^t = [x_{ij}(t)]$	the decision matrix. $x_{ij} = 1$ when target j is assigned to weapon i at stage s, 0 otherwise.	0 or 1
$nReg$	the number of regions generated according to clustering results.	10
T_m	the size of mating neighborhood.	10

TABLE 4. Comparison of IGD.

	MOPSO	NSGA-II	MOEA/D	MOEA/D-AWA	MOEA/D-DRA	MOEA/D-NRSA
Scenario 1	11.1324$^{\approx}$[4] (6.1559)	**3.2964**§[1] (**0.68301**)	14.2667$^{\approx}$[5] (6.8231)	14.3857$^{\approx}$[6] (6.1336)	6.4768§[2] (3.6763)	9.432[3] (3.2441)
Scenario 2	62.2521†[6] (9.9724)	33.5765†[4] (8.257)	26.5712†[2] (6.9178)	28.0543†[3] (9.3934)	43.6096†[5] (13.2294)	**9.1659**[1] (**1.8853**)
Scenario 3	123.5934†[6] (21.1632)	83.0583†[5] (12.5166)	60.5586†[3] (7.5994)	55.0677†[2] (12.6602)	72.9352†[4] (21.6528)	**15.4781**[1] (**4.0882**)
Scenario 4	209.048†[6] (24.1416)	155.7388†[5] (17.0389)	103.9695†[4] (12.5106)	62.9119†[2] (18.8779)	90.9577†[3] (17.6501)	**25.2055**[1] (**7.1735**)
Scenario 5	343.517†[6] (19.789)	269.8616†[5] (16.2021)	199.6392†[4] (17.3752)	99.9532†[2] (6.8487)	121.6246†[3] (15.7116)	**46.0709**[1] (**11.3789**)
Scenario 6	398.8553†[6] (33.4931)	387.7986†[5] (39.8755)	244.0224†[4] (27.8119)	165.8478†[2] (20.4502)	186.9923†[3] (36.3413)	**43.471**[1] (**13.092**)
†/§/≈	5/0/1	5/1/0	5/0/1	5/0/1	5/1/0	

a single run with minimum IGD value on six instances in different scenarios.

For the instances corresponding to each scenario, we give the mean values of the three metrics under different algorithms. In addition, Wilcoxon's rank-sum test with a 5% significance level was performed to compare whether the difference between the mean values of the metrics of MOEAD-NRSA and other algorithms is significant. The symbols †, §, and ≈ indicate that the performance of MOEAD-NRSA is better than, worse than, or similar to that of the comparison algorithm according to Wilcoxon's rank-sum test, respectively. We set it in bold for the best average metric value in each scenario; the standard deviation is in parentheses below the average. The statistical results for the metrics IGD, GD, and DM are presented in Tables 4, 5, and 6, respectively. Fig. 7 shows the PF of a single run with minimum IGD value on 6 instances in different scenarios. The average ranking of the metrics and the statistics of Wilcoxon's test results are given in Table 7. Fig. 8 shows the comparison of IGD box plots of different algorithms in 6 instances.

Through the simulation results, we can understand that when solving small-scale problems, NSGA-II ranks first in the three metrics of IGD, GD, and DM; and the algorithm based on the MOEA/D framework is better in Scenario 1. Performance is unsatisfactory. The main reason is that the small size of the problem means that the search space is small, and there is no noticeable difference in the weight λ corresponding to different subproblems, which will cause the

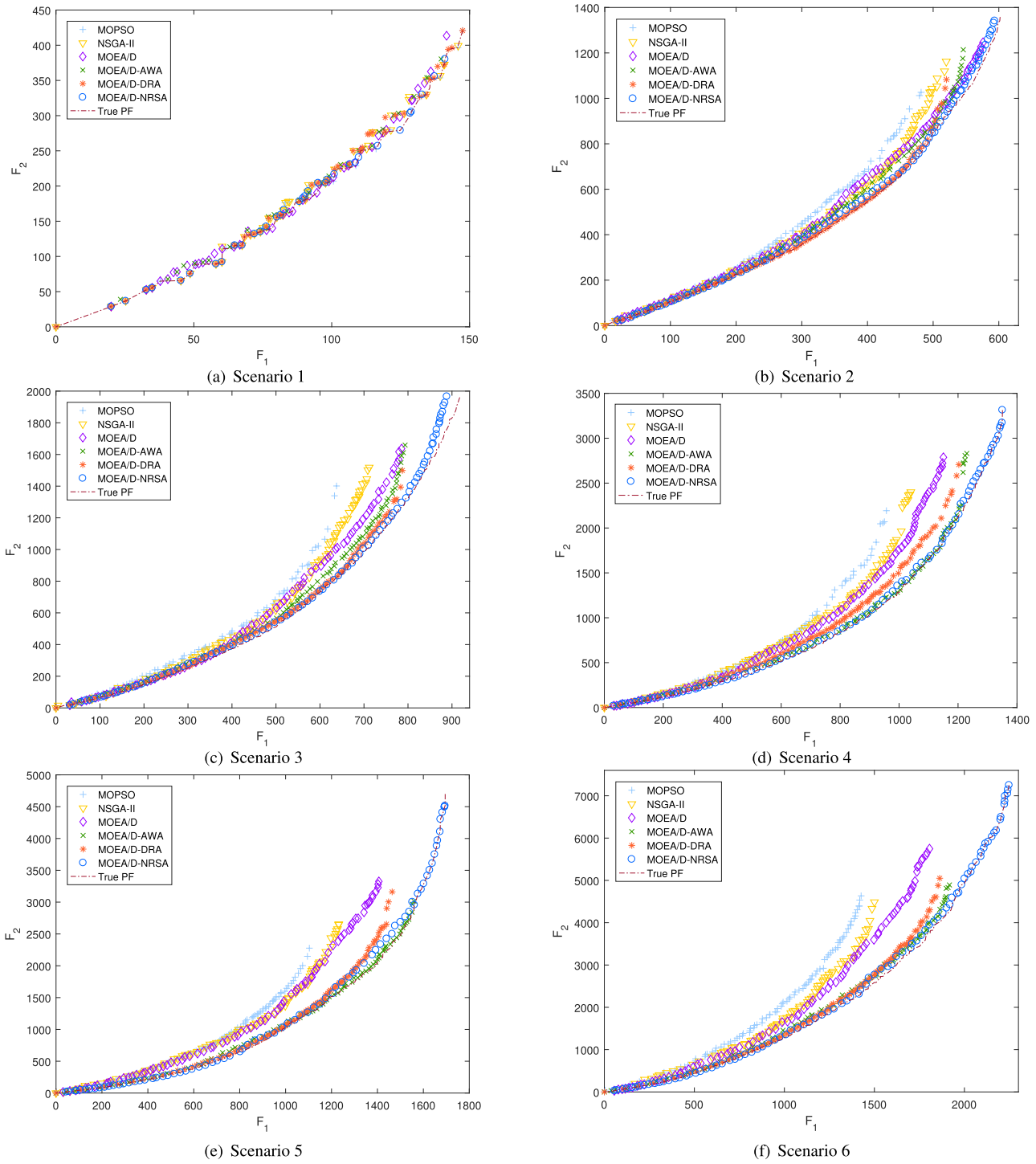

FIGURE 7. (a)–(f): PF of a single run with minimum IGD value on 6 instances in different scenarios.

offspring to lose diversity and make the problem fall into the local optimum. And NSGA-II, which includes the principle of random matching, has more advantages. As the problem scale expands, the advantages of MOEA/D algorithms begin to manifest, especially MOEA/D-NRSA, which ranks first among multiple metrics.

The advantages of MOEA/D-NRSA are undeniable in the distribution of target space points. In Fig. 7, except for instance 1, in other medium-scale and large-scale problems (Scenario 2 - 6), MOEA/D-NRSA can obtain solutions that are closer to the true Pareto front. Especially at the tail of the Pareto front, MOEA/D-NRSA obtains more non-dominated

TABLE 5. Comparison of GD.

	MOPSO	NSGA-II	MOEA/D	MOEA/D-AWA	MOEA/D-DRA	MOEA/D-NRSA
Scenario 1	3.0818[4]† (3.174)	**1.2812**$^{\approx}$[1] (**0.26335**)	4.7363†[5] (3.6167)	5.7631†[6] (3.0528)	1.4392$^{\approx}$[2] (0.28121)	1.5266[3] (0.31527)
Scenario 2	48.3163†[6] (8.3937)	32.2543†[4] (7.3452)	33.6128†[5] (8.9991)	15.5372†[3] (3.4152)	14.7561†[2] (5.5571)	**9.5859**[1] (**2.8479**)
Scenario 3	64.1355†[5] (13.7237)	68.2003†[6] (13.0421)	63.063†[4] (9.2428)	26.2501†[3] (6.6449)	14.7234 $^{\approx}$[2] (5.8366)	**14.3098**[1] (**5.8065**)
Scenario 4	121.8231†[5] (13.1731)	138.3105†[6] (19.4444)	104.0408†[4] (14.083)	**20.3484**$^{\approx}$[1] (**5.583**)	41.0318†[3] (13.3013)	25.25[2] (8.6845)
Scenario 5	170.8446†[5] (13.3775)	196.6103†[6] (24.8935)	160.19†[4] (18.2897)	**20.7399**$^{\approx}$[1] (**4.5024**)	53.7963$^{\approx}$[3] (22.6404)	44.1679[2] (17.7271)
Scenario 6	311.7493†[6] (24.3291)	191.1918†[4] (29.5524)	225.314†[5] (25.1755)	47.549$^{\approx}$[2] (8.6014)	68.7118†[3] (26.5524)	**43.6564**[1] (**18.3537**)
†/§/≈	6/0/0	5/0/1	6/0/0	3/0/3	3/0/3	

TABLE 6. Comparison of DM.

	MOPSO	NSGA-II	MOEA/D	MOEA/D-AWA	MOEA/D-DRA	MOEA/D-NRSA
Scenario 1	377.712$^{\approx}$[3] (52.5667)	**413.1078**§[1] (**12.0186**)	331.2911$^{\approx}$[6] (49.9206)	359.2254$^{\approx}$[4] (30.8291)	389.9703§[2] (31.1231)	334.7466[5] (24.1882)
Scenario 2	1054.0429†[5] (85.5321)	1207.9664†[3] (81.5932)	1372.9418$^{\approx}$[2] (51.9732)	1159.195†[4] (108.7919)	1030.3172†[6] (98.1293)	**1389.0565**[1] (**54.9982**)
Scenario 3	1366.0978†[6] (149.5258)	1655.7849†[3] (116.5295)	1925.9335†[2] (71.7555)	1599.2243†[4] (133.7776)	1428.473†[5] (143.3828)	**2154.4427**[1] (**51.2542**)
Scenario 4	2138.5582†[6] (174.2385)	2604.7803†[4] (183.4858)	2996.1196†[2] (105.347)	2650.4569†[3] (188.2114)	2515.6398†[5] (149.0849)	**3619.9107**[1] (**66.953**)
Scenario 5	2406.6914†[6] (71.7714)	2993.0105†[5] (136.4842)	3390.1737†[3] (90.6365)	3432.6473†[2] (60.1137)	3337.3979†[4] (240.5576)	**5089.2727**[1] (**214.7726**)
Scenario 6	4775.9735†[5] (279.6756)	4173.4419†[6] (321.8854)	5767.6964†[2] (208.8984)	5043.8819†[3] (201.8037)	4999.5105†[4] (422.7012)	**7316.0218**[1] (**160.0561**)
†/§/≈	5/0/1	5/1/0	4/0/2	5/0/1	5/1/0	

solutions than other algorithms. The IGD and DM values in Tables 4 and 6 can also illustrate this advantage, which reflects the effectiveness of the INT method. The INT method can obtain a better approximation of the whole Pareto front and make the distribution of the solution more even, maintaining the diversity of the evolutionary population to a certain extent.

However, the improvement of DM metrics means the decline of GD metrics. When the metric value of DM increases, the area equivalent to P that needs to be calculated by (28) becomes more extensive, so the metric value of GD will become worse. This phenomenon is also reflected in Table 3. The GD measurement value of MOEA/D-NRSA in Table 5 does not have a great advantage. After statistics, the GD value of MOEA/D-AWA is better. However, according to the results of Wilcoxon's rank-sum test, it can be seen that the GD measures of MOEA/D-NRSA and MOEA/D-AWA are similar, there is no significant difference between the GD measures of MOEA/D-NRSA and MOEA/D-AWA, indicating that the two are at the same level. The numerical approximation of the GD metric also shows that the convergence of MOEA/D-NRSA is only slightly inferior to MOEA/D-AWA. This result proves the effectiveness of our proposed population evolution strategy to a certain extent.

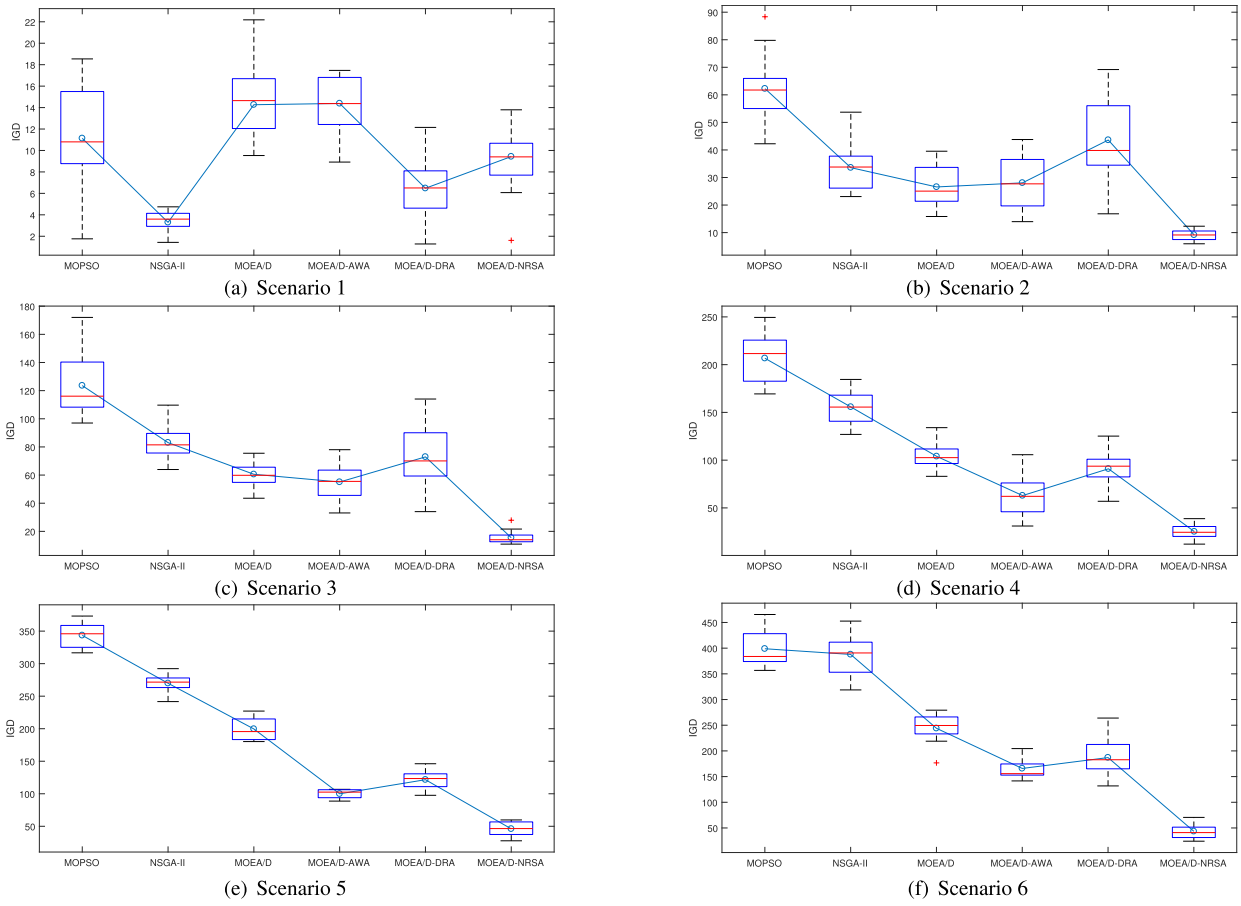

FIGURE 8. (a)–(f): Comparison of IGD box plots of different algorithms in 6 instances.

TABLE 7. Overall performance of four algorithms on the six instances in terms of IGD, GD, and DM metrics.

Algorithm	Mean Rank	Total †/§/≈
MOPSO	5.33	16/0/2
NSGA-II	4.11	15/2/1
MOEA/D	3.67	15/0/3
MOEA/D-AWA	2.94	13/0/5
MOEA/D-DRA	3.38	13/2/3
MOEA/D-NRSA	1.55	–

When the advantages of IGD and DM metrics are apparent, the GD metrics can be maintained at a high level.

Fig. 8 shows the comparison of IGD box plots of different algorithms in 6 instances. These box plots cover the mean value and variation interval of the IGD metric values obtained by 6 algorithms in a total of 150 simulations in 6 examples. By comparison, we can find that the change range of IGD of MOEA/D-NRSA is relatively tiny in the medium-scale and large-scale problems. Only MOEA/D-AWA has less fluctuations in large-scale issues than MOEA/D-NRSA. It shows that the algorithm's stability is better, and it is easier to obtain ideal results during operation.

The mean rank and the total count of †/§/≈ presented in Table 7 shows the statistical results of the three metrics, MOEA/D-NRSA ranks first among all algorithms. NSGA-II has the same performance as MOEA/D, and they have certain advantages in solving small-scale and large-scale problems, respectively. MOEA/D-AWA ranks second with its excellent performance in IGD and GD metrics, and its GD metrics are even better than MOEA/D-NRSA on medium-scale issues.

V. CONCLUSION AND FUTURE WORK

This paper mainly studies the modeling and solving methods of heterogeneous weapon platforms' cooperative task assignment problem. Firstly, we propose a bi-objective optimization model that can describe the multi-stage weapon-target assignment problem. The model considers the probability of damage, ammunition consumption, and the loss of the platform itself. Secondly, to effectively solve the model, the MOEA/D-NRSA method obtained by improving the MOEA/D framework is proposed. In MOEA/D-NRSA, the convergence and distribution of the solution are considered at the same time. In terms of convergence, a population evolution mechanism based on niche technology is proposed. This mechanism can make better use of neighborhood

information and spontaneously select suitable parents and produce offspring through the sharing of individuals.

On the other hand, the introduction of the INT method achieves a better approximation to the Pareto front and maintains the diversity of solutions. Finally, the simulation results show the advantages of MOEA/D-NRSA over other comparative algorithms. It does not perform well on small-scale problems, but on medium-scale and large-scale problems, the distribution of MOEA/D-NRSA is better than other algorithms. We know that each solution represents a kind of offensive plan in the objective space, and the better distribution of the solution on the Pareto front is equivalent to obtaining a more different plan. The advantage of MOEA/D-NRSA in terms of distribution proves that more decision-makers can use it to obtain offensive schemes that meet their preferences.

Judging from the current research results, the dynamics, uncertainty, and coordination of the actual combat process have not been well described in the model. The multi-stage model designed in this paper is only a product between SWTA and DWTA, and cannot fully describe the dynamics of confrontation. Regarding the uncertainty in the battlefield, this paper only caters to this feature by randomly generating some parameter values in a specific interval, rather than describing and processing uncertain factors through models. The limitations mentioned above in this paper lead to the direction of our future research. In terms of problem models, we can change our thinking and design models with individuals rather than timelines. The timeline can be used as a standard for sorting out the logic of actions. This modeling method can control the actions of each individual and simulate the dynamics of the entire combat area. In dealing with uncertain factors, it is necessary to find a quantifiable parameter to describe them and design a treatment method for uncertain factors. Finally, it is possible to study the specific collaborative actions between individuals in terms of collaboration instead of describing the collaboration through simple numerical stacking or some connection relationships between individuals.

REFERENCES

[1] R. K. Ahuja, A. Kumar, K. C. Jha, and J. B. Orlin, "Exact and heuristic algorithms for the weapon-target assignment problem," *Oper. Res.*, vol. 55, no. 6, pp. 1136–1146, Dec. 2007.

[2] S. P. Lloyd and H. S. Witsenhausen, "Weapons allocation is NP-complete," in *Proc. Summer Comput. Simul. Conf.*, 1986, pp. 1054–1058.

[3] A. Kline, D. Ahner, and R. Hill, "The weapon-target assignment problem," *Comput. Oper. Res.*, vol. 105, pp. 226–236, May 2019.

[4] A. S. Manne, "A target-assignment problem," *Oper. Res.*, vol. 6, no. 3, pp. 346–351, 1957.

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

[6] O. Karasakal, "Air defense missile-target allocation models for a naval task group," *Comput. Oper. Res.*, vol. 35, no. 6, pp. 1759–1770, Jun. 2008.

[7] A. G. Kline, D. K. Ahner, and B. J. Lunday, "Real-time heuristic algorithms for the static weapon target assignment problem," *J. Heuristics*, vol. 25, no. 3, pp. 377–397, Jun. 2019.

[8] S. Zhihua, F. Zhu, and Z. Duolin, "A heuristic genetic algorithm for solving constrained weapon-target assignment problem," in *Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst.*, vol. 1, Nov. 2009, pp. 336–341.

[9] Z. R. Bogdanowicz, A. Tolano, K. Patel, and N. P. Coleman, "Optimization of weapon–target pairings based on kill probabilities," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1835–1844, Dec. 2013.

[10] W. Yanxia, Q. Longjun, G. Zhi, and M. Lifeng, "Weapon target assignment problem satisfying expected damage probabilities based on ant colony algorithm," *J. Syst. Eng. Electron.*, vol. 19, no. 5, pp. 939–944, Oct. 2008.

[11] G. Shang, "Solving weapon-target assignment problems by a new ant colony algorithm," in *Proc. Int. Symp. Comput. Intell. Design*, vol. 1, Oct. 2008, pp. 221–224.

[12] R. R. Hill and E. A. Pohl, "Heuristics and their use in military modeling," in *Wiley Encyclopedia of Operations Research and Management Science*. Hoboken, NJ, USA: Wiley, 2010.

[13] A. Tokgöz and S. Bulkan, "Weapon target assignment with combinatorial optimization techniques," *Int. J. Adv. Res. Artif. Intell.*, vol. 2, no. 7, pp. 39–50, 2013.

[14] Y. Zhou, X. Li, Y. Zhu, and W. Wang, "A discrete particle swarm optimization algorithm applied in constrained static weapon-target assignment problem," in *Proc. 12th World Congr. Intell. Control Automat. (WCICA)*, Jun. 2016, pp. 3118–3123.

[15] B. Xin, J. Chen, Z. Peng, L. Dou, and J. Zhang, "An efficient rule-based constructive heuristic to solve dynamic weapon-target assignment problem," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 41, no. 3, pp. 598–606, May 2011.

[16] B. Xin, J. Chen, J. Zhang, L. Dou, and Z. Peng, "Efficient decision makings for dynamic weapon-target assignment by virtual permutation and tabu search heuristics," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 6, pp. 649–662, Nov. 2010.

[17] C. Leboucher, H.-S. Shin, P. Siarry, R. Chelouah, S. L. Ménec, and A. Tsourdos, "A two-step optimisation method for dynamic weapon target assignment problem," *Recent Advances on Meta-Heuristics and Their Application to Real Scenarios*. Rijeka, Croatia: InTech, 2013, pp. 109–129.

[18] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, Mar. 2011.

[19] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5. Boston, MA, USA: Springer, 2007.

[20] B. Xin, Y. Wang, and J. Chen, "An efficient marginal-return-based constructive heuristic to solve the sensor–weapon–target assignment problem," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 12, pp. 2536–2547, Dec. 2019.

[21] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genet. Algorithms Appl.* Hillsdale, NJ, USA: Lawrence Erlbaum, 1985, pp. 93–100.

[22] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[23] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and AntColony," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1845–1859, Dec. 2013.

[24] J. Chen, J. Li, and B. Xin, "DMOEA-εC: Decomposition-based multiobjective evolutionary algorithm with the ε-constraint framework," *IEEE Trans. Evol. Comput.*, vol. 21, no. 5, pp. 714–730, Feb. 2017.

[25] X. Cai, Z. Mei, Z. Fan, and Q. Zhang, "A constrained decomposition approach with grids for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 564–577, Aug. 2018.

[26] Y. Li, Y. Kou, and Z. Li, "An improved nondominated sorting genetic algorithm III method for solving multiobjective weapon-target assignment part I: The value of fighter combat," *Int. J. Aerosp. Eng.*, vol. 2018, pp. 1–23, Jun. 2018.

[27] S. Ding, C. Chen, Q. Zhang, B. Xin, and P. M. Pardalos, *Metaheuristics for Resource Deployment under Uncertainty in Complex Systems*, 1st ed. Boca Raton FL, USA: CRC Press, 2021.

[28] K. Miettinen, *Nonlinear Multiobjective Optimization*, vol. 12. Boston, MA, USA: Springer, 1998,

[29] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[30] D. E. Goldberg, J. Richardson, and Others, "Genetic algorithms with sharing for multimodal function optimization," in *Genetic Algorithms and Their Applications Proceedings of the Second International Conference on Genetic Algorithms*. Hillsdale, NJ, USA: Lawrence Erlbaum, 1987, pp. 41–49.

[31] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A Niched Pareto genetic algorithm for multiobjective optimization," in *Proc. 1st IEEE Conf. Evol. Comput. IEEE World Congr. Comput. Intell.*, Jun. 1994, pp. 82–87.

[32] L. Juan, C. Jie, and X. Bin, "Efficiently solving multi-objective dynamic weapon-target assignment problems by NSGA-II," in *Proc. 34th Chin. Control Conf. (CCC)*, Jul. 2015, pp. 2556–2561.
[33] Q. Zhang, H. Li, D. Maringer, and E. Tsang, "MOEA/D with NBI-style Tchebycheff approach for portfolio management," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8.
[34] Z. Fan, Y. Fang, W. Li, X. Cai, C. Wei, and E. Goodman, "MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems," *Appl. Soft Comput.*, vol. 74, pp. 621–633, Jan. 2019.
[35] S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 421–437, Feb. 2016.
[36] W. Xu, C. Chen, S. Ding, and P. M. Pardalos, "A bi-objective dynamic collaborative task assignment under uncertainty using modified MOEA/D with heuristic initialization," *Expert Syst. Appl.*, vol. 140, Feb. 2020, Art. no. 112844.
[37] S. Ding, C. Chen, B. Xin, and P. M. Pardalos, "A bi-objective load balancing model in a distributed simulation system using NSGA-II and MOPSO approaches," *Appl. Soft Comput.*, vol. 63, pp. 249–267, Feb. 2018.
[38] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, Jun. 2014.
[39] W. Khan and Q. Zhang, "MOEA/D-DRA with two crossover operators," in *Proc. U.K. Workshop Comput. Intell. (UKCI)*, Sep. 2010, pp. 1–6.
[40] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[41] J. Li, B. Xin, P. M. Pardalos, and J. Chen, "Solving bi-objective uncertain stochastic resource allocation problems by the CVaR-based risk measure and decomposition-based multi-objective evolutionary algorithms," *Ann. Oper. Res.*, vol. 296, nos. 1–2, pp. 639–666, Jan. 2021.
[42] S. Zapotecas-Martínez, B. Derbel, A. Liefooghe, D. Brockhoff, H. E. Aguirre, and K. Tanaka, "Injecting CMA-ES into MOEA/D," in *Proc. Annu. Conf. Genetic Evol. Comput.*, Jul. 2015, pp. 783–790.
[43] M. Afzalirad and J. Rezaeian, "A realistic variant of bi-objective unrelated parallel machine scheduling problem: NSGA-II and MOACO approaches," *Appl. Soft Comput.*, vol. 50, pp. 109–123, Jan. 2017.
[44] C. R. Raquel and P. C. Naval, "An effective use of crowding distance in multiobjective particle swarm optimization," in *Proc. Conf. Genetic Evol. Comput. (GECCO)*, Jun. 2005, pp. 257–264.

XIAOCHEN WU received the B.S. degree in automation and the M.Eng. degree in control engineering from the Beijing Institute of Technology, Beijing, China, in 2013 and 2016, respectively, where he is currently pursuing the Ph.D. degree in control science and engineering.

His current research interests include multi-route planning, damage assessment, dynamic self-organization of agents, and multi-objective optimization.

CHEN CHEN received the B.S. degree in automation and the Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2004 and 2009, respectively.

She is currently a Professor with the School of Automation, Beijing Institute of Technology. Her current research interests include complicated systems, multi-objective optimization, and distributed simulation.

SHUXIN DING received the B.E. degree in automation and the Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2012 and 2019, respectively.

From 2016 to 2017, he was a Visiting Scholar of industrial and systems engineering with the University of Florida, Gainesville, FL, USA. He is currently an Assistant Researcher with the Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Ltd. His current research interests include railway scheduling, evolutionary computation, multi-objective optimization, and optimization under uncertainty.