

Research Article

An Improved Nondominated Sorting Genetic Algorithm III Method for Solving Multiobjective Weapon-Target Assignment Part I: The Value of Fighter Combat

You Li ¹, Yingxin Kou,¹ and Zhanwu Li ^{1,2}

¹Aeronautics and Astronautics Engineering College, Air Force Engineering University, Xi'an, Shaanxi 710038, China

²College of Electronic Communication, Northwestern Polytechnical University, Xi'an 710072, China

Correspondence should be addressed to Zhanwu Li; afeulzw@189.cn

Received 12 December 2017; Revised 11 March 2018; Accepted 22 March 2018; Published 19 June 2018

Academic Editor: Mahmut Reyhanoglu

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

Multiobjective weapon-target assignment is a type of NP-complete problem, and the reasonable assignment of weapons is beneficial to attack and defense. In order to simulate a real battlefield environment, we introduce a new objective—the value of fighter combat on the basis of the original two-objective model. The new three-objective model includes maximizing the expected damage of the enemy, minimizing the cost of missiles, and maximizing the value of fighter combat. To solve the problem with complex constraints, an improved nondominated sorting algorithm III is proposed in this paper. In the proposed algorithm, a series of reference points with good performances in convergence and distribution are continuously generated according to the current population to guide the evolution; otherwise, useless reference points are eliminated. Moreover, an online operator selection mechanism is incorporated into the NSGA-III framework to autonomously select the most suitable operator while solving the problem. Finally, the proposed algorithm is applied to a typical instance and compared with other algorithms to verify its feasibility and effectiveness. Simulation results show that the proposed algorithm is successfully applied to the multiobjective weapon-target assignment problem, which effectively improves the performance of the traditional NSGA-III and can produce better solutions than the two multiobjective optimization algorithms NSGA-II and MPACO.

1. Introduction

With the rapid development of military air combat, the weapon-target assignment (WTA) problem has attracted worldwide attention [1]. The WTA problem is a classic scheduling problem that aims to assign weapons to maximize military effectiveness and meet all constraints. So, it is important to find a proper assignment of weapons to targets.

The study of the WTA problem can be traced back to the 1950s and 1960s when Manne [2] and Day [3] built the model of the WTA problem. From the perspective of the quantity of objective functions, Hosein and Athans [4] classify the WTA problem into two classes: the single-objective weapon-target assignment problem and the multiple-objective weapon-target assignment (MWTAs) problem. When taking the time factor into account, Galati and Simaan [5] divide the WTA problem into two categories: dynamic

weapon-target assignment problem and static weapon-target assignment problem. The current research status of various WTA problems are summarized in Table 1.

In contrast to the single-objective weapon-target assignment problem, MWTAs can take different criteria into consideration that are more in line with real combat decision making. In this paper, we mainly focus on the static multiobjective weapon-target assignment (SMWTAs) problem, which aims at finding proper static assignments.

The combination of simulation and optimization algorithms to solve the SMWTAs problem is not new. At present, a number of studies address this problem. In Liu et al. [16], an improved multiobjective particle swarm optimization (MOPSO) was used to solve the SMWTAs problem with two objective functions: maximum enemy damage probability and minimum total firepower unit. The specific example they used contains only 7 platforms and 10 targets.

TABLE 1: Summary of variant metaheuristic algorithms and implementation of various WTA [6].

Researchers	Year	Metaheuristic algorithm	Implementation (WTA)
Lee et al. [6]	2002	IS + ACO	Static single objective
Lee et al. [7]	2002	GA	Static single objective
Lee et al. [8]	2003	GA + ACO	Static single objective
Galati and Simaan [5]	2007	Tabu	Dynamic single objective
Lee [9]	2010	VLSN	Static single objective
Xin et al. [10]	2010	VP + tabu	Dynamic single objective
Li and Dong [11]	2010	DPSO + SA	Dynamic single objective
Chen et al. [12]	2010	SA	Static single objective
Xin et al. [13]	2011	Rule-based heuristic	Dynamic multiobjective
Fei et al. [14]	2012	Auction algorithm	Static single objective
Bogdanowicz et al. [15]	2013	GA	Static single objective
Liu et al. [16]	2013	MOPSO	Static multiobjective
Zhang et al. [17]	2014	MOEA/D	Static multiobjective
Ahner and Parson [18]	2015	Dynamic programming	Dynamic multiobjective
Li et al. [19]	2015	NSGA-II, MOEA/D	Static multiobjective
Dirik et al. [20]	2015	MILP	Dynamic multiobjective
Hongtao and Fengju [21]	2016	CSA	Static single objective
Li et al. [22]	2016	MDE	Dynamic multiobjective
Li et al. [23]	2017	MPACO	Static multiobjective

Zhang et al. [17] proposed a decomposition-based evolutionary multiobjective optimization method based on the MOEA/D algorithm. Considering the constraints of attack resource and damage probability, a mathematic model on weapon-target assignment was formulated. Both the proposed repair method and appropriate decomposition approaches can effectively improve the performance of the algorithm. But the algorithm has not been tested on a large-scale WTA problem, and it has a low convergence speed.

In the work of Li et al. [19], a new optimization approach for the MWTA problem was developed based on the combination of two types of multiobjective optimizers: NSGA-II (domination-based) and MOEA/D (decomposition-based) enhanced with an adaptive mechanism. Then, a comparison study among the proposed algorithms, NSGA-II and MOEA/D, on solving instances of a three-scale MWTA problem was performed, and four performance metrics were used to evaluate each algorithm. They only applied the proposed adaptive mechanism to the MWTA problem, but they did not verify the behavior of the proposed adaptive mechanism on standard problems. In addition, they also considered the next step to solve the MWTA problem with an improved version of NSGA-II (called NSGA-III [24]).

In our previous work [23], we proposed a modified Pareto ant colony optimization (MPACO) algorithm to solve the bi-objective weapon-target assignment (BOWTA) problem and introduce the pilot operation factor into a WTA mathematic model. The proposed algorithm and two multi-objective optimization algorithms NSGA-II and SPEA-II were applied to solve different scales of instances. Simulation results show that the MPACO algorithm is successfully applied in the field of WTA, which improves the performance of the traditional Pareto ant colony optimization

(P-ACO) algorithm effectively and produces better solutions than the other two algorithms.

Although the above methods have remarkable effects on solving the SMWTA problem, all of them considered two objectives, maximizing the expected damage of the enemy and minimizing the cost of missiles, without considering the attack power. Due to the fact that fighters cannot destroy the targets at once, we put forward the value of fighter combat to evaluate the ability of sustained operational capability. On the basis of the original double-objective model, we propose the three-objective model, which is closer to real air combat. The new three-objective model includes maximizing the expected damage of the enemy, minimizing the cost of missiles, and maximizing the value of fighter combat. As the number of objectives is increased from two to three, the performance of evolutionary multiobjective algorithms (EMOAs) may deteriorate. They face some difficulties as follows: (i) A large fraction of the population is nondominated. (ii) The evaluation of diversity is computationally complex. (iii) The recombination operation may be inefficient [25]. Recently, EMOAs like the nondominated sorting genetic algorithm III (NSGA-III) [26] have been proposed to deal with these difficulties and scale with the number of objectives.

In 2014, Deb and Jain [24, 26] proposed a reference point-based many-objective evolutionary algorithm following the NSGA-II framework (namely, NSGA-III). The basic framework of the NSGA-III remains similar to the NSGA-II algorithm, but the NSGA-III improves the ability to solve the multiobjective optimization problem (MOP) by changing the selection mechanism of its predecessor. Namely, the main difference is the substitution of crowding distance for a selection based on well-distributed and adaptive reference points. These reference points help maintain the diversity of

population members and also allow the NSGA-III to perform well on MOP with differently scaled objective values. This is an advantage of the NSGA-III and another reason why we choose the NSGA-III algorithm to solve the SMWTA problem.

The NSGA-III has been successfully applied to real-world engineering problems [27, 28] and has several proposed variants, such as combining different variation operators [29], solving monoobjective problems [30], and integrating alternative domination schemes [31]. As far as we know, none of the previous related work has studied the MWTWA problem of three objective functions and applied the NSGA-III algorithm to solve the MWTWA problem.

In this paper, we have proposed an improved NSGA-III (I-NSGA-III) for solving the SMWTA problem. The proposed algorithm is used to seek better Pareto-optimal solutions between maximizing the expected damage, minimizing the cost, and maximizing the value of fighter combat. Based on the framework of the original NSGA-III, the proposed algorithm is devised with several attractive features to enhance the optimization performance, including an improvement strategy of reference points and an online operator selection mechanism.

Improvement Strategy of Reference Points. We can see from studies [24, 26] that reference points of the original NSGA-III are uniformly distributed on a hyperplane to guide solutions to converge. The locations of these reference points are predefined, but the true Pareto front of the SMWTA problem is unknown beforehand. So the mismatches between the reference points and the true Pareto front may degrade the search ability of the algorithms. If appropriate reference points can be continuously generated during the evolution according to information provided by the current population, it will be possible to achieve a solution set with good performances. Therefore, we add the improvement strategy of reference points to the original NSGA-III algorithm, that is, continuously generating good reference points and eliminating useless reference points.

Online Operator Selection Mechanism. Crossover and mutation operators used in the evolutionary process of optimization with NSGA-III can generate offspring solutions to update the population and seriously affect search capability. The task of choosing the right operators depends on experience and knowledge about the problem. The online operator selection mechanism proposed in this paper aims to automatically select operators from a pool with simulated binary crossover, DE/rand/1, and nonlinear differential evolution crossover. Different crossover operators can be selected online according to the information of generations. Another benefit of this mechanism is that the operator choice can adapt to the search landscape and improve the quality of Pareto-optimal solutions.

The rest of this paper is organized as follows. Section 2 reviews the related work. In Section 3, a new mathematical model of the SMWTA problem and assumption descriptions are presented. Section 4 provides the introduction of the NSGA-III algorithm and presents the proposed I-NSGA-III

for solving the WTA problem. Detailed improvements of the proposed algorithm are also introduced in Section 4. Section 5 is divided into two subsections as follows: (i) In order to verify the proposed algorithm, five state-of-the-art algorithms are considered for comparison studies. (ii) The proposed algorithm and others, like NSGA-III [9], MPACO [7], and NSGA-II [16], are tested on the SMWTA problem. Section 6 concludes the paper and presents a direction for future work.

2. Related Work

Many realistic problems contain several (two or more) conflicting objectives that are to be minimized or maximized simultaneously [32]. Most single-objective optimization problems can find only one solution (others may lack the appropriate conditions), but multiobjective optimization problems (MOPs) can find a set of Pareto solutions that consider all the objective functions and constraints. Generally, multiobjective optimization can be presented as follows [33]:

$$\begin{aligned} y &= f(x) = [f_1(x), f_2(x), \dots, f_m(x)], \\ e(x) &= [e_1(x), e_2(x), \dots, e_k(x)] \geq 0, \\ x &= (x_1, x_2, \dots, x_n) \in X, \\ y &= (y_1, y_2, \dots, y_m) \in Y, \end{aligned} \quad (1)$$

where X denotes the space of decision variables and Y denotes the space of objective functions.

There have been various studies on multiobjective evolutionary optimization besides NSGA-III, such as a set-based genetic algorithm for interval many-objective optimization problems, set-based many-objective optimization guided by a preferred region, a many-objective evolutionary algorithm using a one-by-one selection strategy, and many-objective evolutionary optimization based on reference points. The review on the related work will be further enriched if these studies are included.

2.1. Set-Based Evolutionary Optimization. The goal of a multiobjective evolutionary algorithm (MOEA) is to seek a Pareto solution set which is well converged, evenly distributed, and well extended. If a set of solutions and its performance indicators are taken as the decision variable and objectives of a new optimization problem, respectively, it is more likely that a Pareto-optimal set that satisfies the performance indicators will be obtained. Based on this idea, a many-objective optimization (MaOP) can be transformed into an MOP with two or three objectives, and then a series of set-based evolutionary operators are employed to solve the transformed MOP [34]. Compared with the traditional MOEAs, set-based MOEAs have two advantages: (i) the new objectives are used to measure a solution set and (ii) each individual of the set-based evolutionary optimization is a solution set consisting of several solutions of the original problem.

Researchers have carried out studies on set-based MOEAs, including the frameworks, the methods of transforming objectives, the approaches for comparing set-based individuals, and so on. The first set-based MOEA was proposed by Bader et al. [35]. In their work, solutions in a population are firstly divided into a number of solution sets of the same size, and then the hypervolume indicator is adopted to assess the performance of those sets. In the method proposed by Zitzler et al. [36], not only is the preference relation between a pair of set-based individuals defined, but the representation of preferences, the design of the algorithm, and the performance evaluation are also incorporated into a framework. Bader et al. [35] presented a set-based Pareto dominance relation and designed a fitness function reflecting the decision maker's preference to effectively solve MaOPs. A comparison of results with traditional MOEAs shows that the proposed method is effective. Besides, Gong et al. [37] also presented a set-based genetic algorithm for interval MaOPs based on hypervolume and imprecision.

2.2. Local Search-Based Evolutionary Optimization. Ishibuchi and Murata [38] firstly proposed the study combining MOEA and local search method, IM-MOGLS for short. In their work, a weighted sum approach is used to combine all objectives into one objective. After generating offspring by genetic operators, a local search is conducted starting from each new individual, optimizing the combined objectives. Based on IM-MOGLS, Ishibuchi et al. [39] add more selectivity to its starting points. Ishibuchi and Narukawa [40] proposed a hybrid of NSGA-II and local search. Knowles and Corne [41] combined local search with PAES. The use of gradients appeared in the Pareto descent method (PDM) proposed by Harada et al. in [42] and in the work of Bosman [43]. One of the few applications of achievement scalarization functions (ASFs) in MOEA area was done by Sindhya et al. [44]. MOEA was combined with a rough set-based local search in the work of Santana-Quintero et al. [45]. A genetic local search algorithm for multiobjective combinatorial optimization (MOCO) was proposed by Jazkiewicz [46]. Firstly, Pareto ranking and a utility function are applied to obtain the best solutions. Secondly, pairs of solutions are selected randomly to undergo recombination. Finally, local search is applied to offspring pairs. In their study, MOCO is used to successfully solve the traveling salesperson problem (TSP).

The above studies combine an MOEA with classical local search methods; however, none of them applies the local search method to theoretically identify poor solutions in a population. Abouhawwash et al. [47] proposed a new hybrid MOEA procedure that first identified poorly converged nondominated solutions and then improved it by using an ASF-based local search operator. They encouraged researchers to pay more attention to the Karush Kuhn Tucker proximity metric (KKTTPM) and other theoretical optimality properties of solutions in arriving at better multiobjective optimization algorithms.

2.3. Reference Point-Based Evolutionary Optimization. Existing reference point-based approaches usually adopt only one reference point to represent the decision maker's ideal solution. Wierzbicki [48] firstly proposed a reference point approach in which the goal is to achieve a Pareto solution closest to a supplied reference point of aspiration level based on solving an achievement scalarization problem. Deb and Sundar [32] introduced the decision maker's preference to find a preferred set of solutions near the reference point. Decomposition strategies have also been incorporated into reference point approaches to find preferred regions in the method proposed by Mohammadi et al. [49].

Up to date, there is only a few researches on achieving the whole Pareto-optimal solution set by employing multiple reference points. Figueira et al. [50] proposed a parallel multiple reference point approach for MOPs. In their work, the reference points are generated by estimating the bounds of the Pareto front, and solutions near each reference point can be obtained in parallel. This priori method is very convenient; however, the later evolution process increases the computational complexity. Wang et al. [51] proposed a preference-inspired coevolutionary approach. Although solutions and reference points are optimized simultaneously during the evolution process, the fitness value of an individual is calculated by the traditional Pareto dominance. In the work done by Deb and Jain [24], a hyperplane covering the whole objective space is obtained according to the current population, then a set of well-distributed reference points are generated on the hyperplane. However, the Pareto fronts of most practical problems are not uniformly distributed in the whole objective space, and it is necessary to adopt reference points which are adaptive to various problems. Liu et al. [52] proposed a reference point-based evolutionary algorithm (RPEA) method. However, the value of δ which seriously affects the performance of the algorithm is a constant during evolution. In addition, the Tchebychev approach is only adopted in this study, and it may not be appropriate to all kinds of problems.

2.4. Indicator-Based Evolutionary Optimization. Compared with the above approaches, indicator-based evolutionary algorithms (IBEAs) [53] adopt a single indicator which accounts for both convergence and distribution performances of a solution set. Because solutions can be selected one by one based on the performance indicator, the algorithm is also called one-by-one selection evolutionary optimization. The hypervolume is usually adopted as the indicator in IBEAs. However, the computational complexity for calculating hypervolume increases exponentially as the number of objectives increases. So it is hard to be used to solve MaOPs. To address this issue, Bader and Zitzler [54] proposed an improved hypervolume-based algorithm—HypE. In their work, Monte Carlo simulations are applied to estimate the hypervolume. This method can save computational resources while ensuring the accuracy of the estimated hypervolume. In recent years, some algorithms [55, 56] are also proposed to enhance the computational efficiency of IBEAs for solving MaOPs.

Motivated by simultaneously measuring the distance of the solutions to the Pareto-optimal front, and maintaining a sufficient distance between each other, Liu et al. [57] proposed a many-objective evolutionary algorithm using a one-by-one selection strategy, 1by1EA for short. However, there are two issues in this algorithm. (i) In 1by1EA, the contour lines formed by the convergence indicator have a similar shape to that of the Pareto-optimal front of an optimization problem. However, the shape of the Pareto-optimal front of a practical optimization problem is frequently unknown beforehand. (ii) The algorithm does not include a mechanism that can adaptively choose an appropriate convergence indicator or use an ensemble of multiple convergence indicators during the evolution. Based on the above two issues, we have improved the original NSGA-III.

3. Problem Formulation

The WTA formation can be described as finding a proper assignment of weapon units to target units as illustrated in Figure 1. Some formulation of the problem, including the assumptions and the new three-objective mathematical model, are introduced in this section.

3.1. Assumption Description. In this research, to establish a reasonable WTA mathematical model, the following assumptions can be defined:

Assumption 1. We assume that the mathematical model is composed of F fighters, M missiles, and T targets and the opposing groups are not necessarily equal in quantity. (Each fighter is equivalent to one platform, which possess different kinds and quantities of missiles).

Assumption 2. Each fighter can use different missiles to attack one target. (Each missile can only attack one target).

Assumption 3. The distributed unit total of each type of missile cannot exceed the number of assigned missile unit resources in a military air operation.

Assumption 4. We assume that the probability of a kill, which is labeled as q_{ij} , between the missile (i th unit of M) and the unit being attacked (j th unit of T) is provided.

Assumption 5. If the target is within the work area, a missile can be assigned effectively. If not, the missile is not.

3.2. Mathematical Model. Multiobjective WTA optimization is used to seek a balance among the maximum expected damage, minimum missile consumption, and maximum combat value. Thus, definitions and constraints related to the optimization model are shown as follows.

Definition 1. One introduces a new objective—the value fighters engage in combat on the basis of the bi-objective WTA model that one established in the literature [23]. The multiobjective model of WTA is to maximize the total

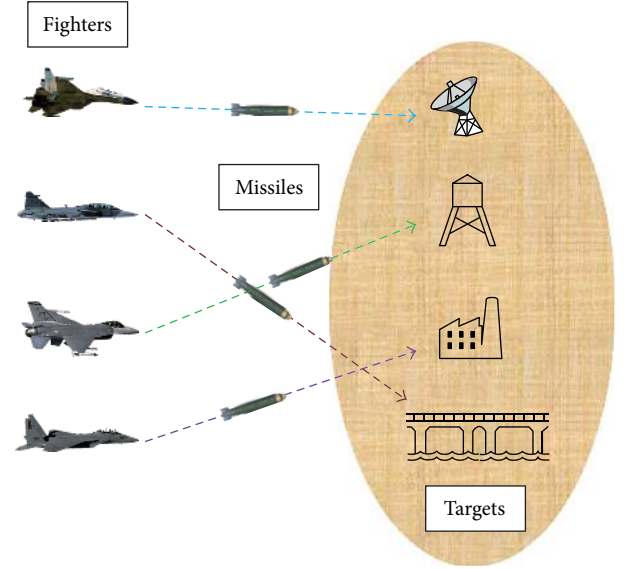


FIGURE 1: Illustration of the WTA problem.

effectiveness of attack, minimize the cost of missiles, and maximize the value of fighter combat. The mathematical functions of the model are shown as

$$\begin{aligned}
 f_1 &= \max \sum_{j=1}^n \left[1 - \prod_{i=1}^m \left(1 - \rho_k q_{ij} \right)^{x_{ij}} \right], \\
 f_2 &= \min \sum_{i=1}^m \sum_{j=1}^n c_i x_{ij}, \\
 f_3 &= \max \sum_{k=1}^s \left(1 - \frac{1}{A_k} w_k \right), \\
 f &= \min \left(\frac{1}{f_1}, f_2, \frac{1}{f_3} \right).
 \end{aligned} \tag{2}$$

Definition 2. A_k represents the number of missiles carried by the k th fighter, and w_k represents the number of missiles that have been launched on the k th fighter.

Definition 3. $q_{ij} \in (0, 1)$ represents the kill probability of each $M(i = 1, 2, \dots, m)$ missile attacking different $T(j = 1, 2, \dots, n)$ targets.

Definition 4. The decision table $X = [x_{ij}]_{M \times T}$ can be described as Table 2, where x_{ij} is a Boolean value and represents whether i missile is assigned to j target. The relationship between the Boolean value and missile allocation is shown in Table 3.

Definition 5. The constant c_i represents the cost of i th missile in this paper.

Definition 6. Based on the real battlefield, we assume that the number of missiles per fighter is not more than 4.

TABLE 2: The decision table of WTA.

	T_1	T_2	T_3	...	T_n
M_1	x_{11}	x_{12}	x_{13}	...	x_{1n}
M_2	x_{21}	x_{22}	x_{23}	...	x_{2n}
M_3	x_{31}	x_{32}	x_{33}	...	x_{3n}
M_4	x_{41}	x_{42}	x_{43}	...	x_{4n}
M_5	x_{51}	x_{52}	x_{53}	...	x_{5n}
M_6	x_{61}	x_{62}	x_{63}	...	x_{6n}
\vdots	\vdots	\vdots	\vdots	x_{ij}	\vdots
M_m	x_{11}	x_{12}	x_{13}	...	x_{mn}

TABLE 3: The relationship between Boolean value and missile allocation.

Missile allocation	Boolean value
i missile of k fighter unit is assigned to target j	$x_{ij} = 1$
i missile of k fighter unit is not assigned to target j	$x_{ij} = 0$

Definition 7. ρ_k is the pilot operation factor proposed by our previous article [23] ($\rho_k \in (0, 1)$).

Constraint. Three constraints that the above function variables must satisfy are shown in Table 4.

4. Nondominated Sorting Genetic Algorithm III

4.1. Introduction of NSGA-III. The basic framework of the NSGA-III algorithm remains similar to the NSGA-II algorithm with significant changes in its mechanism [24]. But unlike in NSGA-II, a series of reference points are introduced to improve the diversity and convergence of NSGA-III. The proposed improved NSGA-III is based on the structure of NSGA-III; hence, we give a description of NSGA-III here.

The algorithm starts with an initial population (feasible solutions) of size N and a series of widely distributed G -dimensional reference points. p represents the division and is given by the user. Das and Dennis's systematic approach [58] is used to place reference points on the normalized hyperplane having an intercept of one on each axis. The total number of reference points (H) in an M -objective problem is given by

$$H = \binom{G+p-1}{p}. \quad (3)$$

The NSGA-III uses a set of reference directions to maintain diversity among solutions. A reference direction is a ray starting at the origin point and passing through a reference point, as illustrated in Figure 2. The population size N is chosen to be the smallest multiple of four greater than H , with the idea that for every reference direction, one population member is expected to be found [30].

Let us suppose the parent population at the generation t is P_t (of size N). The offspring population Q_t having N members is obtained by recombination and mutation of P_t . C_t is the combination of parent and offspring population (of size $2N$). To preserve elite members, C_t is sorted to different non-domination levels (L_1, L_2, \dots, L_l). Thereafter, individuals of each nondomination level are selected to construct a new population S_t , starting from the first nondomination level L_1 , until the size of $S_t \geq N$ (the first time larger than N). Suppose that the last level included is the l th level. In most situations, individuals of the l th level are only sorted partially by the diversity maintenance operator. This is achieved by computing crowding distance for the l th level in NSGA-II. But the NSGA-III replaces it with reference direction-based niching. Before the above operation, objective values are normalized by formulas (4), (5), and (6).

$$f'_g(x) = f_g(x) - z_g^{\min}, \quad (4)$$

$$\text{ASF}(x, w) = \frac{\max_{g=1}^G f'_g(x)}{w_g}, \quad (5)$$

$$f_g^n(x) = \frac{f'_g(x)}{a_g - z_g^{\min}}, \quad (6)$$

where the ideal point of the population S_t is determined by identifying the minimum value (z_g^{\min}), for each objective function $g = 1, 2, \dots, G$ and by constructing the ideal point $\bar{z} = (z_1^{\min}, z_2^{\min}, \dots, z_G^{\min})$. $f'_g(x)$ denotes the translated objective functions ($x \in S_t$). $\text{ASF}(x, w)$ denotes the extreme point value in each objective axis, and w_g is the weight vector of each objective (when $w_g = 0$, it will be replaced by a small number 10^{-6}). $f_g^n(x)$ denotes the normalized objective functions, and a_g represents the intercept of the g th objective axis.

After the normalizing operation, the original reference points calculated by formulas (4) and (6) lie on this normalized hyperplane. In order to associate each population member with a reference point, the shortest perpendicular distance between each individual of S_t and each reference line is calculated.

Finally, a niche-preservation strategy is employed to select individuals from l th level that are associated with each reference point. α_h represents the number of population members from $P_{t+1} = S_t/L_l$ connected to the h th reference point. The specific niche-preservation strategy is shown as follows:

- (1) The reference point set $J_{\min} = \{h : \arg \min_h \alpha_h\}$ with the min α_h is identified. When $|J_{\min}| > 1$, one ($\bar{h} \in J_{\min}$) is selected at random.
- (2) According to the value of $\alpha_{\bar{h}}$, two cases are discussed.
 - (i) $\alpha_{\bar{h}} = 0$
 - (a) If one or more members in front the l th level are associated with the h th reference point,

TABLE 4: Detailed information about the constraints.

Constraint	Explanation	Formula
Missile assignment constraint	At most, one missile is assigned to one target	$\sum_{i=1}^m x_{ij} \geq 1 \quad \forall j = 1, 2, \dots, n.$
Quantity assignment constraint	One missile can attack only one target	$\sum_{j=1}^n x_{ij} \leq 1 \quad \forall i = 1, 2, \dots, m.$
Parameter constraint	The total allocation does not exceed the total number of missiles	$\sum_{i=1}^m \sum_{j=1}^n x_{ij} \leq m.$

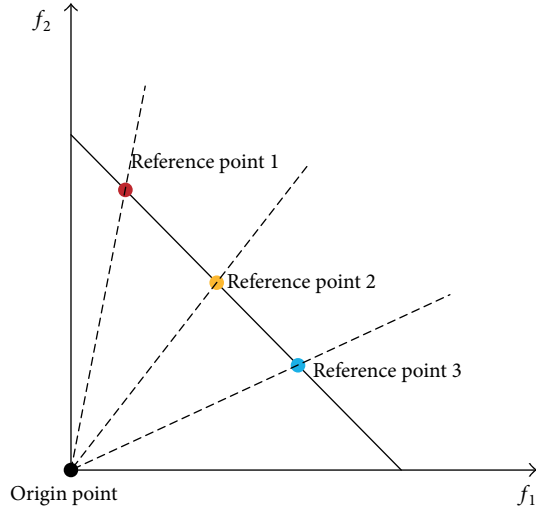


FIGURE 2: Three reference points are shown on a normalized reference line for a two-objective problem.

the one having the shortest perpendicular distance from the h th reference line is added to P_{t+1} . The count $\alpha_{\bar{h}}$ will also add one.

- (b) If one member in the front l th level is associated with the h th reference point, the h th reference point will be ignored for the current generation.

(ii) $\alpha_{\bar{h}} \geq 1$

A randomly chosen member from the front l th level that is associated with the h th reference point is added to P_{t+1} , and the count $\alpha_{\bar{h}}$ is then incremented by one.

After counts are updated, the above procedure will be repeated for individuals to fill the entire P_{t+1} .

The flow chart of the NSGA-III algorithm is shown in Figure 3.

4.2. Improvement Strategy of Reference Points. In this section, an improvement strategy of reference points is proposed. The strategy mainly includes two parts: (i) generation of new reference points and (ii) elimination of useless reference points.

NSGA-III requires a set of reference points to be supplied before the algorithm can be applied [26]. If the user does not put forward specific requirements for the Pareto-optimal front, a structured set of points created by Das and Dennis's

approach [58] is located on a normalized hyperplane. NSGA-III was originally designed to find Pareto-optimal points that are closer to each of these reference points, and the positions of the structured reference points are predefined at the beginning. However, the true Pareto front of a practical optimization problem (like the MWTA problem) is usually unknown beforehand, so the preset reference points may not reflect the development trend of the true Pareto-optimal front. In this study, a set of reference points with good performances in convergence and distribution are created by making full use of information provided by the current population. Due to the increase of the new reference points, the total number of reference points increases, and the computational complexity of the algorithm increases simultaneously. In order to keep the convergence speed of the algorithm, we propose the elimination mechanism of the reference point.

4.2.1. Generation of New Reference Points. We can learn from the above NSGA-III algorithm that the P_{t+1} population is created and the niche count α for different supplied reference points is updated after the niche operation. All reference points are expected to be useful in finding nondominated fronts ($\alpha = 1$ for every reference point). If the h th reference point is not associated with any population member, the niche count of the h th reference point (α_h) is zero. If the NSGA-III will never find an associated population member for the h th reference point, the h th reference point is considered useless. It is then better to replace the h th reference point with a new reference point that correctly reflects the direction of the Pareto-optimal front. However, we do not know if the h th reference point is eventually useful in advance. Under this circumstance, we simply add a set of reference points by adopting the formulas (7) and (8). The scale of new reference points equals the number of $\alpha = 0$ reference points.

$$h_{\text{new}}^g = \min_{x \in P_t} f_g(x) + \varepsilon_g, \quad (7)$$

$$\varepsilon_g = \delta \left(f_g^{\max} - f_g^{\min} \right), \quad (8)$$

where δ is a random number that belongs to $(0, 1)$ and $h_{\text{new}}^g = (h_{\text{new}}^1, h_{\text{new}}^2, \dots, h_{\text{new}}^G)$ denotes a new reference point for the objective g . $f_g^{\max}(x)$ and $f_g^{\min}(x)$ are the maximal and minimal values, respectively, of the g th objective.

The pseudo code can be demonstrated as follows:

In many cases, the total number of reference points will be greatly increased by the above operations, and many of the new reference points eventually become

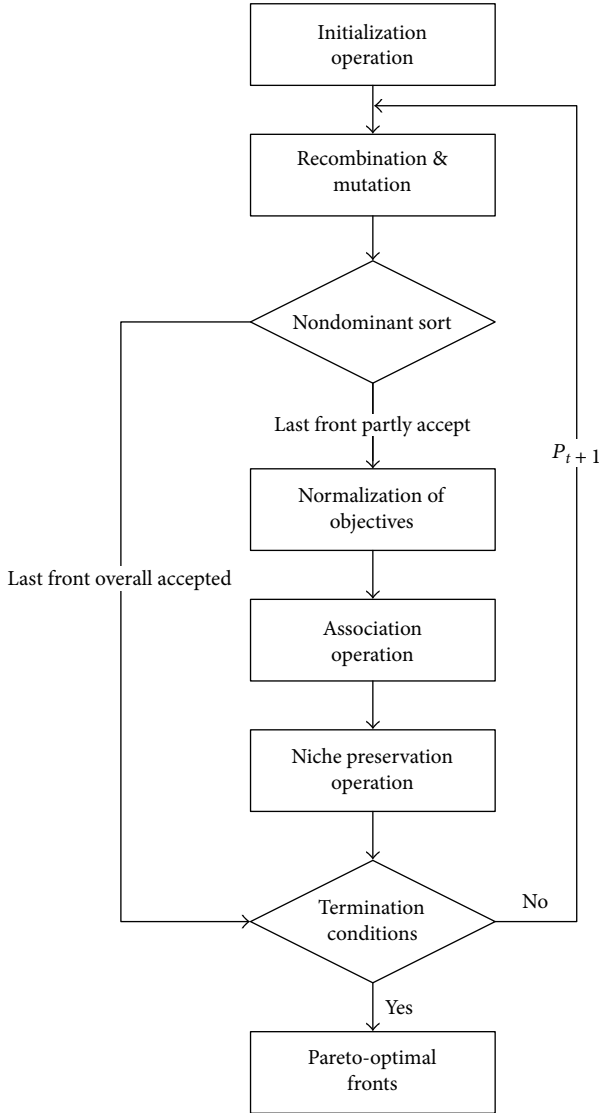


FIGURE 3: A flow chart of the NSGA-III algorithm.

useless. With the large increase of reference points, the algorithm will also be slowed down. Thus, we consider keeping the convergence speed of the algorithm by eliminating useless reference points, as described in the following subsection.

4.2.2. Elimination of Useless Reference Points. After new reference points are generated by the above operation, the niche count α of all reference points will be recalculated and recorded. Note that the total value of niche count α is equal to population size N (namely, $\sum_{h=1}^H \alpha_h = N$). Ideally, each reference point is exactly associated with one solution from the P_{t+1} population; that is, solutions are well-distributed among the reference points. Then, all reference points for $\alpha=0$ are removed from the total reference point set H . However, in order to maintain uniform distribution of the reference points, the reference points obtained by Das and Dennis's

systematic approach [58] will not be deleted. Thus, the existing reference points consist of two parts: (i) the original reference points (even if their niche count is zero) and (ii) all $\alpha = 1$ reference points.

Based on the niche count α of the respective reference points and information provided by the current population, the reference point set is adaptively redefined by generation and elimination operations. The improvement strategy for the reference points is intended to maintain diversity and guide the solutions closer to the Pareto-optimal front.

4.3. Online Operator Selection Mechanism. In multiobjective evolutionary algorithms (MOEAs), crossover and mutation operators are used to generate offspring solutions to update the population, and it can seriously affect search capability. In the original NSGA-III algorithm, only a simulated binary crossover (SBX) operator is adopted, and different crossover operators cannot be selected online according to the information of generations. Therefore, we propose a strategy based on the performance of previous generations to select operators adaptively from a pool, that is, an online operator selection (OOS) mechanism. According to a study in the literature [59], adaptive operator selection can select an appropriate strategy to adapt to the search landscape and produce better results.

Based on the information collected by the credit assignment methods, the OOS is applied to select operators for generating new solutions. In this paper, we use a probability method that uses a roulette wheel-like process for selecting an operator to solve the dilemma of exploration and exploitation.

Probability matching (PM) is one of the famous probability operator selection methods. The formula for calculating the probability of operator op being selected at next generation $t + 1$ is shown below [60]:

$$d_{op}(t + 1) = d_{\min} + (1 - K \cdot d_{\min}) \cdot \frac{\lambda_{op}(t + 1)}{\sum_{i=1}^K \lambda_i(t + 1)}, \quad (9)$$

where K is the number of operators and d_{\min} is the minimal probability of any operator. λ_{op} is the quality associated with operator op . Clearly, the sum of probabilities for all operators is 1 ($\sum_{i=1}^K \lambda_i(t + 1) = 1$). If one operator gets rewards during many generations and the others' rewards are almost 0, its maximum selection probability d_{\max} is equal to $1 + (1 - K)d_{\min}$.

The pool ζ in the proposed algorithm is composed of three well-known strategies: simulated binary crossover (SBX), DE/rand/1, and nonlinear differential evolution crossover (NDE). The parent individuals x_1 , x_2 , and x_3 are randomly selected from the population P_t in the original NSGA-III algorithm. The small difference in our algorithm is that the first parent individual x_1 of all strategies is equal to the current solution. Three operators are shown as below.

4.3.1. Simulated Binary Crossover. The simulated binary crossover (SBX) operator is proposed by Deb and Agrawal

and is found to be particularly useful in problems where the upper and lower bounds of the multiobjective global optimum are not known a priori [61]. Two offspring solutions, y_1 and y_2 , are created from two parent solutions, x_1 and x_2 , by formulas (10) and (11) as follows:

$$y_1 = \frac{1}{2}[(1 - \beta_1)x_1 + (1 + \beta_1)x_2], \quad (10)$$

$$y_2 = \frac{1}{2}[(1 + \beta_2)x_1 + (1 - \beta_2)x_2], \quad (11)$$

where the spread factor β_i is defined as the ratio of the absolute difference in offspring values to that of the parents and is a random number; β_i can be calculated by the formula (12) as follows:

$$\beta_i = \begin{cases} (2\varphi_i)^{1/(\eta_c+1)}, & \varphi_i \leq \frac{1}{2}, \\ [2(1 - \varphi_i)]^{-1/(\eta_c+1)}, & \text{otherwise,} \end{cases} \quad (12)$$

where φ_i is a random number between $[0, 1]$, and η_c is the crossover parameter given by the user.

4.3.2. DE/rand/1. The DE/rand/1 operator is one of the most commonly used DE variants [62], and all different solutions are randomly chosen from the population. So, this strategy does not generate biased or special search directions, and then a new direction is selected at random each time. The DE/rand/1 strategy can be defined by formula (13).

$$y = x_1 + U \cdot (x_2 - x_3), \quad (13)$$

where U represents the mutation scaling factor.

4.3.3. Nonlinear Differential Evolution Crossover. The nonlinear differential evolution crossover (NDE) strategy was presented in the literature [63] for the MOEA/D framework and was a hybrid crossover operator based on polynomials. The advantage of this strategy is that it ignores the values of the crossover rate and the mutation scaling factor. The offspring can be generated by formula (14).

$$y = \psi^2 \mu_1 + \psi \mu_2 + \mu_3, \quad (14)$$

where ψ is generated according to an interpolation probability (V_{itr}) [63] and can be defined by formula (15).

$$\psi = \begin{cases} \text{rand}[0, 2], & V_{itr} \geq 1, \\ \text{rand}[2, 3], & \text{otherwise.} \end{cases} \quad (15)$$

The parameters μ_1 , μ_2 , and μ_3 are given by formulas (16), (17), and (18) accordingly.

$$\mu_1 = \frac{x_3 - 2x_2 + x_1}{2}, \quad (16)$$

$$\mu_2 = \frac{4x_2 - 3x_3 - x_1}{2}, \quad (17)$$

$$\mu_3 = x_3. \quad (18)$$

4.4. The Proposed NSGA-III Algorithm. In this paper, we add the improvement strategy of reference points and online operator selection mechanism to the NSGA-III framework. The pseudo code of the proposed NSGA-III is shown in Algorithm 2, where differences with the original NSGA-III are set in italic and bold-italic.

Compared with the original NSGA-III, the proposed algorithm has two main parts. Bold-italic marks are the improvement strategy of the reference points, and italic marks are the online operator selection mechanism.

- (1) In the bold-italic parts, the reference points that satisfy the condition (Section 4.2.2) are first removed according to the niche count α of all reference points. Second, new reference points are generated by referring the process of Algorithm 1. Based on the niche count α of each reference point and information provided by the current population, the reference point set is adaptively redefined by generation and elimination operations.
- (2) There are two differences between our proposed algorithm and the original NSGA-III in the italic parts.
 - (i) The first difference is the selection of the operator to be used (Step 4), which occurs based on the probabilities associated with each operator. With the success probability of an operator increasing, the most successful operator will be selected more often, and the quality of the solutions will also be improved theoretically. Because the same operator is selected by a deterministic approach during all recombination (Step 5) of the same generation, an undesirable bias should be introduced to select the best performance operator in the initial generations. So, stochastic selection mechanisms are applied in this paper.
 - (ii) Other differences are the calculation of the rewards (Step 22) and the update of information with each operator (Step 23).
 - (a) In this paper, the op in the pool ζ has associated a probability $d_{op}(t)$ of being selected at generation t by formula (9). The adapted process is based on its quality $\lambda_{op}(t)$, which is updated according to a reward $e_{op}(t)$. The reward $e_{op}(t)$ adopted in this paper is a Boolean value ($e_{op}(t) = \{0, 1\}$) in Step 22. If x is generated by the operator op, x does not belong to P_t but belongs to P_{t+1} ; then, $e_{op}(t) = 1$; otherwise, $e_{op}(t)$ is equal to 0.

```

Input:  $P_{t+1}, \alpha, \delta$ 
Output:  $h_{new}^g$ 
1: After niche operation,  $P_{t+1}$  is created and  $\alpha$  is updated;
2: for  $h_{new}^g = h_{new}^1 \rightarrow h_{new}^G$  do
3:   for  $g = 1 \rightarrow G$  do
4:     According to formula (7) and (8), then generate the reference points  $h_{new}^g$  based on these selected
     individuals;
5:     check position  $h_{new}^g$ ;
6:     if  $h_{new}^g$  lie on the first quadrant then
7:       check existence  $h_{new}^g$ ;
8:       if  $h_{new}^g$  doesn't exist in then
9:         store  $h_{new}^g$ ;
10:      end if
11:    end for
13: end for
14: Output  $h_{new}^g$ .

```

ALGORITHM 1: Generation of New Reference Points.

```

Input:  $P_0, I_{max}, N, H$ 
Output: Pareto solutions
1: Initialize uniform distribution reference points  $H$ ;
2:  $t = 0$ ;
3: while termination conditions are not satisfied ( $t < I_{max}$ ) do
4:    $op \leftarrow \text{Select operator}()$ ;
5:    $C_t = Q_t \cup \text{Recombination \& Mutation}(P_t)$ ;
6:    $(L_1, L_2, \dots, L_l) = \text{Non-dominated-sort}(C_t)$ ;
7:    $i = 1$ ;
8:   repeat
9:      $P_{t+1} = P_{t+1} \cup L_i$ ;
10:     $i = i + 1$ ;
11:  until  $|P_{t+1}| \geq N$ ;
12:  if  $|P_{t+1}| = N$  then
13:    break;
14:  else
15:    Normalize the objectives;
16:    Delete the useless reference points;
17:    Associate each solution in  $P_{t+1}$  with a reference point;
18:    Compute niche count  $\alpha$  of reference points;
19:    Fill  $P_{t+1}$  with  $N - |P_{t+1}|$  solutions from  $L_i$  using niching information;
20:    Generate new reference points;
21:  end if
22:  Calculate operator rewards ( $P_t, P_{t+1}$ );
23:  Update operator information ();
24:   $t = t + 1$ ;
25: end while
26: return Pareto-optimal front.

```

ALGORITHM 2: The Proposed NSGA-III Algorithm.

(b) After calculating the rewards, the quality $\lambda_{op}(t)$ of the operator op available in the pool can be updated by formula (19):

$$\lambda_{op}(t+1) = (1 - \theta) \cdot \lambda_{op}(t) + \theta \cdot e_{op}(t), \quad (19)$$

where θ is the adaption rate ($\theta \in (0, 1)$).

Finally, the operator selection probabilities are updated by formula (9).

5. Experiment Results and Analysis

5.1. Test Problem. In order to verify the proposed algorithm, five state-of-the-art algorithms are considered for comparison studies. They are NSGA-III-OSD [64], NSGA-III [24], NSGA-

II [65], MPACO [23], and MOEA/D [66]. NSGA-III and NSGA-II are the traditional NSGA algorithms. NSGA-III-OSD is an improved version of the NSGA-III based on objective space decomposition. In MOEA/D algorithm, a predefined set of weight vectors is applied to maintain the solution diversity. MPACO is the algorithm we proposed before. All algorithms are tested with 4 different benchmark MaOP problems known as DTLZ1 to DTLZ4 [67].

For DTLZ, 4 instances (DTLZ1–4) with 3, 5, 8, 10, and 15 objectives are used. Based on the work of Deb et al. [67], the number of decision variables is set as $DV = G + r - 1$, where $r = 5$ for DTLZ1 and $r = 10$ for DTLZ2–4. According to the work [68], the number of decision variables is set as $DV = p_v + dv$. The parameters the position-related variable p_v and the distance-related value dv are set to $2 \times (G - 1)$ and 20, respectively. The main characteristics of all benchmark problems are shown in Table 5.

In this subsection, inverted generational distance (IGD) indicator [69] is used to evaluate the quality of a set of obtained nondominated solutions. This indicator can measure the convergence and diversity of solutions simultaneously. Smaller IGD values indicate that the last nondominated population has better convergence and coverage of the Pareto front. Each algorithm is conducted 30 runs independently on each test problem. Aiming to be as fair as possible, in each run, all the comparison algorithms perform the same maximum iteration I_{\max} as shown in Table 6. The population size used in this study for different numbers of objectives is shown in Table 7.

The six algorithms considered in this study need to set some parameters, and five of them are shown in Table 8. The parameters of the MPACO algorithm can be found in the literature [23].

Comparison results of I-NSGA-III with five other MOEAs in terms of IGD values on different objectives of DTLZ1–4 test problems are presented in Tables 9–12. It shows both the median and standard deviation of the IGD values on 30 independent runs for the six compared MOEAs, where the best median and standard deviation are highlighted in italic.

Based on the statistical results of DTLZ1, we can see that I-NSGA-III shows better performance than the other five MOEAs on three-, eight- and ten-objective test problems. For five- and fifteen-objective problems, it achieves the second smallest IGD value. Furthermore, NSGA-III can obtain the best IGD value on the fifteen-objective test problem and MPACO can obtain the best IGD value on the five-objective test problem. NSGA-II can deal with three-objective instances but works worse on more than three objectives.

Based on the statistical results of DTLZ2, we can see that the performance of I-NSGA-III, NSGA-III-OSD, and MOEA/D is comparable in this problem. The NSGA-III is worse than two improved algorithms (I-NSGA-III and MOEA/D) and MOEA/D, however, is better than MPACO. Based on the IGD values obtained by NSGA-II on three- and five-objective problems, we find that this algorithm can perform well. But when the number of objectives increases, NSGA-II still works worst among the six algorithms.

Based on the statistical results of DTLZ3, we find that I-NSGA-III performs significantly best among six algorithms

TABLE 5: Characteristics of test problems.

Test problem	Characteristics
DTLZ1	Linear, multimodal
DTLZ2	Concave
DTLZ3	Concave, multimodal
DTLZ4	Concave, biased

TABLE 6: Maximum iteration for each test problem.

	The number of objectives				
	3	5	8	10	15
DTLZ1	400	600	750	1000	1500
DTLZ2	250	350	500	750	1000
DTLZ3	1000	1000	1000	1500	2000
DTLZ4	600	1000	1250	2000	3000

TABLE 7: Number of population size.

The number of objectives	The size of populations	Divisions
3	92	{12}
5	212	{6}
8	156	{3,2}
10	276	{3,2}
15	136	{2,1}

TABLE 8: Parameter setting of each algorithm.

	η_c	η_m	p_c	p_m	Other parameters
I-NSGA-III			—		$d_{\min} = 0.1$ $U = 0.5$
	30				$\mu = 0.2$
NSGA-III-OSD [64]		20		$1/DV$	$\mu = 0.7$ $\theta = 5$
NSGA-III [24]			1.0		—
NSGA-II [65]					—
					$T = 20$
MOEA/D [66]	20				$\delta = 0.9$ $\theta = 5$

on different objectives of the DTLZ3 problem. NSGA-III-OSD can achieve a similar IGD value as NSGA-III. Although MOEA/D can achieve the close to smallest IGD value on the three- and five-objective test problems, it significantly worsens on more than five-objective test problems. MPACO can defeat MOEA/D on eight-, ten- and fifteen-objective test problems. In addition, NSGA-II is still helpless for more than three-objective problems.

Based on the statistical results of DTLZ4, we find that I-NSGA-III performs significantly better than the other five MOEAs on almost all DTLZ4 test problems, except the eight-objective instance. Furthermore, the standard deviation of IGD obtained by I-NSGA-III shows that the proposed

TABLE 9: Median and standard deviation of the IGD values achieved by each algorithm on DTLZ1 (the best medians are in italic font).

	<i>G</i>	I-NSGA-III	NSGA-III-OSD	NSGA-III	MPACO	MOEA/D	NSGA-II
DTLZ1	3	<i>3.687e-04</i>	4.087e-04	4.056e-04	7.063e-04	5.709e-04	1.861e-03
		<i>7.68e-05</i>	2.14e-04	1.76e-03	1.25e-04	5.75e-04	1.16e-02
	5	3.492e-04	2.863e-03	3.119e-03	<i>3.428e-04</i>	3.781e-04	2.321e-02
		1.10e-04	2.45e-03	2.62e-03	<i>4.06e-03</i>	7.05e-05	3.22e-03
	8	<i>2.559e-03</i>	3.125e-03	3.356e-03	3.863e-03	3.414e-03	4.023e-02
		<i>2.280e-04</i>	2.32e-03	4.31e-03	3.29e-03	2.50e-02	1.26e-01
	10	2.942e-03	3.243e-03	3.397e-03	4.163e-03	4.596e-03	5.873e-02
		<i>3.62e-04</i>	1.28e-02	2.87e-03	6.44e-03	6.54e-03	4.13e-02
	15	2.212e-02	2.315e-02	<i>2.080e-02</i>	2.273e-02	2.271e-02	4.873e+01
		2.15e-03	2.76e-02	<i>2.87e-04</i>	2.15e-02	6.56e-03	3.32e+01

TABLE 10: Median and standard deviation of the IGD values achieved by each algorithm on DTLZ2 (the best medians are in italic font).

	<i>G</i>	I-NSGA-III	NSGA-III-OSD	NSGA-III	MPACO	MOEA/D	NSGA-II
DTLZ2	3	<i>4.921e-03</i>	4.944e-03	5.212e-03	6.898e-03	5.021e-03	7.915e-03
		<i>6.09e-05</i>	1.45e-04	1.08e-04	1.94e-03	1.01e-04	1.865e-03
	5	<i>2.540e-02</i>	2.549e-02	2.593e-02	2.821e-02	2.548e-02	3.815e-02
		<i>3.17e-04</i>	4.24e-05	3.37e-04	2.25e-03	1.58e-03	9.29e-03
	8	<i>3.438e-02</i>	3.459e-02	3.461e-02	4.045e-02	3.451e-02	6.992e-02
		<i>2.01e-03</i>	1.46e-03	1.23e-03	1.18e-02	3.43e-03	4.97e-02
	10	3.563e-02	3.573e-02	3.581e-02	4.092e-02	<i>3.552e-02</i>	7.865e-02
		1.56e-03	1.322e-03	2.19e-03	8.63e-03	<i>9.83e-04</i>	7.65e-03
	15	4.638e-02	<i>4.618e-02</i>	4.656e-02	5.252e-02	4.622e-02	3.561e+01
		2.22e-03	<i>2.23e-03</i>	1.89e-03	8.50e-03	9.99e-04	1.42e+01

TABLE 11: Median and standard deviation of the IGD values achieved by each algorithm on DTLZ3 (the best medians are in italic font).

	<i>G</i>	I-NSGA-III	NSGA-III-OSD	NSGA-III	MPACO	MOEA/D	NSGA-II
DTLZ3	3	<i>6.164e-03</i>	6.456e-03	6.353e-03	8.129e-03	6.192e-03	8.092e-03
		<i>1.10e-04</i>	6.54e-03	4.14e-03	1.96e-03	5.33e-04	2.74e-02
	5	<i>2.639e-02</i>	2.783e-02	2.679e-02	2.896e-02	2.658e-02	6.013e-02
		<i>7.93e-05</i>	6.12e-03	3.77e-03	3.83e-03	5.75e-04	5.18e-01
	8	<i>3.582e-02</i>	3.632e-02	3.812e-02	4.040e-02	4.378e-02	6.816e-02
		<i>1.65e-03</i>	7.12e-03	8.49e-03	6.91e-03	1.69e-01	1.63e-01
	10	<i>3.666e-02</i>	3.706e-02	3.797e-02	4.260e-02	5.684e-02	8.562e-02
		<i>1.11e-03</i>	3.25e-03	5.74e-03	1.12e-02	6.79e-02	2.11e-01
	15	<i>4.636e-02</i>	6.509e-02	6.488e-02	5.403e-02	7.613e-02	9.866e+01
		<i>2.13e-03</i>	8.32e-02	3.42e-02	6.25e-02	2.04e-01	7.11e+01

TABLE 12: Median and standard deviation of the IGD values achieved by each algorithm on DTLZ4 (the best medians are in italic font).

	<i>G</i>	I-NSGA-III	NSGA-III-OSD	NSGA-III	MPACO	MOEA/D	NSGA-II
DTLZ4	3	<i>6.165e-03</i>	7.171e-03	1.436e-02	1.851e-02	5.514e-02	6.411e-02
		<i>2.14e-05</i>	3.45e-04	3.73e-03	2.13e-01	2.64e-03	3.58e-01
	5	<i>2.696e-02</i>	2.919e-02	2.733e-02	3.057e-02	4.622e-02	5.536e-02
		<i>6.38e-04</i>	6.51e-04	6.57e-04	1.58e-03	1.53e-01	1.99e-01
	8	<i>4.747e-02</i>	<i>4.74e-02</i>	4.753e-02	5.222e-02	7.999e-02	9.623e-02
		<i>9.73e-04</i>	<i>8.56e-04</i>	9.83e-04	6.75e-02	2.46e-02	1.25e-01
	10	<i>4.535e-02</i>	4.621e-02	5.09e-02	5.899e-02	7.92e-02	8.478e-02
		<i>1.53e-04</i>	6.19e-05	1.87e-02	6.02e-03	1.87e-02	9.56e-02
	15	<i>5.831e-02</i>	5.893e-02	6.777e-02	6.581e-02	9.01e-02	9.969e-02
		<i>1.21e-02</i>	1.37e-02	3.91e-02	1.46e-02	3.52e-01	7.48e-02

algorithm is rather robust. NSGA-III-OSD shows the closest overall performance to I-NSGA-III. MOEA/D is significantly worse than MPACO. NSGA-II still has no advantage on the DTLZ4 test problem.

The above results show that the proposed I-NSGA-III could perform well on almost all the instances in DTLZ1–4, and the obtained solution sets have good convergence and diversity. In the next subsection, the proposed algorithm will be applied to solve an SMWTA problem.

5.2. SMWTA Problem

5.2.1. Parameter Setting. For the MWTa problem, the population size N of the I-NSGA-III algorithm is 150, and the maximum number of iterations I_{\max} is 200. According to the approach in the literature [58], the total number of reference points H is set to 120. The DE scaling factor U , polynomial mutation probability p_m , and PM minimum probability d_{\min} are used in this work, as they are frequently used in the literature [68]. Some parameters for the I-NSGA-III are shown in Table 13.

Table 14 shows the number of reference points H , population size N , and maximum iteration I_{\max} for different algorithms. Other parameters in the NSGA-III algorithm are the same as those in the literature [24]. The parameters of the MPACO algorithm and the NSGA-II algorithm can be found in the literature [23].

5.2.2. Simulation Environment. We can see from Table 15 that the proposed algorithm has been implemented in C++ on a CPU Intel(R) Core(TM) i5-4460T with 1.90 GHz and 8 GB of RAM. The operating system is Windows 7 64-bit.

5.2.3. Numerical Experiments and Analysis. We use the same specific instance as in our previous work [23] 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. Appendix A shows missile damage probability p_{ij} ($i = 1, \dots, 12, j = 1, \dots, 10$), pilot operation factor ρ_k ($k = 1, \dots, 4$), and cost of missiles c_i .

First, an enumeration approach [19] is employed to get a set of evenly distributed true optimal solutions and thus obtain evenly distributed true Pareto solutions (PSs) for the specific instance. Second, in order to verify the applicability and feasibility of the proposed algorithm, we apply I-NSGA-III, NSGA-III, MPACO, and NSGA-II to find PSs in the instance. The statistical results are shown in Figures 4–7.

Figures 1–4 show the distribution of the true PSs solved by the enumeration method and final PSs obtained by I-NSGA-III, NSGA-III, MPACO, and NSGA-II. We can see that the optimization results of the I-NSGA-III algorithm are obviously better than those of the other algorithms because the I-NSGA-III can find close and even PSs in the objective space. It is evident that the algorithm can guarantee the quality of solutions. So the I-NSGA-III for optimizing the SMWTA is well verified to be feasible.

In Figure 1, 150 evenly distributed solutions in the real Pareto front are found, while only 120 reference points are preset. Since the reference point and the solution are one-

TABLE 13: Parameters of I-NSGA-III algorithm (Part).

Parameter	Value
η_c	30
η_m	20
d_{\min}	0.1
p_m	0.07
U	0.5
H	120
N	150
I_{\max}	200

TABLE 14: Number of reference points, population size, and maximum number of iterations for all algorithms.

	I-NSGA-III	NSGA-III	MPACO	NSGA-II
Reference points (H)	120	120		
Population size (N)	150	120	150	150
Maximum iteration (I_{\max})		200		

to-one correspondence, the effectiveness of the improvement strategy of reference points—generating new reference points and eliminating useless reference points operations—is demonstrated. This strategy increases the number of reference points from 120 to 150, improves the efficiency of the algorithm, and finds more Pareto-optimal solutions that meet the requirements. Meanwhile, due to the adoption of the online operator selection mechanism in the I-NSGA-III algorithm, the search landscape is reduced and the quality of the solutions is improved. However, in Figure 2, the original NSGA-III algorithm can only find 95 solutions on the premise of the initial 120 reference points, and only a small number of individuals are located at the Pareto front. Therefore, the original NSGA-III is less efficient than the algorithm proposed in this paper for solving the SMWTA problem. As we can see from Figures 3 and 4, although MPACO is superior to NSGA-II to some extent, the two algorithms are obviously inferior to the I-NSGA-III algorithm.

We analyze results from Appendix B and Appendix C as follows:

When funds for national defense are sufficient and detailed enemy information are available, we can choose Scheme 1, which costs the most money and obtains the greatest expected damage to the enemy to complete a fatal attack. As we are in a repressive state of military power, we can accomplish the task with only one attack in Scheme 1. However, this situation is rare in real combat. When we have only a small amount of information about the enemy or it is difficult to launch a large-scale attack, we choose one scheme among Schemes 147, 149, and 150 that can achieve maximum fighter combat value to launch a probing attack. In these three schemes, taking into account the least cost in terms of money and the greatest expected damage value, we should choose Scheme 150.

TABLE 15: Parameters of simulation environment.

Computer specifications	Intel(R) Core(TM) i5-4460T CPU at 1.90 GHz with 8.00 GB RAM
Operating system	Windows 7 (x64) operating system
Language	C++
Software	Microsoft Visual C++ 6.0

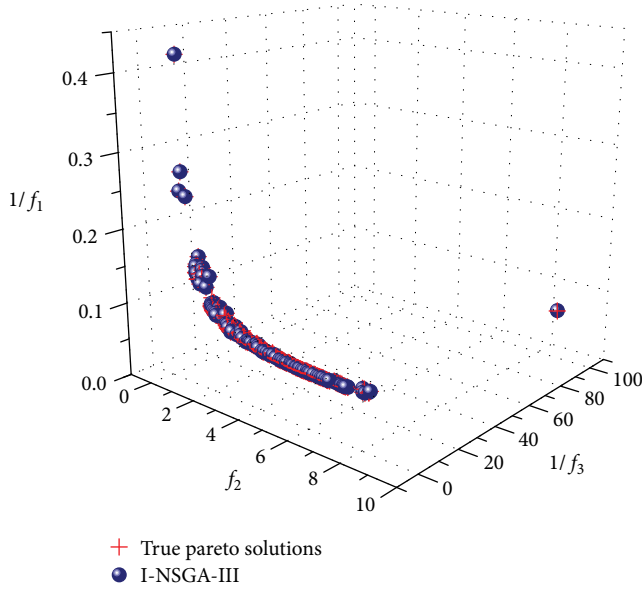


FIGURE 4: Plots of the true PSs and the final PSs found by I-NSGA-III on the specific instance.

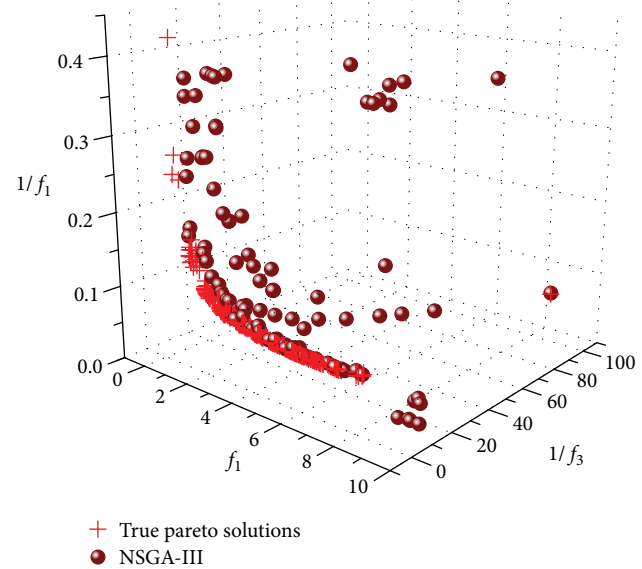


FIGURE 5: Plots of the true PSs and the final PSs found by NSGA-III on the specific instance.

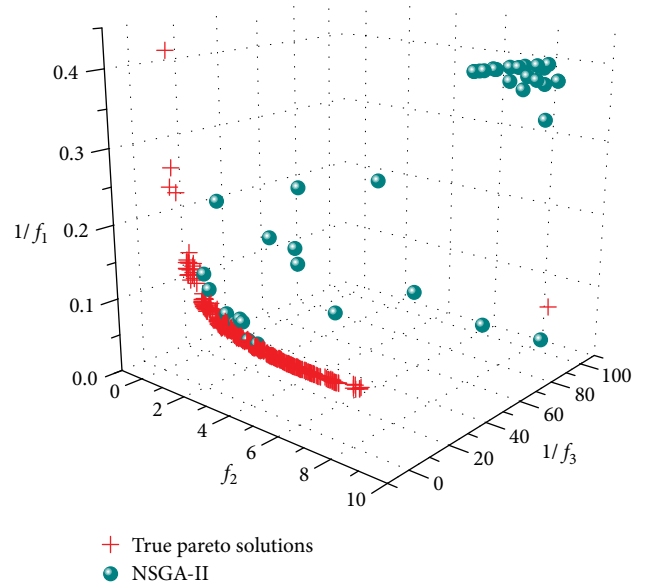


FIGURE 6: Plots of the true PSs and the final PSs found by NSGA-II on the specific instance.

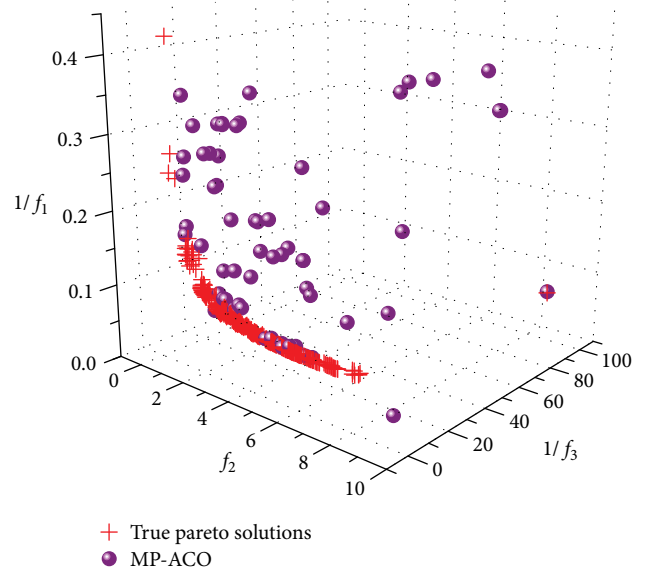


FIGURE 7: Plots of the true PSs and the final PSs found by MPACO on the specific instance.

Considering that funds are insufficient, we can only choose Scheme 148. Considering that all targets must be allocated and that the cost of missiles should be minimized, we can only choose Scheme 4.

Solving the SMWTA problem is the foundation of the dynamic multiobjective weapon-target assignment problem (DMWTA). The goal of the DMWTA is to provide a set of near-optimal or acceptable real-time decisions in real air combat. So, the time performance of algorithms is also an important index. In the end part, we test four algorithms on the specific instance in 30 runs and record the iteration

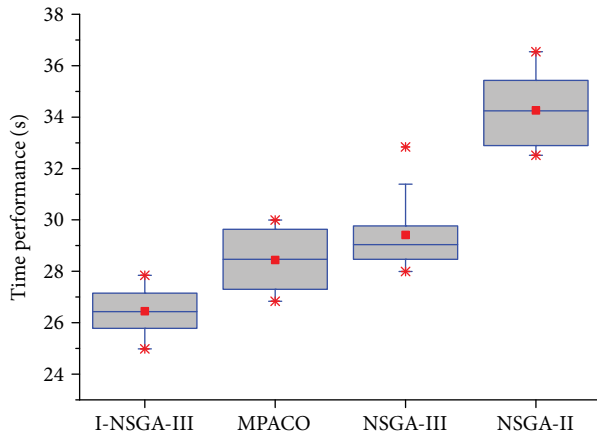


FIGURE 8: Time performance of four different algorithms. * represent extreme outliers.

time of each algorithm. The statistical results of time performances are shown in Figure 8.

In real air combat situations, pilots often make deadly decisions within seconds or even within milliseconds. We can see from Figure 5 that I-NSGA-III has a time advantage in solving the special instance compared with other algorithms, and the time performance of the NSGA-II is worst among the four algorithms. Although improvement of the strategy of reference points affects the iteration rate, an appropriate strategy can be selected by an online operator selection mechanism to improve the mutation efficiency and the quality of solutions. Compared with the original NSGA-III algorithm, the improvement strategy of reference points plays a more important role than the online operator selection mechanism in the time performance field.

In this section, we do some work as follows: Firstly, we use four classic test problems (DTLZ1–4) to evaluate the proposed algorithm and compare it with the other five state-of-the-art algorithms. Secondly, we test the four different algorithms on a specific example, verify the applicability and feasibility of the proposed algorithm, give a comparison study among the four algorithms, and show the corresponding distribution results in Appendix B and Appendix C. Thirdly, we show the time performance of four algorithms in 30 runs. To summarize, I-NSGA-III has been proved to be an effective technique for the SMWTA optimization problem and is obviously the best among the four algorithms.

6. Conclusion

We apply NSGA-III to the WTA problem and propose the I-NSGA-III to solve SMWTA in this paper. The main contributions of the thesis are summarized as follows: on the one hand, the expected damage to the enemy and the cost of missiles are taken into account from a practical viewpoint; in terms of the other objective—the value of fighter combat was introduced to make the model in line with real air combat. In this paper, an improvement strategy of reference points and an online operator selection mechanism are proposed and embedded into the original NSGA-III algorithm to improve the performances of the I-NSGA-III algorithm. The experiments have

shown that I-NSGA-III can find better Pareto solutions than the other three algorithms for the SMWTA problem. More importantly, I-NSGA-III is more suitable for solving the problem from the time performance viewpoint.

However, we have mainly studied the SMWTA problem; few studies have focused on dynamic problems, which are more instructive to real air combat. In recent years, more and more studies have begun to pay attention to DWTA problems. A further study on this topic is one of our future tasks.

Appendix

Appendix A contains three data tables. These three tables come from our previous work [23] and are used to express the data used in the instance to verify the performance of the proposed algorithm. In Table 16, each data represents the damage probability of different missiles attacking different targets. In Table 17, ρ_k is the pilot operation factor and represents the talent, training time, and operation stability of the pilot which may affect the attack performance. In Table 18, c_i represents the cost of i th missile (c_i). The larger the value is, the more the missile costs.

The data in Appendix B represent the results and the corresponding distribution by I-NSGA-III. In each scheme, the first three columns represent the values of the three objective functions, and the latter twelve columns represent the corresponding results of the WTA distribution. (As an example, in Scheme 33, $1/f_1 = 0.07168$, $f_2 = 5.72$, and $1/f_3 = 0.70588$. Missile 2 is assigned to Target 3, Missiles 3 and 4 are assigned to Target 7, Missiles 6 and 10 are assigned to Target 9, Missiles 8 and 12 are assigned to Target 5, and Missile 9 is assigned to Target 1).

In order to compare the statistical results of all algorithms, the statistical results obtained by NSGA-III, MPACO, and NSGA-II are given in Appendix C and shown in Figures 6–8.

A. The Value of the Specific Example Used in This Paper

TABLE 16: Missile damage probability.

Missile unit	Target									
	1	2	3	4	5	6	7	8	9	10
1	0.82	0.85	0.78	0.75	0.52	0.88	0.44	0.76	0.72	0.56
2	0.56	0.72	0.88	0.46	0.64	0.47	0.68	0.45	0.48	0.75
3	0.45	0.62	0.54	0.73	0.84	0.76	0.78	0.42	0.53	0.65
4	0.56	0.42	0.76	0.84	0.73	0.83	0.86	0.62	0.78	0.82
5	0.45	0.58	0.81	0.44	0.63	0.59	0.78	0.77	0.65	0.70
6	0.46	0.61	0.55	0.68	0.75	0.83	0.73	0.66	0.82	0.48
7	0.66	0.71	0.65	0.44	0.86	0.79	0.59	0.85	0.53	0.56
8	0.56	0.42	0.76	0.84	0.73	0.72	0.44	0.75	0.48	0.47
9	0.88	0.78	0.44	0.67	0.56	0.86	0.58	0.65	0.73	0.42
10	0.56	0.88	0.68	0.45	0.75	0.73	0.61	0.76	0.84	0.78
11	0.84	0.54	0.44	0.42	0.65	0.56	0.71	0.55	0.45	0.88
12	0.83	0.76	0.84	0.62	0.82	0.75	0.42	0.68	0.57	0.54

TABLE 17: Pilot operation factor table.

Fighter	F_1			F_2			F_3			F_4		
Missile unit	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}
ρ_k	0.96			0.95			0.98			0.93		

TABLE 18: The cost of each missile.

Missile unit	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}
c_i	0.62	0.63	0.69	0.80	0.72	0.90	0.96	0.68	0.72	0.65	0.66	0.65

B. The Value of PSs and Results of WTA Distribution

TABLE 19

	$1/f_1$	f_2	$1/f_3$	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}
1	0.06822	8.68000	100.00000	9	3	5	7	7	9	5	5	1	9	7	3
2	0.06860	8.02000	3.00000	9	3	7	7	7	9	5	3	1	9	0	5
3	0.06853	8.00000	4.00000	9	3	5	7	7	9	5	0	1	9	7	3
4	0.06867	7.96000	2.00000	9	3	7	7	0	9	5	3	1	9	7	5
5	0.06879	7.88000	2.00000	9	3	7	0	7	9	5	3	1	9	7	5
6	0.06880	7.72000	4.00000	9	3	5	7	7	9	0	3	1	9	7	5
7	0.06912	7.34000	1.71429	9	3	5	7	7	9	5	0	1	9	0	3
8	0.06917	7.37000	1.50000	9	3	5	7	7	9	5	3	1	9	0	0
9	0.06924	7.30000	1.20000	9	3	7	7	0	9	5	3	1	9	0	5
10	0.06936	7.12000	1.71429	9	3	5	7	7	0	5	3	1	9	0	9
11	0.06956	7.27000	1.20000	9	3	0	7	0	9	5	3	1	9	7	5
12	0.06956	7.06000	1.33333	9	3	7	7	0	0	5	3	1	9	7	5
13	0.06957	7.23000	1.20000	9	3	7	0	7	9	5	3	1	0	7	5
14	0.06961	7.22000	1.20000	9	3	7	0	7	9	5	3	1	9	0	5
15	0.06973	7.16000	1.00000	9	3	7	0	0	7	5	3	1	9	7	5
16	0.06987	6.92000	1.33333	9	3	5	0	7	7	0	3	1	9	7	5
17	0.06991	6.69000	1.09091	9	3	5	7	7	9	5	0	1	0	0	3
18	0.06996	6.72000	1.00000	9	3	5	7	7	9	5	3	1	0	0	0
19	0.06997	6.82000	2.00000	9	3	5	7	9	0	0	5	1	9	7	3
20	0.07002	6.44000	1.20000	9	3	5	7	7	0	5	0	1	9	0	3
21	0.07004	6.65000	0.85714	9	3	7	7	0	9	5	3	1	0	0	5
22	0.07015	6.40000	0.92308	9	3	7	7	0	0	5	3	1	9	0	5
23	0.07041	6.57000	0.85714	9	3	7	0	7	9	5	3	1	0	0	5
24	0.07053	6.32000	0.92308	9	3	7	0	7	0	5	3	1	9	0	5
25	0.07056	6.35000	0.92308	9	3	5	7	0	9	0	5	1	0	7	3
26	0.07060	6.10000	1.00000	9	3	5	7	0	0	0	3	1	9	7	5
27	0.07087	6.51000	0.75000	9	3	7	0	0	9	5	5	1	0	7	3
28	0.07094	6.26000	0.80000	9	3	7	0	0	0	5	3	1	9	7	5
29	0.07130	6.00000	0.66667	9	3	7	7	0	9	5	5	1	0	0	0
30	0.07142	5.75000	0.70588	9	3	7	7	0	0	5	5	1	9	0	0
31	0.07152	5.94000	0.70588	9	3	0	7	0	9	5	0	1	0	7	5
32	0.07165	5.67000	0.75000	9	3	5	7	0	9	0	0	1	0	7	5
33	0.07168	5.72000	0.70588	0	3	7	7	0	9	0	5	1	9	0	5
34	0.07169	5.92000	0.66667	9	3	7	0	7	9	5	5	1	0	0	0

TABLE 19: Continued.

	$1/f_1$	f_2	$1/f_3$	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}
35	0.07192	5.45000	0.75000	9	3	5	7	0	0	0	5	1	9	7	0
36	0.07200	5.83000	0.63158	9	3	7	0	0	9	5	0	1	0	7	5
37	0.07211	5.41000	0.75000	9	3	0	7	0	0	0	5	1	9	7	5
38	0.07212	5.58000	0.66667	9	3	7	0	0	0	5	0	1	9	7	5
39	0.07224	5.61000	0.63158	9	3	7	0	0	0	5	5	1	9	7	0
40	0.07247	5.55000	0.63158	9	3	7	0	0	9	0	5	1	0	7	5
41	0.07278	5.88000	0.60000	9	0	7	0	0	9	5	5	1	0	7	3
42	0.07306	5.32000	0.57143	9	3	7	7	0	9	5	0	1	0	0	0
43	0.07319	5.07000	0.60000	9	3	7	7	0	0	5	0	1	9	0	0
44	0.07341	5.29000	0.57143	9	3	0	7	0	9	5	0	1	0	7	0
45	0.07347	5.24000	0.57143	9	3	7	0	7	9	5	0	1	0	0	0
46	0.07374	5.17000	0.52174	9	3	7	0	0	7	5	0	1	9	0	0
47	0.07400	5.02000	0.60000	9	3	5	7	0	9	0	0	1	0	7	0
48	0.07404	4.93000	0.54545	9	3	7	0	0	0	5	0	1	9	7	0
49	0.07413	4.77000	0.63158	9	3	5	7	0	0	0	0	1	9	7	0
50	0.07451	4.79000	0.60000	9	3	5	7	0	0	0	5	1	9	0	0
51	0.07490	4.87000	0.54545	9	3	5	0	0	7	0	0	1	9	7	0
52	0.07502	5.23000	0.50000	9	0	7	0	0	9	5	3	1	0	7	0
53	0.07520	4.95000	0.52174	9	7	7	0	0	0	5	3	1	9	0	0
54	0.07522	4.81000	0.60000	9	0	5	7	0	0	0	5	1	9	0	3
55	0.07538	4.62000	0.57143	9	3	7	0	0	0	0	0	1	9	7	5
56	0.07611	4.63000	0.48000	9	3	0	7	0	9	5	0	1	0	0	0
57	0.07625	4.38000	0.50000	9	3	0	7	0	0	5	0	1	9	0	0
58	0.07674	4.36000	0.50000	9	3	5	7	0	9	0	0	1	0	0	0
59	0.07689	4.11000	0.52174	9	3	5	7	0	0	0	0	1	9	0	0
60	0.07728	4.68000	0.46154	9	0	0	7	0	9	5	3	1	0	0	0
61	0.07742	4.04000	0.54545	9	3	5	7	0	0	0	0	0	9	0	1
62	0.07772	4.52000	0.44444	9	3	7	0	0	9	5	0	1	0	0	0
63	0.07786	4.27000	0.46154	9	3	7	0	0	0	5	0	1	9	0	0
64	0.07809	4.16000	0.50000	9	0	5	7	0	0	0	3	1	9	0	0
65	0.07895	4.57000	0.42857	9	0	7	0	0	9	5	3	1	0	0	0
66	0.07910	4.32000	0.44444	9	0	7	0	0	0	5	3	1	9	0	0
67	0.07920	4.21000	0.46154	9	3	7	0	0	9	0	0	1	0	0	5
68	0.07987	3.90000	0.50000	9	3	7	0	0	0	0	0	0	9	1	5
69	0.08025	3.99000	0.46154	9	3	7	0	0	0	0	5	1	9	0	0
70	0.08042	3.97000	0.48000	9	3	7	0	0	0	0	0	1	0	7	5
71	0.08140	3.73000	0.42857	9	3	0	7	0	0	5	0	1	0	0	0
72	0.08235	3.39000	0.46154	9	3	5	7	0	0	0	0	0	9	0	0
73	0.08269	3.90000	0.38710	0	3	7	0	0	9	5	0	1	0	0	0
74	0.08275	3.78000	0.41379	9	0	0	7	0	0	5	3	1	0	0	0
75	0.08308	3.60000	0.44444	9	3	0	7	0	9	0	0	0	0	0	5
76	0.08325	3.62000	0.40000	9	3	7	0	0	0	5	0	1	0	0	0
77	0.08347	3.55000	0.41379	9	3	7	0	0	0	5	0	0	9	0	0
78	0.08351	3.51000	0.42857	9	0	5	7	0	0	0	3	1	0	0	0
79	0.08372	3.44000	0.44444	9	0	5	7	0	0	0	3	0	9	0	0
80	0.08408	3.95000	0.37500	0	0	7	0	0	9	5	3	1	0	0	0
81	0.08424	3.85000	0.40000	0	0	7	0	0	9	5	0	0	9	0	3
82	0.08436	3.67000	0.40000	0	0	7	0	0	0	5	0	1	9	0	3
83	0.08472	3.85000	0.38710	9	0	7	0	0	9	5	3	0	0	0	0
84	0.08491	3.49000	0.41379	1	3	7	0	0	9	0	0	0	0	0	5
85	0.08518	3.24000	0.42857	9	3	7	0	0	0	0	0	0	9	0	5

TABLE 19: Continued.

	$1/f_1$	f_2	$1/f_3$	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}
86	0.08599	3.34000	0.40000	9	3	7	0	0	0	0	5	1	0	0	0
87	0.08693	3.29000	0.37500	0	3	0	7	0	9	5	0	0	0	0	0
88	0.08755	3.01000	0.38710	9	3	0	7	0	0	5	0	0	0	0	0
89	0.08840	2.74000	0.40000	9	3	5	7	0	0	0	0	0	0	0	0
90	0.08904	3.18000	0.35294	0	3	7	0	0	9	5	0	0	0	0	0
91	0.08911	3.06000	0.37500	9	0	0	7	0	0	5	3	0	0	0	0
92	0.08964	2.79000	0.40000	0	0	5	7	0	0	0	0	0	9	0	3
93	0.08999	2.79000	0.38710	9	0	5	7	0	0	0	3	0	0	0	0
94	0.09133	2.95000	0.35294	9	0	7	0	0	0	5	3	0	0	0	0
95	0.09158	2.98000	0.35294	0	0	7	0	0	0	5	3	0	9	0	0
96	0.09169	2.84000	0.36364	9	3	5	0	0	7	0	0	0	0	0	0
97	0.09267	2.92000	0.35294	0	0	7	0	0	9	0	3	0	0	0	5
98	0.09288	2.62000	0.36364	9	3	7	0	0	0	0	5	0	0	0	0
99	0.09365	2.67000	0.36364	0	0	7	0	0	0	0	3	0	9	0	5
100	0.09366	3.20000	0.34286	0	0	0	0	0	9	5	3	0	0	7	0
101	0.09528	2.87000	0.35294	0	3	0	0	0	9	0	5	0	0	7	0
102	0.09603	2.59000	0.36364	9	3	0	0	0	0	0	5	0	0	7	0
103	0.09658	2.61000	0.36364	9	0	0	0	0	0	0	3	0	0	7	5
104	0.09662	2.74000	0.35294	0	0	7	0	0	0	0	3	9	0	0	5
105	0.09706	2.55000	0.37500	3	7	0	0	0	0	0	0	0	9	0	5
106	0.09733	2.93000	0.34286	0	3	0	0	0	7	0	5	9	0	0	0
107	0.09777	2.98000	0.34286	9	0	0	0	0	0	5	3	7	0	0	0
108	0.09789	2.71000	0.35294	3	0	7	0	0	0	0	5	9	0	0	0
109	0.09955	2.65000	0.35294	9	3	0	0	0	0	0	5	7	0	0	0
110	0.10275	2.38000	0.34286	9	0	0	7	0	0	5	0	0	0	0	0
111	0.10392	2.11000	0.35294	9	0	5	7	0	0	0	0	0	0	0	0
112	0.10424	2.14000	0.35294	0	0	5	7	0	0	0	0	0	9	0	0
113	0.10445	2.35000	0.34286	0	0	0	7	0	9	0	0	0	0	0	5
114	0.10571	2.27000	0.32432	9	0	7	0	0	0	5	0	0	0	0	0
115	0.10695	2.10000	0.34286	9	0	0	7	0	0	0	5	0	0	0	0
116	0.10779	2.05000	0.35294	9	5	0	7	0	0	0	0	0	0	0	0
117	0.10850	2.21000	0.32432	9	0	5	0	0	7	0	0	0	0	0	0
118	0.10883	1.99000	0.33333	0	0	7	0	0	0	0	0	0	9	0	5
119	0.10884	2.52000	0.31579	0	0	0	0	0	9	5	0	0	0	7	0
120	0.11016	1.99000	0.32432	9	0	7	0	0	0	0	5	0	0	0	0
121	0.11105	1.94000	0.33333	9	5	7	0	0	0	0	0	0	0	0	0
122	0.11152	2.58000	0.30769	0	0	0	0	0	7	5	0	9	0	0	0
123	0.11280	1.93000	0.33333	9	0	0	0	0	0	0	0	0	0	7	5
124	0.11319	1.90000	0.33333	9	7	0	0	0	0	0	0	0	0	0	5
125	0.11357	2.24000	0.31579	0	0	0	0	0	9	0	5	0	0	7	0
126	0.11463	1.96000	0.32432	9	0	0	0	0	0	0	5	0	0	7	0
127	0.11503	1.93000	0.32432	9	7	0	0	0	0	0	5	0	0	0	0
128	0.11560	1.91000	0.33333	9	5	0	0	0	0	0	0	0	0	7	0
129	0.11648	2.30000	0.30769	0	0	0	0	0	7	0	5	9	0	0	0
130	0.11774	1.92000	0.33333	9	0	0	0	0	0	0	0	0	7	0	5
131	0.11968	2.02000	0.31579	9	0	0	0	0	0	0	5	7	0	0	0
132	0.13725	1.70000	0.30769	0	0	0	7	0	9	0	0	0	0	0	0
133	0.13881	1.42000	0.31579	9	0	0	7	0	0	0	0	0	0	0	0
134	0.14259	1.59000	0.29268	0	0	7	0	0	9	0	0	0	0	0	0
135	0.14490	1.34000	0.30000	0	0	7	0	0	0	0	0	0	9	0	0
136	0.15017	1.56000	0.29268	0	0	0	0	0	9	0	0	0	0	7	0

TABLE 19: Continued.

	$1/f_1$	f_2	$1/f_3$	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}
137	0.15204	1.28000	0.30000	9	0	0	0	0	0	0	0	0	0	7	0
138	0.15274	1.25000	0.30000	9	7	0	0	0	0	0	0	0	0	0	0
139	0.15276	1.86000	0.28571	0	0	0	0	0	9	5	0	0	0	0	0
140	0.15531	1.62000	0.28571	0	0	0	0	0	7	0	0	9	0	0	0
141	0.15858	1.55000	0.29268	0	0	0	0	0	9	0	0	0	0	0	5
142	0.16067	1.27000	0.30000	9	0	0	0	0	0	0	0	0	0	0	5
143	0.16082	1.38000	0.29268	0	0	0	0	0	0	0	0	9	0	7	0
144	0.16223	1.58000	0.28571	0	0	0	0	0	9	0	5	0	0	0	0
145	0.16441	1.30000	0.29268	9	0	0	0	0	0	0	5	0	0	0	0
146	0.17473	1.40000	0.28571	0	0	0	0	0	0	0	5	9	0	0	0
147	0.24888	0.90000	0.26667	0	0	0	0	0	9	0	0	0	0	0	0
148	0.25407	0.62000	0.27273	9	0	0	0	0	0	0	0	0	0	0	0
149	0.27956	0.72000	0.26667	0	0	0	0	0	0	0	0	9	0	0	0
150	0.42517	0.68000	0.26667	0	0	0	0	0	0	0	9	0	0	0	0

C. The Statistical Results of NSGA-III, MPACO, and NSGA-II

TABLE 20

NSGA-III	$1/f_1$	f_2	$1/f_3$
1	0.06952	8.68	100
2	0.07019	7.72	4
3	0.07036	8.02	3
4	0.07093	6.82	2
5	0.07122	7.37	1.5
6	0.07145	6.47	1.09091
7	0.0719	7.16	1
8	0.07208	6.02	1
9	0.07256	6.2	0.8
10	0.07341	5.3	0.66667
11	0.0741	6.48	0.63158
12	0.07627	4.62	0.57143
13	0.07792	5.3	0.54545
14	0.07802	5.6	0.52174
15	0.07985	4.87	0.54545
16	0.08105	5.85	0.5
17	0.0817	4.95	0.44444
18	0.08333	3.97	0.48
19	0.08402	4.27	0.4
20	0.08459	3.89	0.48
21	0.0884	3.21	0.42857
22	0.09031	4.21	0.4
23	0.09554	3.56	0.35294
24	0.09752	3.36	0.4
25	0.10173	3.36	0.35294
26	0.10595	3.67	0.34286
27	0.10874	3.55	0.34286
28	0.10988	2.99	0.31579

TABLE 20: Continued.

NSGA-III	$1/f_1$	f_2	$1/f_3$
29	0.11019	2.89	0.34286
30	0.1147	2.71	0.31579
31	0.11756	2.65	0.32432
32	0.12685	2.55	0.33333
33	0.13384	2.24	0.31579
34	0.15216	2.03	0.29268
35	0.16145	1.92	0.3
36	0.17051	1.99	0.29268
37	0.17871	1.27	0.3
38	0.18988	1.34	0.29268
39	0.25602	1.27	0.27273
40	0.27956	1.34	0.26667
41	0.1	7.8	42.5
42	0.1	7.2	33.5
43	0.1	6.8	27.5
44	0.1	6.2	18.5
45	0.1	5.6	12.5
46	0.1	5	7.1
47	0.16	4.4	4.675
48	0.16	3.8	2.975
49	0.1	4.4	5.9
50	0.16	3.2	1.775
51	0.1	3.8	5.6625
52	0.1	3.2	5.6125
53	0.22	3.4	2.6125
54	0.22	2.8	0.625
55	0.37577	0.996	3.34975
56	0.06623	7.3388	3.34975

TABLE 20: Continued.

NSGA-III	$1/f_1$	f_2	$1/f_3$
57	0.24409	2.1668	3.34975
58	0.20791	2.7876	3.34975
59	0.17189	3.5728	3.34975
60	0.13616	4.561	3.34975
61	0.10086	5.7982	3.34975
62	0.20791	2.7876	3.34975
63	0.20791	8.7876	3.34975
64	0.38368	2.3574	3.34975
65	0.35316	0.996	3.34975
66	0.28033	1.678	3.34975
67	0.0352	9.9994	3.34975
68	0.40638	8.8566	3.34975
69	0.38431	8.337	3.34975
70	0.38979	8.547	3.34975
71	0.32185	2.349	3.34975
72	0.03548	9.6992	3.34975
73	0.13622	4.5612	3.34988
74	0.13634	4.5616	3.35023
75	0.03543	9.6266	3.56358
76	0.3168	1.295	3.57137
77	0.14243	4.0068	3.60527
78	0.13617	4.561	3.61382
79	0.28181	1.8256	3.70112
80	0.06495	9.9334	3.74135
81	0.05946	9.998	3.77945
82	0.03234	9.2458	3.78025
83	0.28132	1.678	3.7807
84	0.35609	1.433	3.8949
85	0.38276	2.0834	4.91208
86	0.05692	9.7142	5.08312
87	0.14241	6.1792	5.13159
88	0.3838	8.7272	5.54011
89	0.31968	1.9758	6.90731
90	0.37994	7.7678	8.09498
91	0.38328	2.2392	8.89547
92	0.38012	1.3922	9.5236
93	0.40638	8.8566	9.88592
94	0.38335	8.6474	61.00714
95	0.35273	0.996	86.38796

TABLE 21

MPACO	$1/f_1$	f_2	$1/f_3$
1	0.06952	8.68	100
2	0.07256	6.2	0.8
3	0.07341	5.3	0.66667
4	0.0741	6.48	0.63158
5	0.07627	4.62	0.57143
6	0.07792	5.3	0.54545

TABLE 21: Continued.

MPACO	$1/f_1$	f_2	$1/f_3$
7	0.07802	5.6	0.52174
8	0.07985	4.87	0.54545
9	0.08105	5.85	0.5
10	0.09752	3.36	0.4
11	0.10173	3.36	0.35294
12	0.10595	3.67	0.34286
13	0.10874	3.55	0.34286
14	0.10988	2.99	0.31579
15	0.11019	2.89	0.34286
16	0.1147	2.71	0.31579
17	0.17051	1.99	0.29268
18	0.17871	1.27	0.3
19	0.18988	1.34	0.29268
20	0.25602	1.27	0.27273
21	0.27956	1.34	0.26667
22	0.03346	9.2458	3.34975
23	0.35273	0.9962	3.35025
24	0.15292	9.0056	3.56627
25	0.2462	2.3204	3.64173
26	0.20956	3.0006	3.67611
27	0.31718	1.437	3.77513
28	0.24848	2.4014	3.79922
29	0.28457	1.876	3.82353
30	0.17966	4.1672	4.10582
31	0.25343	6.60826	4.11647
32	0.13914	2.5606	4.14042
33	0.28639	2.0884	4.35678
34	0.14175	3.698	4.46223
35	0.14277	2.987	4.51727
36	0.08449	2.43	4.55926
37	0.17589	4.6238	4.7343
38	0.08002	2.0406	4.7849
39	0.14984	5.929	4.83138
40	0.12355	7.461	4.8318
41	0.14186	6.0588	4.98583
42	0.28437	2.3816	5.13553
43	0.18122	4.935	5.1871
44	0.21446	3.8735	5.25744
45	0.21328	3.9618	5.30667
46	0.24787	9.3158	5.31994
47	0.19132	5.1524	5.51549
48	0.32606	3.1232	5.76914
49	0.21922	4.3486	6.03888
50	0.17991	5.6686	6.33581
51	0.1799	5.6758	6.34759
52	0.32229	2.2544	6.54919
53	0.29146	5.6102	6.90432
54	0.32349	2.3852	7.04616
55	0.32211	2.4028	7.11377
56	0.32594	2.9264	9.26279

TABLE 21: Continued.

MPACO	$1/f_1$	f_2	$1/f_3$
57	0.36071	2.9998	13.2611
58	0.37503	6.6086	45.73848
59	0.37731	7.084	51.68306
60	0.34764	9.1058	56.91861
61	0.34774	9.1278	57.17808
62	0.34857	5.3214	59.41081
63	0.38565	8.1212	66.20195

TABLE 22

NSGA-II	$1/f_1$	f_2	$1/f_3$
1	0.1909	5.455	5.98965
2	0.145	6.875	6.01718
3	0.2138	4.338	6.01943
4	0.2458	2.358	3.71603
5	0.2866	5.658	4.24055
6	0.2113	5.455	4.83263
7	0.0840	4.27	0.4
8	0.0845	3.89	0.48
9	0.1338	2.24	0.31579
10	0.1521	2.03	0.29268
11	0.1009	5.7982	61.439
12	0.4026	7.8266	61.853
13	0.4026	7.8266	61.853
14	0.4030	7.931	63.3753
15	0.4033	8.013	64.5881
16	0.3863	9.349	64.9901
17	0.404	8.212	67.5876
18	0.3515	9.987	67.8457
19	0.4041	8.236	67.9489
20	0.4034	8.04	69.4217
21	0.0686	7.9656	69.50514
22	0.39702	9.0474	72.81714
23	0.40559	8.6368	74.26286
24	0.4045	8.3316	74.43411
25	0.39136	9.1984	75.73334
26	0.40568	8.661	77.74505
27	0.3862	9.349	77.8771
28	0.0569	9.64	77.8771
29	0.4068	8.971	79.7996
30	0.2171	2.787	81.1023
31	0.4075	9.169	83.2120
32	0.38	7.797	83.7143
33	0.3862	9.349	86.3879
34	0.3999	8.848	86.3879
35	0.38637	9.349	86.3879
36	0.10173	3.36	0.35294
37	0.10595	3.67	0.34286

TABLE 22: Continued.

NSGA-II	$1/f_1$	f_2	$1/f_3$
38	0.10874	3.55	0.34286
39	0.10988	2.99	0.31579

Conflicts of Interest

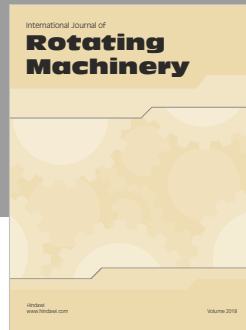
The authors declare that there is no conflict of interest regarding the publication of this paper.

References

- [1] R. N. Rai and N. Bolia, "Optimal decision support for air power potential," *IEEE Transactions on Engineering Management*, vol. 61, no. 2, pp. 310–322, 2014.
- [2] A. S. Manne, "A target-assignment problem," *Operations Research*, vol. 6, no. 3, pp. 346–351, 1958.
- [3] R. H. Day, "Allocating weapons to target complexes by means of nonlinear programming," *Operations Research*, vol. 14, no. 6, pp. 992–1013, 1966.
- [4] P. A. Hosein and M. Athans, *Preferential Defense Strategies. Part I: The Static Case*, MIT Laboratory for Information and Decision Systems, Cambridge, MA, USA, 1990.
- [5] D. Galati and M. Simaan, "Effectiveness of the Nash strategies in competitive multi-team target assignment problems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 1, pp. 126–134, 2007.
- [6] Z. J. Lee, C. Y. Lee, and S. F. Su, "An immunity-based ant colony optimization algorithm for solving weapon–target assignment problem," *Applied Soft Computing*, vol. 2, no. 1, pp. 39–47, 2002.
- [7] Z. J. Lee, S. F. Su, and C. Y. Lee, "A genetic algorithm with domain knowledge for weapon-target assignment problems," *Journal of the Chinese Institute of Engineers*, vol. 25, no. 3, pp. 287–295, 2002.
- [8] 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*, pp. 278–285, Springer, Berlin, Heidelberg, 2003.
- [9] M.-Z. Lee, "Constrained weapon–target assignment: enhanced very large scale neighborhood search algorithm," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 1, pp. 198–204, 2010.
- [10] 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 Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 6, pp. 649–662, 2010.
- [11] 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.
- [12] P. Chen, B.-j. Shen, L.-s. Zhou, and Y.-w. Chen, "Optimized simulated annealing algorithm for thinning and weighting large planar arrays," *Journal of Zhejiang University SCIENCE C*, vol. 11, no. 4, pp. 261–269, 2010.

- [13] 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 Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 3, pp. 598–606, 2011.
- [14] A. G. Fei, L. Y. Zhang, and Q. J. Ding, "Multi-aircraft cooperative fire assignment based on auction algorithm," *Systems Engineering and Electronics*, vol. 34, no. 9, pp. 1829–1833, 2012.
- [15] Z. R. Bogdanowicz, A. Tolano, K. Patel, and N. P. Coleman, "Optimization of weapon-target pairings based on kill probabilities," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1835–1844, 2013.
- [16] X. Liu, Z. Liu, W. S. Hou, and J. H. Xu, "Improved MOPSO algorithm for multi-objective programming model of weapon-target assignment," *Systems Engineering and Electronics*, vol. 35, no. 2, pp. 326–330, 2013.
- [17] Y. Zhang, R. N. Yang, J. L. Zuo, and X. Jing, "Weapon-target assignment based on decomposition-based evolutionary multi-objective optimization algorithms," *Systems Engineering and Electronics*, vol. 36, no. 12, pp. 2435–2441, 2014.
- [18] D. K. Ahner and C. R. Parson, "Optimal multi-stage allocation of weapons to targets using adaptive dynamic programming," *Optimization Letters*, vol. 9, no. 8, pp. 1689–1701, 2015.
- [19] J. Li, J. Chen, B. Xin, and L. H. Dou, "Solving multi-objective multi-stage weapon target assignment problem via adaptive NSGA-II and adaptive MOEA/D: A comparison study," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, pp. 3132–3139, Sendai, Japan, May 2015.
- [20] N. Dirik, S. N. Hall, and J. T. Moore, "Maximizing strike aircraft planning efficiency for a given class of ground targets," *Optimization Letters*, vol. 9, no. 8, pp. 1729–1748, 2015.
- [21] L. Hongtao and K. Fengju, "Adaptive chaos parallel clonal selection algorithm for objective optimization in WTA application," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 6, pp. 3459–3465, 2016.
- [22] N. Li, W. Huai, and S. Wang, "The solution of target assignment problem in command and control decision-making behaviour simulation," *Enterprise Information Systems*, vol. 11, pp. 1–19, 2017.
- [23] Y. Li, Y. Kou, Z. Li, A. Xu, and Y. Chang, "A modified Pareto ant colony optimization approach to solve biobjective weapon-target assignment problem," *International Journal of Aerospace Engineering*, vol. 2017, Article ID 1746124, 14 pages, 2017.
- [24] 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 Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [25] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Behavior of evolutionary many-objective optimization," in *Tenth International Conference on Computer Modeling and Simulation (uksim 2008)*, pp. 266–271, Cambridge, UK, April 2008.
- [26] 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 Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.
- [27] W. Mkaouer, M. Kessentini, A. Shaout et al., "Many-objective software remodularization using NSGA-III," *ACM Transactions on Software Engineering and Methodology*, vol. 24, no. 3, pp. 1–45, 2015.
- [28] F. Chen, J. Zhou, C. Wang, C. Li, and P. Lu, "A modified gravitational search algorithm based on a non-dominated sorting genetic approach for hydro-thermal-wind economic emission dispatching," *Energy*, vol. 121, pp. 276–291, 2017.
- [29] Y. Zhu, J. Liang, J. Chen, and Z. Ming, "An improved NSGA-III algorithm for feature selection used in intrusion detection," *Knowledge-Based Systems*, vol. 116, pp. 74–85, 2017.
- [30] H. Seada and K. Deb, "U-NSGA-III: a unified evolutionary optimization procedure for single, multiple, and many objectives: proof-of-principle results," in *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 34–49, Guimarães, Portugal, 2015.
- [31] Y. Yuan, H. Xu, and B. Wang, "An improved NSGA-III procedure for evolutionary many-objective optimization," in *Proceedings of the 2014 conference on Genetic and evolutionary computation - GECCO '14*, pp. 661–668, New York, NY USA, July 2014.
- [32] K. Deb and J. Sundar, "Reference point based multi-objective optimization using evolutionary algorithms," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation - GECCO '06*, pp. 635–642, Seattle, WA, USA, July 2006.
- [33] R. M. F. Alves and C. R. Lopes, "Using genetic algorithms to minimize the distance and balance the routes for the multiple traveling salesman problem," *2015 IEEE Congress on Evolutionary Computation (CEC)*, 2015, pp. 3171–3178, Sendai, Japan, May 2015.
- [34] D. W. Gong, X. F. Ji, and X. Y. Sun, "Solving many-objective optimization problems using set-based evolutionary algorithms," *Acta Electronica Sinica*, vol. 42, no. 1, pp. 77–83, 2014.
- [35] J. Bader, D. Brockhoff, S. Welten, and E. Zitzler, "On using populations of sets in multiobjective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 140–154, Nantes, France, April 2009.
- [36] E. Zitzler, L. Thiele, and J. Bader, "On set-based multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 58–79, 2010.
- [37] D. Gong, J. Sun, and Z. Miao, "A set-based genetic algorithm for interval many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 47–60, 2018.
- [38] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 28, no. 3, pp. 392–403, 1998.
- [39] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 204–223, 2003.
- [40] H. Ishibuchi and K. Narukawa, "Performance evaluation of simple multiobjective genetic local search algorithms on multi-objective 0/1 knapsack problems," in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, pp. 441–448, Portland, OR, USA, June 2004.
- [41] J. D. Knowles and D. W. Corne, "M-PAES: a memetic algorithm for multiobjective optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, pp. 325–332, La Jolla, CA, USA, July 2000.

- [42] K. Harada, J. Sakuma, K. Ikeda, I. Ono, and S. Kobayashi, "Local search for multiobjective function optimization: Pareto descent method," *Transactions of the Japanese Society for Artificial Intelligence*, vol. 21, no. 4, pp. 350–360, 2006.
- [43] P. A. N. Bosman, "On gradients and hybrid evolutionary algorithms for real-valued multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 51–69, 2012.
- [44] K. Sindhya, K. Deb, and K. Miettinen, "A local search based evolutionary multi-objective optimization approach for fast and accurate convergence," in *International Conference on Parallel Problem Solving from Nature*, pp. 815–824, Dortmund, Germany, September 2008.
- [45] L. V. Santana-Quintero, A. G. Hernández-Díaz, J. Molina, C. A. Coello Coello, and R. Caballero, "DEMORS: a hybrid multi-objective optimization algorithm using differential evolution and rough set theory for constrained problems," *Computers & Operations Research*, vol. 37, no. 3, pp. 470–480, 2010.
- [46] A. Jaskiewicz, "Genetic local search for multi-objective combinatorial optimization," *European Journal of Operational Research*, vol. 137, no. 1, pp. 50–71, 2002.
- [47] M. Abouhawwash, H. Seada, and K. Deb, "Towards faster convergence of evolutionary multi-criterion optimization algorithms using Karush Kuhn Tucker optimality based local search," *Computers & Operations Research*, vol. 79, pp. 331–346, 2017.
- [48] A. P. Wierzbicki, "The use of reference objectives in multiobjective optimization," in *Multiple criteria decision making theory and application*, pp. 468–486, Springer, Berlin Heidelberg, 1980.
- [49] A. Mohammadi, M. N. Omidvar, and X. Li, "Reference point based multi-objective optimization through decomposition," in *2012 IEEE Congress on Evolutionary Computation*, pp. 1–8, Brisbane, QLD, Australia, June 2012.
- [50] J. R. Figueira, A. Liefooghe, E. G. Talbi, and A. P. Wierzbicki, "A parallel multiple reference point approach for multi-objective optimization," *European Journal of Operational Research*, vol. 205, no. 2, pp. 390–400, 2010.
- [51] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 474–494, 2013.
- [52] Y. Liu, D. Gong, X. Sun, and Y. Zhang, "Many-objective evolutionary optimization based on reference points," *Applied Soft Computing*, vol. 50, pp. 344–355, 2017.
- [53] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International Conference on Parallel Problem Solving from Nature*, pp. 832–842, Birmingham, UK, September 2004.
- [54] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.
- [55] M. Wagner and F. Neumann, "A fast approximation-guided evolutionary multi-objective algorithm," in *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference - GECCO '13*, pp. 687–694, Amsterdam, Netherlands, July 2013.
- [56] R. H. Gómez and C. A. Coello Coello, "Improved metaheuristic based on the R2 indicator for many-objective optimization," in *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference - GECCO '15*, pp. 679–686, Madrid, Spain, July 2015.
- [57] Y. Liu, D. Gong, J. Sun, and Y. Jin, "A many-objective evolutionary algorithm using a one-by-one selection strategy," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2689–2702, 2017.
- [58] I. Das and J. E. Dennis, "Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multi-criteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.
- [59] K. Li, Á. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 114–130, 2014.
- [60] W. Gong, Á. Fialho, Z. Cai, and H. Li, "Adaptive strategy selection in differential evolution for numerical optimization: an empirical study," *Information Sciences*, vol. 181, no. 24, pp. 5364–5386, 2011.
- [61] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 3, pp. 115–148, 1995.
- [62] S. Biswas, S. Kundu, and S. Das, "Inducing niching behavior in differential evolution through local information sharing," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 246–263, 2015.
- [63] K. Sindhya, S. Ruuska, T. Haanpää, and K. Miettinen, "A new hybrid mutation operator for multiobjective optimization with differential evolution," *Soft Computing*, vol. 15, no. 10, pp. 2041–2055, 2011.
- [64] X. Bi and C. Wang, "An improved NSGA-III algorithm based on objective space decomposition for many-objective optimization," *Soft Computing*, vol. 21, no. 15, pp. 4269–4296, 2017.
- [65] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [66] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [67] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*, Advanced Information and Knowledge Processing, A. Abraham, L. Jain, and R. Goldberg, Eds., pp. 105–145, Springer, London, 2005.
- [68] Y. Yuan, H. Xu, and B. Wang, "An experimental investigation of variation operators in reference-point based many-objective optimization," in *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference - GECCO '15*, pp. 775–782, New York, NY, USA, July 2015.
- [69] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.



Hindawi

Submit your manuscripts at
www.hindawi.com

