IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 18, 2019, accepted April 3, 2019, date of publication April 11, 2019, date of current version April 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2910241

Multi-Objective Weapon Target Assignment Based on D-NSGA-III-A

CHUNQING GAO[1], **YINGXIN KOU**[1], **YOU LI**[1], **ZHANWU LI**[1], **AND AN XU**[1,2]
[1]Aeronautics Engineering College, Air Force Engineering University, Xi'an 710038, China
[2]College of Electronic Communication, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: You Li (firecaf@163.com)

ABSTRACT The multi-objective weapon-target assignment problem, which aims to generate reasonable assignment to meet the objectives, is a typical optimization problem with complex constraints. In order to get close to the actual air combat, the game process between both sides at war is introduced to construct a three-objective mathematical model, which includes the damage of the enemy, the cost of missiles, and the damage value of fighting capacity. Considering the NP-complete nature of multi-objective weapon-target assignment problem, an improved intelligent algorithm (named as D-NSGA-III-A) on the basis of non-dominated sorting genetic algorithm III (NSGA-III) is proposed. In this improved algorithm, first, the non-dominated sorting based on dominance degree matrix is proposed to reduce the unnecessary or repetitive comparisons in ranking schemes, so as to further decrease the time consumption. Second, diversity and convergence are taken into account resorting to the niching information and the dominance ratio when selecting individuals. Third, the adaptive operator selection mechanism, which selects the operators adaptively according to the information of generations from a pool where single point crossover and all bits crossover operators are included, is employed to seek a balance between intensification and diversification within the decision space and to improve the quality of Pareto solutions. From the experiments, the combination of above technologies obtains better Pareto solutions and time performance for solving the static multi-objective target assignment (SMWTA) problem than NSGA-III, MP-ACO, NSGA-II, MOPSO, MOEA/D, and DMOEA-εC.

INDEX TERMS Adaptive operator selection mechanism, dominant degree matrix, multi-objective optimization, non-dominated sorting genetic algorithm III, weapon target assignment.

I. INTRODUCTION

With the rapid development of technology and weapon systems, the air combat mode has been changing a lot. However, the final step to win the combat is always attacking the enemy targets with all available weapon systems. Thus, the Weapon Target Assignment (WTA) problem will never fade away [1]–[5]. The WTA problem is a combinatorial optimization problem, which attempts to find an optimal assignment solution meeting all constraints and maximizing military effectiveness.

From the 1950s, the WTA problem has been studied [1]. In 1990, Hosein and Athans [6], considering the quantity of objective functions, classified the WTA problem into single-objective weapon target assignment problem (SWTA) and multiple-objective weapon target assignment (MWTA) problem. Obviously, the latter is closer to the actual combat. In 2007, Galati and Simaan [7], considering the time factor, divided the WTA problem into dynamic weapon target assignment problem and static weapon target assignment problem. More details see Section 2 of the paper.

Here, the static multi-objective target assignment (SMWTA) problem attracted our attention. To our knowledge, objectives of the SMWTA problem presented in most literatures was limited to the expected damage of the enemy and the cost of missiles. However, the real combat is a game process. When we attack the enemy, we would also be counterattacked [8], [9]. The objective functions of traditional the SMWTA problem did not include the damage value of fighting capacity. Therefore, taking the game process into account, we have improved the original model [10] with a new objective - the damage value of fighting capacity.

Compared with the traditional the SMWTA problem, our three objective functions model has more constraints

The associate editor coordinating the review of this manuscript and approving it for publication was Geng-Ming Jiang.

and it is more complex. Apparently, it belongs to the NP-complete problem and there is no effective deterministic method for it. The traditional algorithms, such as goal programming method [11], minimax method [12], linear weighted method [13], are not suitable for the SMWTA problem because of the extremely time and resource consuming. Actually, most of Multi-Objective Evolutionary Algorithms (MOEAs), such as MOEA/D [14], PESA-II [15], SPEA2 [16], NSGA-II [17], still face different difficulties [18]: i) the convergence of the algorithm is not ensured, ii) the population diversity is insufficient, and iii) the dilemma of balancing the exploration and exploitation.

Recently, the trend of EMOAs mainly focuses on Pareto dominance. The Non-dominated Sorting Genetic Algorithm III (NSGA-III) [19], a latest and promising algorithm to deal with the above problem, is one of the representatives. Compared with NSGA-II, NSGA-III remains its basic framework but has significant difference in the selection mechanism, where NSGA-III replace the crowding distance with reference points. This change of the selection mechanism can contribute to performing well on large dimension problems. From the literature, NSGA-III has been applied to automating software refactoring [20], water resource management [19], integrating alternative domination schemes [21], et al. To our knowledge, there is no research on NSGA-III for the SMWTA problem.

Review our three objective functions: i) the damage of the enemy, ii) the cost of missiles, and iii) the damage value of fighting capacity. To expect the maximizing damage of the enemy and the minimizing cost of missiles, the adopted algorithm must promise the optimal Pareto solutions to obtain optimum assignment. In addition, the damage value of fighting capacity means the enemy would counterattack the fighter, so much so that, we must make a decision and destroy the target in a very short time. In a word, the adopted algorithm should get better Pareto solutions while consume less time. Considering the characteristic of the SMWTA problem and the advantage of NSGA-III algorithm, we improve the algorithm (named as D-NSGA-III-A algorithm) to meet the requirements thus make it more suitable for the problem.

The contributions in the present paper are mainly about two aspects. One hand is about the model of SMWTA problem. Taking the game process into account, we have improved the two-objective model to three-objective model with a new objective - the damage value of fighting capacity. On the other hand is about the algorithm to solve the SMWTA problem. i) Pareto domination is replaced with dominance degree matrix for non-dominated sorting. Thereby, the unnecessary or repetitive comparisons in ranking schemes can be decreased to further decrease the time consumption. ii) The diversity and convergence are both took into account when selecting individuals. Here, the diversity of individuals comes from the niching information while the convergence comes from the dominance ratio. iii) Substituting the simulated binary crossover operator with adaptive operator selection (AOS) mechanism. The AOS mechanism selects the single point crossover operator or all bits crossover operators adaptively according to the information of generations. Above all, the proposed algorithm could seek a balance between intensification and diversification within the decision space and improve the quality of Pareto solutions.

The paper is organized as follows: Section II illustrates the related work on the WTA problem. The assumptions and mathematical model are completed in Section III. In Section IV is the proposed D-NSGA-III-A algorithm, with detailed improvements on NSGA-III algorithm. In Section V are accomplished experiment results, with a comprehensive discussion on performance. Finally, Section 6 concludes the paper together with an outlook to the future work.

II. RELATED WORK

The WTA problem, since the first model was built in the 1950s [1], continues to attract attention of the operation research and military-oriented simulation owing to its importance [22]. As the research moves along, Hosein and Athans in 1990 [6], considering the quantity of objective functions, classified the WTA problem into single-objective or multiple-objective problem. Galati and Simaan in 2007 [7], considering the time factor, divided the WTA problem into dynamic or static problem.

In modern and future battlefields, the benefit of assigning smart weapons to targets particularly rely on the pre-assigned sensors. Moreover, the coordination of sensors and weapons plays a critic role to obtain the best fighting effect. In 2007, the sensor-weapon-target assignment (S-WTA) problem, was first put forward by Bogdanowicz and Coleman [23]. when solving smart weapon assignment problem, considering the sensor-target assignment problem at the same time. After that, more researchers focused on the problem adopting all kinds of algorithms [24]–[27]. Naturally, the S-WTA problem, which is to find the best sensor-target and weapon-target pairings, is also one of the WTA problem [24].

Moreover, the latest researches on WTA problem from this century are shown in Table 1. Obviously, the trend of research is from static to dynamic, from single-objective to multi-objective. All models are important because in the real world there may be cases for which one objective is sufficient, and others that require models with more objectives. Apparently, the multi-objective model considers more elements and factors than single-objective model.

Along with the deepening of research, WTA was proved a NP-complete problem [42] that the calculating time will grow exponentially for WTA optimal solution in pace with the increasing problem scale. Moreover, the most difficulties are caused by the nonlinear nature of the WTA problem and the complexity of the input. [42], [43]. Therefore, lots of intelligent optimization algorithms have been used to solve this problem, such as particle swarm optimization (PSO), genetic algorithm (GA), ant colony algorithm (ACO), and so on, as shown in Table 1. Most of them did some adjustments or improvements on the foundation of the original

TABLE 1. **The latest researches on WTA problem.**

WTA	Static	Dynamic
Single-objective	2002/2002/2003, Lee Z J et al., IS+ACO [2]/ GA [28] /GA+ACO [29] 2007/2009/2013, Bogdanowicz Z R et al. Auction algorithm [23]/ Swt-opt algorithm [24], GA [30] 2010, Lee M Z [31], VLSN 2010, Chen P et al. [32], SA 2012, Fei A G et al. [33], Auction algorithm 2012, Chen H D et al. [26] GA 2015, Wang J et al. [27], improved GA 2016, Liang H T et al. [34], CSA 2018, Xin B et al. [25], Marginal-Return-Based constructive heuristic	2007, Galati D G et al. [7], Tabu 2010, Li Y et al. [35], DPSO+SA 2010, Xin B et al. [3], VP+Tabu
Multi-objective	2013, Liu X et al. [36], MOPSO 2014, Zhang Y et al. [4], MOEA/D 2015, Li J et al. [37], NSGA-II, MOEA/D 2017, Li Y et al. [10], MPACO 2017, Li J et al [38]. DMOEA-εC	2011, Xin B et al. [39], Rule-based heuristic 2015, Ahner D K et al. [40], Dynamic programming 2015, Dirik N et al. [41], MILP 2016, Li N et al. [5], MDE

algorithms to obtain the near optimal solution and to satisfy the real-time requirement for WTA decision.

Let us focus on intelligent algorithms for the SMWTA problem that interests us. In literature [36], Liu *et al.* modified the velocity and position update formulas of PSO algorithm to adapt the multi-objective WTA problem. However, there are only 7 platforms and 10 targets included in the given example. In literature [4], Zhang *et al.* proposed an appropriate decomposition-based MOEA with a repair method, which was simulated to improve the convergence and spacing performance. Nevertheless, this method may not suitable for the actual combat because of the low convergence speed. In literature [37], Li et al. developed an adaptive mechanism to enhance the NSGA-II and MOEA/D thus achieving efficient problem solving. Then, four performance metrics were adopted to compare the performance between the adaptive NSGA-II and adaptive MOEA/D on solving instances of three scales MWTA problems. In literature [10], Li *et al.* proposed and embed five improve strategies into traditional P-ACO algorithm, and simulations of different scales of instances indicated that the improved algorithm produced better solution and consumed less time than NSGA-II and SPEA-II. In literature [38], Li *et al.* improved the MOEA/D with a ε-constraint to solve the two-objective WTA problem. Then, comparison studies among the proposed DMOEA-εC, NSGA-II, and MOEA/D-AWA on solving three different-scale MWTA instances were done. The numerical experiment demonstrated that DMOEA-εC shows advantages on solving the large-scale instance while NSGA-II performs best on small-scale and medium-scale instances.

In 2014, Deb and Jain [19] proposed NSGA-III, an upgrade version of NSGA-II, which seems more efficient for solving multi-objective WTA problems. In briefly, NSGA-III remains the basic frame of NSGA-II, but replaces the crowding distance with reference points. In addition, the number of reference points is close to the population size, which can ensure that individual of the population member is associated with a reference point. Moreover, the reference points are uniformly distributed on normalized hyper-planes, which help maintain the diversity. The above special designs make it perform better than MOEA/D and NSGA-II.

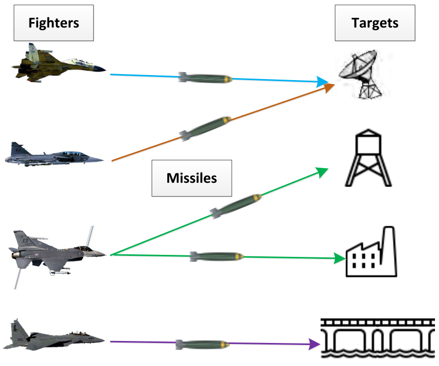


FIGURE 1. **An example of WTA problem.**

Recent years, many achievements have been made in the research of NSGA-III algorithm. In 2014, Jain and Deb [44] proposed an A-NSGA-III to adaptively add or remove the reference points according to the degree of population crowding on the non-dominated Pareto front. Considering the limitation of this algorithm [45], the same authors later developed a more efficient A-NSGA-III, namely A2-NSGA-III, which improve the reposition strategy of reference points. Still in 2014, Yuan *et al.* [21], [46] developed a θ-NSGA-III, which define a new θ-dominant to maintain the convergence and diversity. In 2015, Seada and Deb [47] came up with U-NSGA-III for solving three classes of problems.

The previous researches show that there are two kinds of technologies generally to balance the convergence and diversity. One is defining a new preference relationship [21], [48], [49], and the other is choosing a better operator selection mechanism [19], [50]. In this paper, the D-NSGA-III-A combining both above two technologies is proposed to solve the three-objective SWTA problem.

III. PROBLEM FORMULATION

The WTA problem is a proper assignment of various weapons to different targets according to our combat purposes, as shown in Fig. 1. In order to get closer to the actual combat, a mathematical model is built with three objectives. The detailed descriptions are in the following subsections.

A. ASSUMPTION DESCRIPTION

To obtain a reasonable mathematical model and convenient to draw the consequence, the assumptions are defined as follows:

Assumption 1: There are fighters (F), missiles (M) and targets (T). Based on the actual combat, each fighter can carry different kinds of missiles, but the number of missiles is no more than 4.

Assumption 2: The number of fighters and the number of targets are not necessarily equal. Also, one target can be attacked by different missiles but one missile can only attack one target.

Assumption 3: If the target is in the launch range of the missile, the missile can be effectively assigned, or is not.

B. MATHEMATICAL MODEL

In this paper, minimizing damage value of fighting capacity is introduced as the third objective to construct a SMWTA model on the foundation of two familiar objectives, the maximizing expected damage and the minimizing missile consumption. The model definitions and constraints are shown as follows.

1) THREE-OBJECTIVE MODEL

In the course of military operations, each side struggles to preserve itself and destroy the other. Considering the counterattack of the enemy defense system, our fighters may be attacked. Therefore, the new objective - minimizing damage value of fighting capacity - is proposed to measure our losses, so that we can decide if it is worth carrying out the mission. The new objective is presented as follows:

$$f_3 = \min \sum_{k=1}^{s} \sum_{i=1}^{m} \sum_{j=1}^{n} v_k \left(\frac{u_j}{r_i} x_{ij} \right) \tag{3.1}$$

where v_k represents the value of fighter k. The more important fighter k for our fleet, the larger value of v_k is. r_i represents the optimum launch range index of missile i. The larger value of r_i indicates the longer optimum launch distance for missile i, and vice versa. u_j represents the target defense value. If fighter k attacks target j, the defense system will counterattack fighter k at a certain probability. For example, the more important target j for the enemy is, the larger value of u_j is, and the bigger probability of the fighter being counterattacked is. Thereby, the fighter is less likely to preserve itself. At the same time, the shorter optimum launch distance means our fighter is closer to the enemy defense system, so the fighter is more likely to be counterattacked and has lower probability to preserve itself.

The maximizing expected damage objective is shown in formula (3.2).

$$f_1 = \max \sum_{j=1}^{n} \left[1 - \prod_{i=1}^{m} \left(1 - \rho_k q_{ij}\right)^{x_{ij}} \right] \tag{3.2}$$

where q_{ij} represents the damage probability of missile i attacking target j. $0 < q_{ij} < 1$, $i = 1, 2, \cdots\cdots, m$, $j = 1, 2, \cdots\cdots, n$. m and n denote the number of missiles and targets, respectively. $\rho_k \in (0, 1)$ is the pilot operation factor of fighter k [10], which may affect the maximizing expected damage.

The minimizing missile consumption objective is shown in formula (3.3).

$$f_2 = \min \sum_{i=1}^{m} \sum_{j=1}^{n} c_i x_{ij} \tag{3.3}$$

where $x_{ij} = \{0, 1\}$ represents the assignment decision of missile i and target j. If $x_{ij} = 1$, it means missile i of fighter k is assigned to target j. On the other hand, it means missile i of fighter k is not assigned to target j.

c_i is a positive constant, which means the cost of missile i.

2) CONSTRAINTS AND PROCESSING

Constraints of functions are shown in following formulas:

$$\sum_{j=1}^{n} x_{ij} \leq 1 \tag{3.4}$$

$$1 \leq \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \leq m \tag{3.5}$$

(3.4) is quantity assignment constraint, each missile can only attack only one target, which is promised by coding.

(3.5) is parameters constraint, the total assignment is greater than one but does not exceed the total number of missiles.

IV. PROPOSED ALGORITHM: D-NSGA-III-A

A. INTRODUCTION OF NSGA-III

NSGA-III is an upgrade version of NSGA-II, and they have the similar basic frame but significant difference in the selection mechanism [19]. The details of NSGA-III are as follows.

First, the NSGA-III algorithm initializes a population whose size is N, and defines a series of reference points using Das and Dennis's systematic approach that places points on a normalized hyper-plane. Moreover, the hyper-plane is equally inclined to all objective axes and has an intercept of one on each axis. Considering the M-objective problem, the normalized hyper-plane would be a $(M - 1)$ dimensional unit simplex. And if p divisions are given, the total number of reference points H can be represented as follows:

$$H = \binom{M + p - 1}{p} \tag{4.1}$$

Then, let us consider t-th generation of this algorithm. The parent population is P_{t}, while the offspring population Q_{t} is created from P_{t} by simulated binary crossover and mutation. The P_{t} and Q_{t} both have N members. The next step is to preserve elite members (of size N) from the combination population (of size $2N$) of parent and offspring population, namely $R_t = P_{\mathrm{t}} \cup Q_{\mathrm{t}}$. To achieve this, a non-dominated sorting based on Pareto domination is used to sorted R_t to different non-domination levels (F_1, F_2, and so on).

Thereafter, a new population S_{t} is constructed by individuals, which are selected starting from F_1 level but end until the size of $S_t \geq N$ for the first time (suppose l -th level). Thus, total selected members are $P_{t+1} = \cup_{i=1}^{l-1} F_i$ from one to $(l-1)$ fronts. Thereafter, remaining $(K = N - |P_{t+1}|)$ individuals of the l -th front are chosen to maximize the diversity. When the size of population P_{t+1} is N, the unselected solutions are rejected from population R_t. Compared with NSGA-II, NSGA-III here replaces the crowding distance with reference directions-based niching.

The details of references-based niching are as follows.

First, the objective values and reference points are normalized as follows. First, the ideal point of the population S_t is determined by constructing the ideal point $\bar{z} = \left(z_1^{\min}, z_2^{\min}, \cdots\cdots, z_M^{\min}\right)$, where $z_m^{\min}$ represents the minimum value. The translated objective $f_m^{'}(x)$ can be obtained by subtracting each objective value $f_m(x)$ $(m = 1, 2\cdots, M$, $x \in S_t$) by $z_m^{\min}$. Namely

$$f_m^{'}(x) = f_m(x) - z_m^{\min} \tag{4.2}$$

Thereafter, the extreme point value in corresponding objective axis is identified by finding the minimum solution of function $ASF(x, w)$.

$$ASF(x, w) = \max_{m=1}^{G} f_m^{'}(x) / w_m \tag{4.3}$$

where w_m is the weight vector of corresponding objective, and if $w_m = 0$, we redefine $w_m = 10^{-6}$. For each $f_m^{'}(x)$ $(m = 1, 2\cdots, M, x \in S_t)$, there will be an extreme objective vector, all of which can be used to constitute a M -dimensional linear hyper-plane. The normalized objective functions $f_m^n(x)$ can be normalized as follows.

$$f_m^n(x) = f_m^{'}(x) / \left(a_m - z_m^{\min}\right) \tag{4.4}$$

where a_m represents the intercept on corresponding normalized objective axis.

After normalizing, we define reference lines from the origin to each reference point. Then, we find the closet perpendicular distance of individuals of population S_t from reference lines to associate the reference point with the population member.

At last, the reference-point-associated individuals, the number of which from $P_{t+1} = S_t / F_l$ connected to the k-th reference point is α_k, are chosen by niche-preservation strategy. First, the reference point that has minimum α_k is selected into the set $J_{\min} = \{k : \arg\min_k \alpha_k\}$. When $|J_{\min}| > 1$, one $(\bar{k} \in J_{\min})$ is chosen at random. For case $\alpha_{\bar{h}} = 0$, if there is at least one individual associated with k-th reference point, one having the shortest perpendicular distance will be added to P_{t+1}. Then, the count $\alpha_{\bar{h}}$ will be incremented by one, otherwise not. For case $\alpha_{\bar{h}} \geq 1$, a randomly chosen individual is associated with k-th reference point is added to P_{t+1}, the count $\alpha_{\bar{h}}$ is incremented by one. Finally, update the niche counts and repeat the procedure to fill population P_{t+1}.

B. THE NON-DOMINATED SORTING BASED ON DOMINANCE DEGREE MATRIX

From previous research on non-dominated sorting, the ranking schemes are of high computational complexity. One reason is that there are lots of unnecessary or repetitive comparisons, the other is that vector ranking comparisons are more difficult to deal with than scalar ranking comparisons. To further decrease the time consumption, it is significant to decrease the time of comparisons or to make the comparisons more efficient. Here, a non-dominated sorting based on dominance degree matrix is proposed to replace the non-dominated sorting based on Pareto domination of NSGA-III.

First, let us recall the Pareto domination. Define two objective vectors $A = [a_1, a_2, \cdots a_m]^T \in R^m$ and $B = [b_1, b_2, \cdots b_m]^T \in R^m$. A is claimed to dominant B, if and only if there is $\forall i \in (1, 2, \cdots, m)$, $a_i \leq b_i$, and there is at least one $\forall j \in (1, 2, \cdots, m)$, $a_i < b_i$. In addition, A is called the Pareto optimal solution if there is no objective vector to dominant it in the objective space.

1) PREPARATIONS FOR DOMINANCE DEGREE MATRIX

As above, the Pareto optimal solution can be obtained by researching the relation between element of A and B. Therefore, we can guess that it is also meaningful to count the number of element pairs (a_i, b_i) that satisfy $a_i \leq b_i$. In addition, the dominance degree is defined to represent the dominance strength from A to B.

Definition 1 (Dominance Degree):

$$d(A, B) = |\{i | i \in (1, 2, \cdots, m), a_i \leq b_i\}| \tag{4.5}$$

where $|\cdot|$ denotes the cardinality of the set. It is easy to understand that $0 \leq d(A, B) \leq m$, and when A is claimed to dominate B, $d(A, B) = m$. In addition, when $d(A, B) = d(B, A) = m$, then $A = B$.

Define a set that contains N objective vectors $R = \left\{A^1, A^2, \cdots A^N\right\}$, $A^i = [a_{1i}, a_{2i}, \cdots a_{mi}]^T \in R^m$, $i \in \{1, 2, \cdots, N\}$. Thus, dominance degree matrix on set R can be defined.

Definition 2 (Dominance Degree Matrix):

$$D = \left(d_{ij}\right)_{N \times N} \tag{4.6}$$

where d_{ij} is the dominance degree $d_{ij} = d(A^i, A^j)$, and i, j meet $i, j \in \{1, 2, \cdots, N\}$.

2) CALCULATE THE DOMINANCE DEGREE MATRIX

According to the definition, the dominance degree matrix can be calculated with complexity $\Theta\left(mN^2\right)$. To decrease the computation, Quicksort [51], whose complexity is also $\Theta\left(mN^2\right)$ but $\Theta(mN \log N)$ on average, is adopt to calculate the dominance degree matrix. First, Quicksort is adopted to sort the set of objective vectors on each objective. After sorting, the dominance degree matrix can be constructed with the help of comparison matrix. The details are described below.

First, considering a row vector $W = (w_1, w_2, \cdots w_N) \in R^N$, we define a comparison matrix.

Definition 3 (Comparison Matrix):

$$C_W = \left(cw_{ij}\right)_{N \times N} \tag{4.7}$$

If $w_i \leq w_j$, then $cw_{ij} = 1$, otherwise $cw_{ij} = 0$. According to the property of comparison matrix, if $w_i = w_j$, the elements of i-th rows are same as the elements of j-th rows in comparison matrix C_W, which should be pay more attention to.

Obviously, a set of vectors R contains m objective vectors. Therefore, the dominance degree matrix can be obtained by summing the comparison matrices of all objective vectors. The pseudo codes are as below.

Algorithm 1 Calculate the Dominance Degree Matrix

```
Input: $R = \{A^1, A^2, \cdots A^N\}$, where $A^i = [a_{1i}, a_{2i}, \cdots a_{mi}]^T \in R^m, i \in \{1, 2, \cdots, N\}$.
Output: $D = (d_{ij})_{N\times N}$
1:  Construct an $m \times N$ matrix $A = (A^1, A^2, \cdots A^N)$, and define $A_i$ $(i = 1, \cdots, m)$ to represent its row vectors.
2:  $(d_{ij})_{N\times N} \leftarrow (0)_{N\times N}$
3:  for $i = 1 : m$ do
4:      $[X, Y] = \text{Quicksort}(A_i)$; //Ascending sort, $X$ is the sorted vector, and $Y$ is the index vector, and $X$ meet $X = A_i(Y)$.
5:      $(cw_{ij})_{N\times N} \leftarrow (0)_{N\times N}$; //Calculate the Comparison Matrix from line 5 to 20.
6:      for $j = 1 : N$ do
7:          $cw_{Y(1)j} = 1$;
8:      end for
9:      for $i = 2 : N$ do
10:         if $X(i) = X(i-1)$
11:             for $j = 1 : N$ do
12:                 $cw_{Y(i)j} = cw_{Y(i-1)j}$;
13:             end for
14:             else
15:                 for $j = i : N$ do
16:                     $cw_{Y(i)Y(j)} = 1$;
17:             end for
18:         end if
19:     end for
20:     $C_{A_i} = (cw_{ij})_{N\times N}$; //Generate Comparison Matrix.
21:     $D = D + C_{A_i}$;
22: end for
```

3) THE NON-DOMINATED SORTING BASED ON DOMINANCE DEGREE MATRIX

Let us consider the t-th generation of this algorithm. The combination population of parent and offspring population is R_t with size $2N$., and the dominance degree matrix is $D_{2N\times 2N}$. According to the Pareto dominance, the members have the same objective vectors would be sort into the same non-domination level. To eliminate the same representations of these members, the corresponding elements of D are set to zero. After that, define a $1 \times 2N$ row vector $Max\,(D)$, where the maximum elements from each column of D are included. According to **Definition 1**, if the elements in $Max\,(D)$ are less than m, the indices corresponding to solutions in R_t are non-dominated solutions, which constitute the first non-domination level F_1. Then, delete the row and column vectors corresponding to F_1 from D, and rename the rest of matrix by D_1. Finally, recycle the procedure to obtain F_2, F_3, and so on. Actually, when the size of assigned solutions is larger than N at first time, it is unnecessary to sort the remainder since there are enough solutions. The pseudo codes are as follows.

Algorithm 2 The non-Dominated Sorting Based on Dominance Degree Matrix

```
Input: Population $R_t$ with size $2N$.
Output: Fronts $F = \{F_1, F_2, \cdots\}$
Calculate the dominance degree matrix $D_{2N\times 2N}$
for $i = 1 : 2N, j = i : 2N$ do
    if $d_{ij} == m \&\& d_{ji} == m$
        $d_{ij} = 0$; $d_{ji} = 0$;
    end if
end for
$count = 0$; $k = 1$;
while 1
    $Q \leftarrow \emptyset$; $MD \leftarrow Max\,(D)$;
    for $i = 1 : 2N$ do
        if $MD(i) < m \&\& MD(i) >= 0$
            $Q \leftarrow Q \cup \{i\}$;
            $count + +$;
        end if
    end for
    for $i \in Q$ do
        Delete the elements of $i$-th row and $i$-th column in $D$
    end for
    $F_k \leftarrow$ The indices $Q$ corresponding to solutions in $R_t$; $k = k + 1$;
    if $count >= N$ break; end if//the size of assigned individuals is larger than $N$ at first time.
end while
```

C. ADAPTIVE OPERATOR SELECTION MECHANISM

Recalling the NSGA-III algorithm, its crossover operator only contains simulated binary crossover operator, which is adopted to generate offspring population. As is well known, the single crossover operator is difficult to balance the exploration and exploitation within the decision space for complex problems. Therefore, an adaptive operator selection mechanism, which can select operators adaptively from an operators' pool to adapt to the search landscape, is adopted to produce better solutions.

In this paper, the adaptive selection mechanism is based on the information collected by the credit assignment methods and probability methods using a roulette wheel-like process. Probability Matching (PM) is one of the famous probability methods for selecting an operator. The formulas to calculate the probability of selecting operator are as follows [52]:

$$p_{op}\,(t+1)=p_{\min}+(1-K\cdot p_{\min})\cdot\lambda_{op}\,(t+1)/\sum_{i=1}^{K}\lambda_i\,(t+1) \tag{4.8}$$

where $p_{op}\,(t+1)$ represents the probability of operator op is selected at next generation, $p_{\min}$ is the minimal probability of any operator, K is the number of operators and λ_{op} is the quality associated with operator op. Clearly, $\sum_{i=1}^{K}\lambda_i\,(t+1)=1$. From Eq. (4.8), if there is only one operator obtaining rewards, then $\lambda_{op}\,(t+1)/\sum_{i=1}^{K}\lambda_i\,(t+1)=1$, the maximum selection probability is $p_{\max}=p_{\min}+(1-K\cdot p_{\min})$.

The operators' pool here is composed of the well-known single point crossover and the proposed all bits crossover. From previous research, the single point crossover operator tends to exploit solutions near their parents while the all bits crossover tends to explore new solutions. Apparently, it is beneficial for generating solutions to combine these complementary behaviors. The proposed all bits crossover operators is as follows.

The all bits crossover operator takes the bitwise AND, OR and XOR as the crossover operations. By the proposal, all bits can be crossover thus the operator tends to explore new solutions. Furthermore, a crossover parameter σ_c is given to adjust the degree of crossover. If a random number $\varphi_c \in [0, 1]$ meet $\varphi_c \leq \sigma_c$, the crossover operation can be implemented. The formulas are as below:

$$\begin{cases} y_1 = x_1 \,\&\, x_2 \\ y_2 = x_1 | x_2 \\ y_3 = x_1 \wedge x_2 \end{cases} \tag{4.9}$$

where x_1, x_2 are two parents, and y_1, y_2, y_3 are three offspring solutions.

D. THE PROPOSED NSGA-III ALGORITHM

Based on the original NSGA-III algorithm, the proposed algorithm replaces Pareto domination with dominance degree matrix for non-dominated sorting. In addition, the simulated binary crossover operator is replaced with adaptive operator selection (AOS) mechanism. Here, the proposed algorithm is named as D-NSGA-III-A. The pseudo codes are shown in **Algorithm 3**.

In **Algorithm 3**, the differences between our algorithm and NSGA-III are highlighted with different colors.

In yellow parts, in step 4, the adaptive operator selection is associated with the probability of each operator. With the probability increasing by the success of an operator, the most suitable operator in the pool will be selected more often to improve the solutions. In step 26, the reward $e_{op}\,(t)$ is a Boolean value, which has a great significance on the probability of an operator op is selected at next generation. If an individual generated by operator op belong to P_{t+1} but not to P_t, then $e_{op}\,(t) = 1$. Otherwise, $e_{op}\,(t) = 0$.

Algorithm 3 D-NSGA-III-A Algorithm

```
Input: $P_0, I_{\max}, N, H$
Output: Pareto solutions
1:  Initialize reference points $H$;
2:  $t = 0$;
3:  while $t < I_{\max}$ do
4:      $op \leftarrow$ Adaptively select an operator ();
5:      $Q_t \leftarrow$ Create Offspring Population ($P_t$);
6:      $R_t \leftarrow P_t \cup Q_t$;
7:      $(F_1, F_2, \cdots, F_l)$ = The dominance-degree-matrix-based non-dominated sorting ($R_t$);
8:      $i = 1$; $S_t = \emptyset$;
9:      repeat
10:         $S_t = S_t \cup F_i$;
11:         $i = i + 1$;
12:     until $|S_t| \geq N$;
12:     $F_l = F_i$; //Last front to be included.
13:     if $|S_t| = N$ then
14:         $P_{t+1} = S_t$; break;
15:     else
16:         if $l = 1$ then
17:             Fill $P_{t+1}$ with $\mu N$ ($\mu \in [0, 1]$) solutions from $F_l$ according to the best $D\_ratio$
18:         else
19:             $P_{t+1} = \cup_{i=1}^{l-1} F_i$;
20:         end if
21:         Normalize the objectives;
22:         Associate ($S_t, H$); //Associate each member of $S_t$ with a reference point.
23:         Compute niche count of reference points;
24:         Fill $P_{t+1}$ with $(N - |P_{t+1}|)$ solutions from $F_l$ using niching information;
25:     end if
26:     Calculate operator rewards ($P_t, P_{t+1}$);
27:     Update operator information ();
28:     $t = t + 1$;
29: end while
30: return Pareto solutions.
```

In step 27, the qualities associated with operators are updated by

$$\lambda_{op}\,(t+1) = (1-\theta)\cdot\lambda_{op}\,(t) + \theta\cdot e_{op}\,(t) \tag{4.10}$$

where $\theta \in (0, 1]$ is the adaption rate. After that, the operator selection probabilities are updated by (4.8).

In green part, the non-dominated sorting based on dominance degree matrix is adopted to substitute the non-dominated sorting based on Pareto domination. In terms of time performance, the former only sort the necessary non-dominated fronts (namely $F_1, F_2, \cdots F_l$) while the latter try to sort all fronts (namely $F_1, F_2, \cdots F_l, \cdots$). Apparently, the former is more efficient.

In blue part, when the population size of first front is more than N, the selected individuals only consider the diversity in NSGA-III, but in the proposed algorithm, the diversity and convergence are both taken into account. Here, the diversity of individuals comes from the niching information while the convergence comes from the dominance ratio.

First, let us define dom_str_i, sup_num_i and inf_num_i as follows:

$$dom\_str_i = \sum_{j=1, j\neq i, d_{ij}=m}^{N} d_{ij} \tag{4.11}$$

$$sup\_num_i = \sum_{j=1, j\neq i}^{N} d_{ij} \tag{4.12}$$

$$inf\_num_i = \sum_{j=1, j\neq i}^{N} d_{ji} \tag{4.13}$$

where dom_str_i represents the number of individuals that are dominated by individual A^i multiplied by the number of objectives m, sup_num_i denotes the total number of objectives where individual A^i has superiority than others, and inf_num_i denotes the total number of objectives where individual A^i has inferior than others. From a superficial point of view, bigger the dom_str_i and sup_num_i are better while smaller the inf_num_i is better.

Then, normalize and combine the above three parameters, the dominance ratio D_ratio is came up with to evaluate the quality of the solutions at some level.

$$D\_ratio_i = \frac{dom\_str_i}{\sum_{i=1}^{N} dom\_str_i} + Cd_1 \frac{sup\_num_i}{\sum_{i=1}^{N} sup\_num_i} - Cd_2 \frac{inf\_num_i}{\sum_{i=1}^{N} inf\_num_i} \tag{4.14}$$

where, $Cd_1, Cd_2 \in [0, 1]$ are given scalar weight parameters. After computation, the bigger the D_ratio is, the more likely the individual is excellent. Thus, those individuals with a bigger dominance would be selected prior.

The flow chart of the proposed D-NSGA-III-A algorithm is shown in Fig. 2.

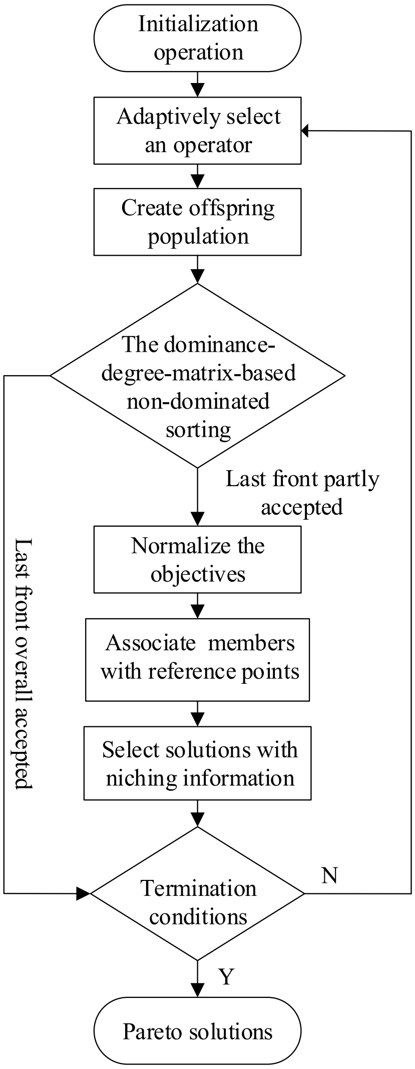

Initialization operation
Adaptively select an operator
Create offspring population
The dominance-degree-matrix-based non-dominated sorting
Last front partly accepted
Last front overall accepted
Normalize the objectives
Associate members with reference points
Select solutions with niching information
Termination conditions
N
Y
Pareto solutions

FIGURE 2. The flow chart of the proposed D-NSGA-III-A algorithm.

TABLE 2. Parameters settings of D-NSGA-III-A and NSGA-III.

	p_m	$I_{\max}$	$p_{\min}$	μ	Cd_1	Cd_2	K
D-NSGA-III-A	0.07	200	0.1	0.25	0.8	0.2	2
NSGA-III							

V. EXPERIMENT RESULTS AND ANALYSIS

A. PARAMETER SETTINGS

1) INFLUENCE OF THE POPULATION SIZE AND THE NUMBER OF REFERENCE POINTS ON THE PERFORMANCE OF D-NSGA-III-A

Though D-NSGA-III-A algorithm is designed for the SMWTA problem, the performance of NSGA-III significantly depends on parameter settings (i.e., the population size N and the total number of reference points H) [53]. The usually used parameter settings might be unsuitable in many cases, and parameter settings require a particular parameter tuning to achieve the best performance of the algorithm. Therefore, in this subsection, the impact of N and H on the performance of D-NSGA-III-A in solving the SMWTA problem is studied. In addition, there is a comparison between the performances of D-NSGA-III-A and NSGA-III under different parameter settings.

The population size N and the total number of reference points H are set as follows: $N \in \{10, 50, 100, 150, 200\}$ and $H \in \{10, 50, 100, 150, 200\}$. The part parameters of D-NSGA-III-A and NSGA-III are shown in Table 2. Moreover, the other parameters can be found in literature [19].

The hypervolume (HV) indicator [54] is used to evaluate the quality of a set of obtained non-dominated solutions. The larger the HV value is, the better the quality of solutions is. Before calculating the HV value, the reference point for

TABLE 3. The value of v_k.

	F1			F2		F3				F4		
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
v_k	0.68			0.86		0.7				0.92		

TABLE 4. The value of r_i.

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
r_i	2.5	2.8	3.6	4.7	4.3	5.6	7.2	4.3	5.5	2.9	3.0	3.8

TABLE 5. The value of u_j.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
u_j	0.85	0.64	0.93	0.72	0.52	0.48	0.88	0.96	0.74	0.55

TABLE 6. Parameters of D-NSGA-III-A algorithm (Part).

Parameter	H	N	$I_{\max}$	p_m	$p_{\min}$	Cd_1	Cd_2	μ
Value	150	150	200	0.07	0.1	0.8	0.2	0.25

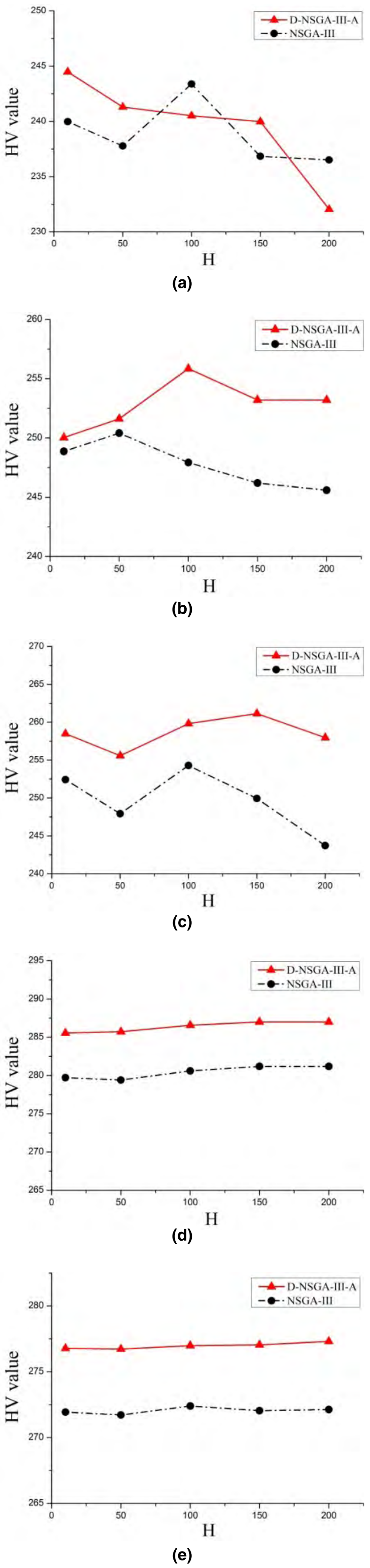
calculating HV is set to $(1, 10, 50)^T$ based on a large number of experimental results.

The same typical instance in our previous work [10] is applied to verify the performance of the algorithm. The instance includes 4 fighters that carry different numbers of missiles (12 missiles in total) and 10 targets. Parameters in the new objective are shown in the following Table 3-4.

Fig. 3 shows the influence of the H values on the performances of D-NSGA-III-A and NSGA-III with the various N values on the typical SMWTA instance. It is discernible that the choice of H and N values affects performance of both algorithms.

Firstly, the impacts of N and H on performance of both algorithms are presented. With an increasing N, the HV value of both algorithms are increased and then decreased. When the value of N exceed 150, the HV value of both algorithms decline. Both algorithms with $N \in \{150, 200\}$ achieve the relatively gentle HV value. When $N = 150$, both algorithms achieve the highest HV value. Therefore, $N = 150$ might be suitable for solving the SMWTA problem with three objectives. As shown in Fig. 3, the increase of H deteriorates the performance of both algorithms with small population size while has slight influence with large values of N. When $H = 150$ or 200, both algorithms with $N = 150$ achieve the highest values of HV on SMWTA instance. That is because that large values of H increase the number of niches in the environmental selection and promote diversity of population.

Secondly, the performance comparisons of both algorithms are presented. When $N > 10$, the value of HV of D-NSGA-III-A is better than that of NSGA-III, and the former is 5.944 more than the latter. The reason why D-NSGA-III-A performs better than NSGA-III can be considered as follows: The improvement strategy of



(a)

(b)

(c)

(d)

(e)

FIGURE 3. Influence of H values on performance of D-NSGA-III-A and NSGA-III with various N. (a) N = 10. (b) N = 50. (c) N = 100. (d) N = 150. (e) N = 200.

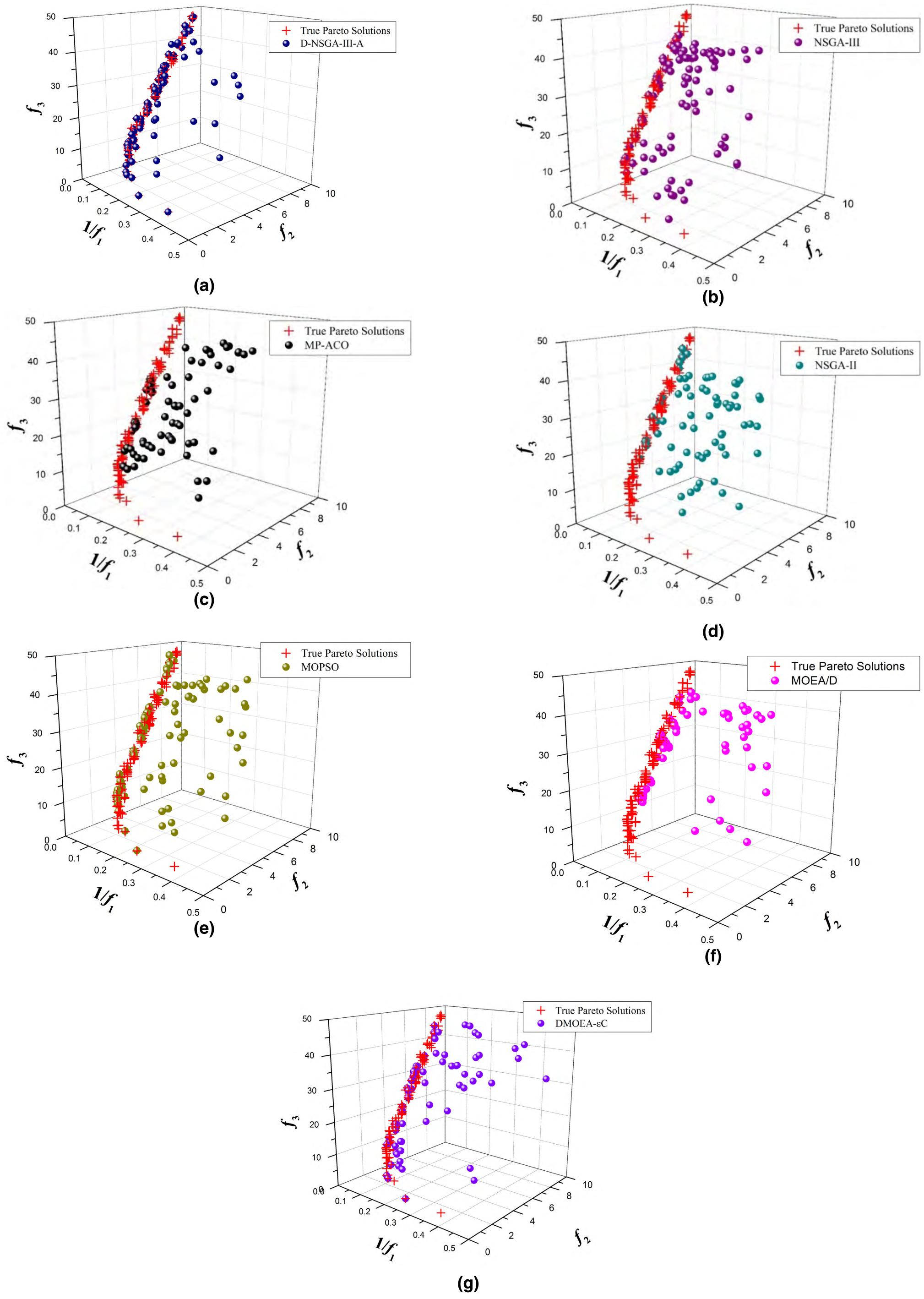


FIGURE 4. Comparison among the true pareto solutions and the final pareto solutions of NSGA-III, MPACO, NSGA-II, MOPSO, MOEA/D and DMOEA-εC. (a) D-NSGA-III-A. (b) NSGA-III. (c) MP-ACO. (d) NSGA-II. (e) MOPSO. (f) MOEA/D. (g) DMOEA-εC.

non-dominated sorting decreases the computation and adaptive operator selection mechanism guarantees the quality of Pareto solutions. Therefore, D-NSGA-III-A can find good non-dominated solutions.

According to the above analysis, $N = 150$ and $H = 150$ are set for D-NSGA-III-A and NSGA-III in the following study.

2) THE OTHER PARAMETER SETTINGS

The part of parameters for the D-NSGA-III-A is shown in Table 6.

To prove the performance of D-NSGA-III-A for solving the SMWTA problem, D-NSGA-III-A, NSGA-III, MP-ACO, NSGA-II, MOPSO, MOEA/D and DMOEA-εC have been employed for comparisons in this subsection. In order to

TABLE 7. The optimization results obtained by D-NSGA-III-A algorithm.

	1/f1	f2	f3	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
1	0.068373584	8.03	47.72109455	1	3	5	7	7	9	5	3	9	9	7	3
6	0.06891	7.34	45.10731	1	3	5	7	7	9	5	0	9	9	0	3
12	0.069696	6.82	39.07857	9	3	5	7	7	0	0	3	1	9	7	5
20	0.070586	6.28	33.99718	1	5	5	7	0	9	0	3	0	9	7	3
24	0.070713	6.1	34.65986	9	3	5	7	7	0	0	3	0	9	1	5
28	0.072165	5.69	29.68972	1	3	5	7	7	9	0	3	0	9	0	0
33	0.071471	5.48	32.70565	9	3	5	7	7	0	0	0	1	9	0	5
43	0.072594	5.3	30.90915	9	3	7	0	0	0	0	5	1	9	7	5
54	0.076009	4.73	23.73451	9	3	5	7	0	0	0	3	0	9	1	0
66	0.078163	4.01	22.51333	9	3	0	7	0	0	0	0	0	9	1	5
75	0.084818	3.32	16.50163	9	3	5	0	7	0	0	0	0	0	1	0
76	0.085522	3.32	16.39094	9	3	5	0	7	0	0	0	0	0	7	0
77	0.085774	3.27	16.223	1	3	7	0	0	0	0	5	0	9	0	0
78	0.085545	3.25	17.09656	9	3	7	0	0	0	0	0	0	0	1	5
79	0.084744	3.24	17.15761	1	3	7	0	0	0	0	0	0	9	0	5
80	0.087269	3.24	15.8079	1	5	7	0	0	0	0	0	0	9	0	3
81	0.087365	3.21	17.51216	1	3	0	0	0	0	0	0	0	9	7	5
90	0.09358	2.6	12.18867	9	3	5	0	0	0	0	0	0	0	7	0
98	0.103038	2.39	14.59817	0	0	5	7	0	9	0	0	0	0	0	0
104	0.110164	1.99	10.86758	9	0	7	0	0	0	0	5	0	0	0	0
117	0.144275	1.31	5.079115	9	0	7	0	0	0	0	0	0	0	0	0
118	0.15204	1.28	5.433661	9	0	0	0	0	0	0	0	0	0	7	0
119	0.152737	1.25	4.460934	9	7	0	0	0	0	0	0	0	0	0	0
120	0.177759	1.25	4.344609	9	3	0	0	0	0	0	0	0	0	0	0
121	0.382966	0.63	2.163636	0	7	0	0	0	0	0	0	0	0	0	0
122	0.254065	0.62	2.297297	9	0	0	0	0	0	0	0	0	0	0	0

TABLE 8. The number of targets and missiles on three cases.

	Targets	Missiles
Case 1	20	24
Case 2	30	36
Case 3	50	60

obtain a fair comparison, the population size N and maximum iteration $I_{\max}$ of NSGA-III are set to 150 and 200, respectively. In addition, the number of reference points of NSGA-III is the same as that of D-NSGA-III-A, namely $H = 150$. Moreover, the other parameters can be found in literature [19]. The parameters of MP-ACO and NSGA-II can be found in literature [10]. The parameters of MOPSO, MOEA/D and DMOEA-εC can be found in literature [36], [37] and [55].

All the algorithms are executed in C++ on a PC with 1.90 GHz CPU, 8GB RAM and Windows 7 OS. First, an enumeration approach [37] is adopted to obtain the near true optimal solutions for the typical instance. Then, D-NSGA-III-A, NSGA-III, MPACO, NSGA-II, MOPSO, MOEA/D and DMOEA-εC algorithms are applied to find Pareto solutions in 30 runs, respectively.

B. RESULTS AND ANALYSIS

The best results in 30 runs of those algorithms are compared with the true optimal solutions to verify the applicability and feasibility of D-NSGA-III-A algorithm, as shown in Fig. 4.

From the above consequences, the number of near true Pareto solutions found are 122(D-NSGA-III-A), 43(NSGA-III), 38(MPACO), 20(NSGA-II), 48(MOPSO), 35(MOEA/D) and 43(DMOEA-εC), respectively. From Fig. 4(a), it is easily obtained that the solutions obtained by D-NSGA-III-A are close and even Pareto solutions in the objective space. As can be seen in Fig. 4(b)-4(g), the other six technologies are obviously inferior to the D-NSGA-III-A algorithm. Therefore, D-NSGA-III-A is feasible for solving the SMWTA problem.

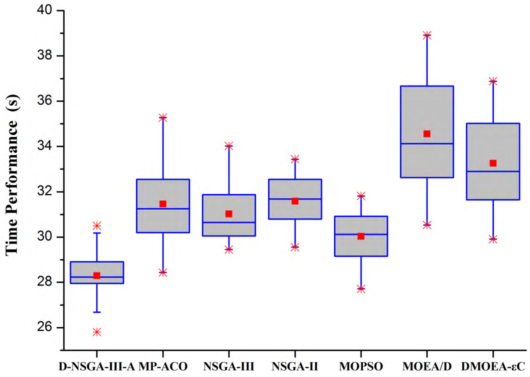


FIGURE 5. Time performances of D-NSGA-III-A, NSGA-III, NSGA-II, MP-ACO, MOPSO, MOEA/D, and DMOEA-εC.

Under the same condition ($N = 150, H = 150$), the NSGA-III can only find 101 solutions on the premise of the initial 150 reference points, and only 43 solutions are located at the Pareto front. The number of true Pareto solutions found by D-NSGA-III-A is 81.33% of H and three times as many as those of NSGA-III. Because of the adaptive operator selection mechanism in the D-NSGA-III-A algorithm,

TABLE 9. Statistical results of IGD metric over three different-scale cases.

	IGD	D-NSGA-III-A	NSGA-III	NSGA-II	MP-ACO	MOPSO	MOEA/D	DMOEA-εC
Case 1	Best	2.430321E-02	2.430732E-02	2.437420E-02	2.483155E-02	2.430531E-02	2.516420E-02	2.483246E-02
	Worst	2.440078E-02	2.445621E-02	2.445684E-02	2.521167E-02	2.444510E-02	2.526687E-02	2.521253E-02
	Average	2.434725E-02	2.434935E-02	2.437880E-02	2.485462E-02	2.434919E-02	2.519882E-02	2.485418E-02
	Median	2.434724E-02	2.434935E-02	2.437628E-02	2.485836E-02	2.434918E-02	2.520629E-02	2.485921E-02
Case 2	Best	5.104245E-02	5.280227E-02	5.286555E-02	5.541345E-02	5.280227E-02	5.743766E-02	5.541123E-02
	Worst	5.128383E-02	5.303353E-02	5.316181E-02	5.578778E-02	5.303318E-02	5.859037E-02	5.578731E-02
	Average	5.112118E-02	5.282429E-02	5.290721E-02	5.545238E-02	5.282531E-02	5.749721E-02	5.545238E-02
	Median	5.112126E-02	5.282429E-02	5.290721E-02	5.545341E-02	5.282532E-02	5.752833E-02	5.545333E-02
Case 3	Best	8.830596E-02	9.764296E-02	1.043581E-01	1.146097E-01	9.789672E-02	1.146327E-01	9.790973E-02
	Worst	8.915346E-02	1.010079E-01	1.071801E-01	1.173186E-01	1.010113E-01	1.195598E-01	1.013186E-01
	Average	8.857890E-02	9.877421E-02	1.045857E-01	1.148224E-01	9.877651E-02	1.152583E-01	9.878241E-02
	Median	8.857892E-02	9.877423E-02	1.045514E-01	1.149136E-01	9.877667E-02	1.153225E-01	9.878367E-02

TABLE 10. Statistical results of SC metric over three different-scale cases.

SC	Case 1	Case 2	Case 3
C(**,1*)/C(1*,**)	0.984/0.976	0.977/0.958	0.957/0.906
C(**,2*)/C(2*,**)	0.964/0.935	0.958/0.901	0.967/0.886
C(**,3*)/C(3*,**)	1/0.863	0.985/0.833	0.975/0.809
C(**,4*)/C(4*,**)	0.979/0.956	0.952/0.911	0.988/0.866
C(**,5*)/C(5*,**)	1/0.822	1/0.802	0.987/0.753
C(**,6*)/C(6*,**)	1/0.854	0.983/0.816	0.971/0.927

the quality of the solutions is improved. Therefore, D-NSGA-III-A is more efficient for solving the SMWTA problem. Comparing the values of three objectives, D-NSGA-III-A has a significant advantage than other six techniques, which verifies the better performance for the SMWTA problem.

Table 7 shows partial assignment results obtained by D-NSGA-III-A. We can get some results from Table 7 as follows:

If there are sufficient money and information about enemy, the scheme 1, which costs the most money and achieves the greatest expected damage, will be the best choice to accomplish a fatal attack in a state of military suppression. Unfortunately, the case is rare in actual combat.

If there is little information about enemy or difficult to organize a large-scale attack, schemes from scheme 75-81 will be available. During these schemes, the target 1, 3, 5, 7 and 9 are attacked. The damage to our fleet is the least in the scheme 80, the cost is the least in the scheme 81 and the greatest expected damage value is achieved in the scheme 75. Compared with scheme 77, the cost increases by 4% but the expected damage only increases by 0.29% in scheme 76. In terms of the least cost, the scheme 77 is more inclined to be chosen.

In the scheme 119 and 120, the missile 1 is assigned to the target 9, but the assignment of the missile 2 is different. In the same cost, the expected damage and the damage value of fighting capacity provided by the scheme 119 achieve 0.152737 and 4.460934. Compared to scheme 120, scheme 119 has saved 0.0250022 damage value, which is optimized by 14.08%, but the damage value of fighting capacity has been only increased by 0.116325 with 2.68% increasement. Thus, the scheme 119 is better than scheme 120. Considering insufficient funds, we can only choose Scheme 122.

Considering the actual air combat, pilots usually make deadly decisions in a very short time. Therefore, the time consumption is also a very important index for these algorithms. Hence, there is an experiment to compare the time consumption of six algorithms. The statistical results in 30 runs are as shown in Fig. 5.

As shown in Fig. 5, the time consumptions are 28.40033s (D-NSGA-III-A), 31.07818s (NSGA-III), 31.57239s (MPACO), 31.58856s (NSGA-II), 29.97364s (MOPSO), 34.41727s (MOEA/D) and 33.13996s (DMOEA-εC), respectively. Apparently, the D-NSGA-III-A has the least time consumption while the MOEA/D has the most time consumption. Compared with NSGA-III, the D-NSGA-III-A method allows for the same population resulting in an 8.62% lower computation time while increasing 20.79% more Pareto solutions. Apparently, the non-dominated sorting based on dominance degree matrix and adaptive operator selection mechanism make a great contribution to the time performance and the quality of Pareto solutions.

C. COMPARISON STUDY

In order to verify the performance of the D-NSGA-III-A algorithm comprehensively, three cases with different problem scales, i.e. small-scale, medium-scale and large-scale cases are adopted. The values of v_k, u_j, r_i are randomly generated within a given range. The details of the scales are given in Table 8.

In the following experiment, the inverted generational distance (IGD) [48] metric, set coverage (SC) metric and time performance are adopted to provide comprehensive comparisons among these algorithms.

1) THE COMPARISON OF IGD

The IGD metric is defined as follows:

$$IGD\left(A^*, P^*\right) = \frac{1}{|P^*|} \sum_{x \in P^*} \min_{y \in A^*} d^*(x, y) \tag{4.15}$$

TABLE 11. Statistical results of time performance over three different-scale cases.

	IGD	D-NSGA-III-A	NSGA-III	NSGA-II	MP-ACO	MOPSO	MOEA/D	DMOEA-εC
Case 1	Max	79.62	86.65	90.62	87.76	82.77	108.65	104.24
	Mean	73.28	77.13	81.21	78.94	74.28	93.32	91.79
	Min	67.96	70.47	73.09	69.47	68.59	81.78	80.37
	Sta. dev.	4.18	5.58	5.38	6.71	4.79	9.57	8.96
Case 2	Max	155.25	166.83	173.69	169.16	159.23	199.24	196.41
	Mean	146.93	153.79	161.47	156.62	149.16	185.90	179.31
	Min	138.68	142.99	152.84	147.35	140.31	167.98	164.56
	Sta. dev.	5.95	8.33	7.26	8.47	7.81	12.44	11.82
Case 3	Max	378.37	390.97	406.46	399.52	391.49	465.28	459.76
	Mean	361.01	372.11	388.34	380.89	374.74	441.63	435.26
	Min	348.69	356.36	375.90	363.36	358.91	420.17	417.55
	Sta. dev.	9.84	10.63	11.97	12.44	11.55	15.49	14.70

where A^* represents the set of the nondominated solutions found so far, $|A^*|$ can be defined as the cardinality of the A^* set, and the larger the $|A^*|$ is, the more well it represents the true Pareto front. P^* is a set of uniformly distributed points along the true Pareto front. d_i^* is the minimum Euclidean distance between x and y. Then, $IGD(A^*, P^*)$ can measure the diversity and convergence of A^* in a sense. Since the lower value means that each point must be closer to the true Pareto Front. Therefore, the lower the value of $IGD(A^*, P^*)$ is, the better the performance of the algorithm.

Details of the parameters and computing system can be found in Section V.A. All the algorithms are implemented over 30 independent runs on small-scale, medium-scale and large-scale cases. The statistical results of IGD metric are shown in Table 9.

As shown in Table 9, on same case, the different colors mean significant differences, while the same color parts are similar. The yellow-part data are the minimum values while the red parts have the maximum values. Furthermore, the smaller the values are, the better the performance of the algorithm is. Obviously, D-NSGA-III-A shows significant advantage (yellow parts) over the other algorithms on all cases. On small-scale and medium-scale cases, according to the values, the DMOEA-εC has the similar performance with MP-ACO. However, on large-scale case, the DMOEA-εC has similar performance with NSGA-III and MOPSO, and the performance is better than that of MP-ACO.

2) THE COMPARISON OF SC

The set coverage (SC) is defined as follows:

$$C\left(A^*, B^*\right) = \frac{1}{|B^*|} \left| \left\{ x \in B^* | \exists y \in A^* : y \text{ dominates } x \right\} \right| \tag{4.16}$$

where A^* and B^* represents the set of the nondominated solutions, $C(A^*, B^*)$ represents the percentage of solutions in A^* that are dominated by at least one in B^*. Here, we must note that the sum of $C(A^*, B^*)$ and $C(B^*, A^*)$ may not equal to 1. If $C(A^*, B^*)$ is greater than $C(B^*, A^*)$, solution A^* is better than solution B^* in some sense.

Here, to verify the performance of D-NSGA-III-A, we just pay our attention to the relations between the proposed algorithm and the others respectively.

In Table 10, $**$ represents the solution of D-NSGA-III-A. $1*, 2*, 3*, 4*, 5*$ and $6*$ denote the best solution of NSGA-III, NSGA-II, MP-ACO, MOPSO, MOEA/D and DMOEA-εC, respectively. From the comparison between paired SC metric average values, the solution of D-NSGA-III-A is better than that of the other six algorithms on overall cases.

3) THE COMPARISON OF TIME PERFORMANCE

In order to compare the time performance, the statistical results in 30 independent runs are shown in Table 11.

In Table 11, from case 1 to case 3, the time consumption of each algorithm increases rapidly as the scale becoming large. It is noted that the D-NSGA-III-A has the minimum values, followed by MOPSO. After that, the NSGA-III and MP-ACO have similar values of time consumption. The MOEA/D and DMOEA-εC also have similar time performance, but it is the worst. From the statistical results, in terms of time consumption, the D-NSGA-III-A performs remarkably better than the other algorithms on all cases. Apparently, it satisfies the real-time requirement better.

According to above analyses of the IGD metric, SC metric and time performance, we can conclude that the proposed D-NSGA-III-A has a good convergence and diversity of the Pareto solutions and time performance for the SMWTA problem on both small-scale and large-scale problems.

VI. CONCLUSION

In this research, D-NSGA-III-A algorithm is proposed to solve the SMWTA problem. First, considering the game process in the actual air combat, a new objective is introduced to construct a three-objective SMWTA mathematical model, which includes the damage of the enemy, the cost of missiles, and the damage value of fighting capacity. Second, the non-dominated sorting based on dominance degree matrix is proposed to substitute the non-dominated sorting based on Pareto domination. Thereby, the unnecessary or repetitive comparisons in ranking schemes can be decreased to further decrease the time consumption. Then, the dominance ratio is proposed to combine with the niching information when selecting individuals. Thus, the diversity and convergence are both took into account. Finally, the simulated binary crossover operator is substituted with adaptive operator selection mechanism to seek a balance between intensification and diversification within the decision space. From the experiments, the combination of above technologies can improve the quality of Pareto solutions and time performance, which are better than those of NSGA-III, NSGA-II, MP-ACO, MOPSO, MOEA/D and DMOEA-εC. Thereby, it verifies that the D-NSGA-III-A algorithm is more suitable for solving the SMWTA problem.

The DMWTA problem usually develops from the SMWTA problem, thus the next step may focus on the DMWTA problem with three objectives. Also, any other technology suitable for the WTA problem is also meaningful and deserves researching.

REFERENCES

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

[2] 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, 2002.

[3] 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.

[4] Y. Zhang, R. N. Yang, J. L. Zuo, and X. Jing, "Weapon-target assignment based on decomposition-based evolutionary multi-objective optimization algorithms," *Syst. Eng. Electron.*, vol. 36, no. 12, pp. 2435–2441, 2014.

[5] N. Li, W. Huai, and S. Wang, "The solution of target assignment problem in command and control decision-making behaviour simulation," *Enterprise Inf. Syst.*, vol. 7, no. 7, pp. 1059–1077, 2016.

[6] P. A. Hosein and M. Athans, *Preferential Defense Strategies. Part I: The Static Case*. Cambridge, MA, USA: MIT Press, 1990.

[7] D. G. Galati and M. A. Simaan, "Effectiveness of the Nash strategies in competitive multi-team target assignment problems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 1, pp. 126–134, Jan. 2007.

[8] D. Li and J. B. Cruz, "Defending an asset: A linear quadratic game approach," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 1026–1044, Apr. 2011.

[9] M. Faied and A. Girard, "Game formulation of multiteam target assignment and suppression mission," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 2, pp. 1234–1248, Apr. 2014.

[10] 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," *Int. J. Aerosp. Eng.*, vol. 2017, Feb. 2017, Art. no. 1746124.

[11] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*. Melbourne, FL, USA: Krieger, 1989.

[12] A. Osyczka, "An approach to multicriterion optimization problems for engineering design," *Comput. Methods Appl. Mech. Eng.*, vol. 15, no. 3, pp. 309–333, 1978.

[13] J. L. Cohon, *Multiobjective Programming and Planning*. North Chelmsford, MA, USA: Courier Corporation, 2013.

[14] 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.

[15] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proc. 3rd Annu. Conf. Genetic Evol. Comput.*, 2001, pp. 283–290.

[16] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. 5th Conf. Evol. Methods Design, Optim. Control Appl. Ind. Problems*, 2001, pp. 95–100.

[17] 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.

[18] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, Jun. 2008, pp. 2424–2431.

[19] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2013.

[20] M. W. Mkaouer, M. Kessentini, S. Bechikh, K. Deb, and, M. Ó. Cinnéide, "High dimensional search-based software engineering: Finding tradeoffs among 15 objectives for automating software refactoring using NSGA-III," in *Proc. Annu. Conf. Genetic Evol. Comput.*, 2014, pp. 1263–1270.

[21] Y. Yuan, H. Xu, and B. Wang, "An improved NSGA-III procedure for evolutionary many-objective optimization," in *Proc. Annu. Conf. Genetic Evol. Comput.*, New York, NY, USA, Jul. 2014, pp. 661–668.

[22] H. I. Mekawey, M. S. A. El-Wahab, and M. Hashem, "Novel goal-based weapon target assignment doctrine," *J. Aerosp. Comput., Inf., Commun.*, vol. 6, no. 1, pp. 2–29, 2009.

[23] Z. R. Bogdanowicz and N. P. Coleman, "Sensor-target and weapon-target pairings based on auction algorithm," in *Proc. 11th WSEAS Int. Conf. Appl. Math.*, Dallas, TX, USA, 2007, Mar. 22-24.

[24] Z. R. Bogdanowicz, "A new efficient algorithm for optimal assignment of smart weapons to targets," *Comput. Math. Appl.*, vol. 58, no. 10, pp. 1965–1969, 2009.

[25] 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.*, to be published. doi: 10.1109/TSMC.2017.2784187.

[26] H. Chen, Z. Liu, Y. Sun, and Y. Li, "Particle swarm optimization based on genetic operators for sensor-weapon-target assignment," in *Proc. 5th Int. Symp. Comput. Intell. Design*, Oct. 2012, pp. 170–173. doi 10.1109/ISCID.2012.194.

[27] J. Wang and C. Chen, "Sensor-weapon joint management based on improved genetic algorithm," in *Proc. 34th Chin. Control Conf.*, Hangzhou, China, Jul. 2015, Jul. 28-30.

[28] Z.-J. Lee, S.-F. Su, and C.-Y. Lee, "A genetic algorithm with domain knowledge for weapon-target assignment problems," *J. Chin. Inst. Eng.*, vol. 25, no. 3, pp. 287–295, 2002.

[29] Z.-J. Lee and W.-L. Lee, "A hybrid search algorithm of ant colony optimization and genetic algorithm applied to weapon-target assignment problems," in *Intelligent Data Engineering and Automated Learning* (Lecture Notes in Computer Science), vol. 2690, J. Liu, Y. Cheung, and H. Yin, Eds. Berlin, Germany: Springer, 2003. doi: 10.1007/978-3-540-45080-1_37.

[30] 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.

[31] M. Z. Lee, "Constrained weapon–target assignment: Enhanced very large scale neighborhood search algorithm," *IEEE Trans. Syst., Man, A, Syst. Humans*, vol. 40, no. 1, pp. 198–204, Jan. 2010.

[32] P. Chen, B. Shen, L. Zhou, and Y. Chen, "Optimized simulated annealing algorithm for thinning and weighting large planar arrays," *J. Zhejiang Univ. Sci. C*, vol. 11, no. 4, pp. 261–269, 2010.

[33] A.-G. Fei, L.-Y. Zhang, and Q.-J. Ding, "Multi-aircraft cooperative fire assignment based on auction algorithm," *Syst. Eng. Electron.*, vol. 34, no. 9, pp. 1829–1833, 2012.

[34] L. Hongtao and K. Fengju, "Adaptive chaos parallel clonal selection algorithm for objective optimization in WTA application," *Optik*, vol. 127, no. 6, pp. 3459–3465, 2016.

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

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

[37] J. Li, J. Chen, B. Xin, and L. Dou, "Solving multi-objective multi-stage weapon target assignment problem via adaptive NSGA-II and adaptive MOEA/D: A comparison study," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Sendai, Japan, May 2015, pp. 3132–3139.

[38] J. Li, J. Chen, B. Xin, and L. Chen, "Efficient multi-objective evolutionary algorithms for solving the multi-stage weapon target assignment problem: A comparison study," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 435–442.

[39] 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 2010.

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

[41] N. Dirik, S. N. Hall, and J. T. Moore, "Maximizing strike aircraft planning efficiency for a given class of ground targets," *Optim. Lett.*, vol. 9, no. 8, pp. 1729–1748, 2015.

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

[43] F. Johansson and G. Falkman, "An empirical investigation of the static weapon-target allocation problem," *3rd Skövde Workshop Inf. Fusion Topics (SWIFT)*, 2009, pp. 1–6.

[44] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.

[45] H. Jain and K. Deb, "An improved adaptive approach for elitist nondominated sorting genetic algorithm for many-objective optimization," in *Proc. 7th Int. Conf. Evol. Multi-Criterion Optim.*, Berlin, Germany: Springer, Mar. 2013, pp. 307–321. doi: 10.1007/978-3-642-37140-0_25.

[46] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16–37, Feb. 2016.

[47] H. Seada and K. Deb, "U-NSGA-III: A unified evolutionary optimization procedure for single, multiple, and many objectives: Proof-of-principle results," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, vol. 9019, 2015, pp. 34–49.

[48] F. Pierro, S.-T. Khu, and D. A. Savic, "An investigation on preference order ranking scheme for multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 17–45, Feb. 2007.

[49] A. Lopez *et al.*, "An alternative preference relation to deal with many-objective optimization problems," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, 2013, pp. 291–306.

[50] S. F. Adra and P. J. Fleming, "Diversity management in evolutionary many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 183–195, Apr. 2011.

[51] M. Li *et al.*, "A comparative study on evolutionary algorithms for many-objective optimization," in *Proc. 7th Int. Conf. Evol. Multi-Criterion Optim.* Berlin, Germany: Springer, 2013, pp. 261–275. doi: 10.1007/978-3-642-37140-0_22.

[52] R. A. Gonçalves, L. M. Pavelski, C. P. de Almeida, J. N. Kuk, S. M. Venske, and M. R. Delgado, *Adaptive Operator Selection for Many-Objective Optimization with NSGA-III* (Lecture Notes in Computer Science), vol. 10173, H. Trautmann *et al.* Eds. Cham, Switzerland: Springer, 2007. doi: 10.1007/978-3-319-54157-0_19.

[53] R. Tanabe and A. Oyama, "The impact of population size, number of children, and number of reference points on the performance of NSGA-III," in *Proc. 9th Int. Conf. Evol. Multi-Criterion Optim.*, New York, NY, USA, May 2017, pp. 606–621.

[54] V. L. Huang *et al.*, *Problem Definitions for Performance Assessment of Multi-Objective Optimization Algorithms*. Singapore: Nanyang Technological Univ., 2007.

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

CHUNQING GAO received the degree from the Aeronautics and Astronautics Engineering College, Air Force Engineering University, in 2016, where he is currently pursuing the Ph.D. degree. His researches are mainly focused on air combat, machine intelligence, and artificial intelligence.

YINGXIN KOU received the M.Sc. and Ph.D. degrees from Air Force Engineering University, Xi'an, China, in 1997 and 2010, respectively, where he is currently a Professor with the Aeronautics and Astronautics Engineering College. He has authored more than 30 journal papers. His research interests include air combat, nonlinear and adaptive control, artificial intelligence, multi-sensor data fusion, and optimized target assignment.

YOU LI received the degree from the Aeronautics and Astronautics Engineering College, Air Force Engineering University, in 2014, where he is currently pursuing the Ph.D. degree. His researches are mainly focused on air combat, multi-sensor data fusion, pattern recognition, and artificial intelligence.

ZHANWU LI received the M.Sc. degree from Air Force Engineering University, Xi'an, China, in 2007, where he is currently an Associate Professor with the Aeronautics and Astronautics Engineering College. He has authored more than 10 journal papers. His researches are mainly focused on multi-sensor data fusion, machine intelligence, and artificial intelligence.

AN XU received the M.Sc. and Ph.D. degrees from Air Force Engineering University, Xi'an, China, in 2009 and 2012, respectively. He is currently a Postdocotoral Researcher with the Electronic Information College, Northwestern Polytechnical University, Xi'an, and a Researcher with the Aeronautics and Astronautics Engineering College, Air Force Engineering University. His research interests include nonlinear and adaptive control, artificial intelligence, and pattern recognition.