

Research Article

NAAM-MOEA/D-Based Multitarget Firepower Resource Allocation Optimization in Edge Computing

Liyuan Deng,¹ Ping Yang,¹ Weidong Liu,¹ Lina Wang,² Sifeng Wang,² and Xiumei Zhang³ 

¹*Xi'an Research Institute of High-Technology, China*

²*School of Computer Science, Qufu Normal University, China*

³*School of Computer Science and Software Engineering, University of Science and Technology Liaoning, China*

Correspondence should be addressed to Xiumei Zhang; aszxm2002@126.com

Received 11 January 2021; Revised 9 February 2021; Accepted 27 February 2021; Published 22 March 2021

Academic Editor: Mohammad R. Khosravi

Copyright © 2021 Liyuan Deng 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.

In the edge environment, the multiobjective evolutionary algorithm based on decomposition (MOEA/D) has been widely used in the research of multitarget firepower resource allocation. However, as the MOEA/D algorithm uses a fixed neighborhood update mechanism, it is impossible to rationally allocate computing resources based on the difficulty of each subproblem optimization, which results in some problems such as reduced population evolution efficiency and poor evolution quality during the calculation process. In order to solve these problems, a decision mechanism for subproblems and population evolution stages is designed, and on this basis, a MOEA/D algorithm based on the neighborhood adaptive adjustment mechanism is proposed to adapt to the edge environment. The optimization model of multiobjective firepower resource allocation based on the maximization of damage effect and the minimization of strike cost is constructed and solved. Using the ZDT series of test functions for comparative experiments, the simulation results show that the proposed algorithm can balance the distribution and convergence of population evolution and obtain satisfactory optimization results.

1. Introduction

In the edge environment, due to the limited computing resources of edge clients, the allocation of firepower resources based on factors such as battlefield situation, weapon performance, and combat objectives reasonably deploying and allocating various types and quantities of weapons and equipment to obtain the best combat effect is an important part of combat planning [1]. The firepower resource allocation optimization problem in edge environment usually constructs a single-objective firepower resource allocation optimization model based on the damage probability objective function, using heuristic genetic algorithm [2], simulated annealing genetic algorithm [3], particle swarm algorithm [4], and ant colony algorithm to solve

the model. In practical problems, the objective function that only considers the single factor of damage probability is obviously not realistic. Literature [5] establishes interception benefit maximization and loss minimization models and used multiobjective quantum behavior particle swarm algorithm with a single/dual potential trap to solve the model. Literature [6] uses a genetic algorithm based on reference point nondominated sorting to solve the optimization problem of multispace-based ground strike weapon multitarget firepower resource allocation. Literature [7] uses the multitarget discrete particle swarm-gravity search algorithm (MODPSO-GSA) to achieve the solution of the multitarget allocation model of coordinated air combat weapons. The decomposition-based multiobjective evolutionary algorithm decomposes the high-dimensional and

complex multiobjective optimization problem into multiple single-objective subproblems by referring to the decomposition strategy in mathematical programming and optimizes the subproblems separately. It has the advantages of high algorithm efficiency and simple operation [8, 9].

The MOEA/D algorithm has been used in the study of multitarget firepower resource allocation in edge environment. Literature [10] comprehensively considers the influence of factors such as weapon type, target number, and damage probability and uses the MOEA/D algorithm as the framework to construct the WMOM/D algorithm for solving the multitarget fire distribution model. Simulation experiments prove that the WMOM/D algorithm has the advantage of solving the problem of small-scale fire distribution. Literature [11] applies the MOEA/D algorithm to the multiobjective fire optimization problem of aircraft carrier formation antisubmarine warfare and proposes the GD-MOEA/D algorithm combining differential evolution and Gaussian mutation operation, which greatly improves the speed of solving the problem and the quality of the solution. Literature [12] integrates the MOEA/D algorithm with the multilevel coevolutionary algorithm and uses the multilevel cooperative MOEA/D algorithm to solve the multiobjective optimization model of the joint fire strike target assignment problem. The simulation experiment proves that the algorithm has good convergence and uniformity.

However, because the MOEA/D algorithm uses a fixed neighborhood update mechanism, the ability to reasonably allocate computing resources is low especially in the edge environment with limited computing resources. So the problems such as reduced population evolution efficiency and poor evolution quality will occur in the calculation process.

To this end, this paper considers the impact of subproblems and the degree of population evolution on the performance of the algorithm, designs the decision mechanism for subproblems and population evolution stages, and proposes a MOEA/D algorithm based on the neighborhood adaptive adjustment mechanism. Compared with the traditional MOEA/D algorithm, the NAAM-MOEA/D algorithm can better balance the convergence and distribution and improve the quality of the solution.

In the simulation experiment, the NAAM-MOEA/D algorithm was compared with the MOEA/D algorithm, the MOEA/D-DE algorithm, and the NSGA-III algorithm. The algorithm running time was reduced by 82.1%, 108.1%, and 153.6%, respectively; the GD value was reduced by 84%, 59%, and 35%, respectively; and the IGD value of the algorithm was reduced by 75%, 56%, and 40%, respectively.

The main innovations of this article are summarized as follows:

- (1) Aiming at the defects of the traditional MOEA/D algorithm's fixed neighborhood update mechanism in solving the multiobjective fire resource allocation problem, a MOEA/D algorithm based on the neighborhood adaptive adjustment mechanism is proposed, which greatly improves the efficiency and quality in edge

- (2) A method for judging the evolution stage of the population based on the attribution of the weight vector and the degree of evolution of the subproblems is proposed, which provides a reliable basis for judging the evolution state of the population
- (3) Based on the population evolution stage judgment method, a neighborhood adaptive adjustment mechanism is constructed and used in the MOEA/D algorithm to improve the convergence and distribution of the algorithm

The organizational structure of the paper is as follows:

Firstly, the related work is discussed in Section 2. Then, the optimization model of firepower resource allocation in edge environment is established in Section 3.1, the construction and decomposition of subproblems are discussed in Section 3.2.1, the shortcomings of traditional MOEA/D algorithm are analyzed in Section 3.2.2, and a mechanism for judging population evolution state is proposed in Section 3.2.3. The neighborhood adaptive adjustment mechanism is proposed in Section 3.2.4 and the steps of the NAAM-MOEA/D algorithm are summarized in Section 3.2.5. Finally, the simulation experiment is carried out in Section 4, and the performance of the algorithm is tested.

2. Related Work

In order to improve the performance of traditional MOEA/D algorithms in edge environment, researchers have proposed a variety of improved algorithms. The MOEA/D-DE algorithm proposed in literature [13] uses a difference operator instead of an evolution operator to enrich the diversity of the population, but the difference operator used by the algorithm is only applicable to a population of a specific size. The MOEA/D-DRA algorithm proposed in literature [14] allocates corresponding computing resources according to the complexity of specific problems and improves the performance of the algorithm by dynamically adjusting resource allocation; however, the proposed resource allocation criteria also have certain limitations. The MOEA/D-GL algorithm proposed in literature [15] embeds the grouping and statistical learning mechanism in the traditional MOEA/D algorithm, which prevents the population from falling into local optimization and improves the diversity of the population, but the overall performance improvement of the algorithm is not significant. The CD-MOEA/D-DE algorithm proposed in literature [16] controls the operation process of the algorithm by formulating control parameters ∂ and balances the performance of a multiobjective optimization problem solving and adaptive ability; however, the algorithm has a certain randomness in the value of the control parameter ∂ and does not have universal applicability.

In addition, the researchers have proposed many specific improvement measures for the shortcomings of the fixed neighborhood update mechanism of the MOEA/D algorithm in solving multiobjective optimization problems especially in the edge conditions with limited computing resources; however, the article does not elaborate on the

mechanism of how the neighborhood size affects the performance of the MOEA/D algorithm. Literature [17] points out that the size of the neighborhood will have an important impact on the performance of the MOEA/D algorithm, which provides important research directions for subsequent researchers. Literature [18] believes that different multiobjective optimization problems require different neighborhood sizes, and that the same multiobjective optimization problem also requires different neighborhood sizes at different stages of the algorithm, and proposes the ENS-MOEA/D algorithm with neighborhood adaptive adjustment capability; however, the ENS-MOEA/D algorithm may fall into local optimization in the later stage of operation. The ADEMO/D-ENS algorithm proposed in literature [19] combines the adaptive differential evolution algorithm with the variable neighborhood decomposition method to achieve the optimization of the algorithm. The MOEA/D-AGR algorithm proposed in literature [20] introduces an adaptive global replacement strategy in the neighborhood update method, which makes up for the shortcomings of the traditional MOEA/D algorithm in terms of global search capabilities. The MOEA/D-NMO algorithm proposed in literature [21] combines mutation strategies with different characteristics and neighborhoods of different sizes to select the best evolutionary combination to ensure the convergence of the algorithm while maintaining the diversity of the algorithm. The algorithms proposed in literature [19], literature [20], and literature [21] have all made improvements to the fixed field, but they all have certain limitations in application.

Although the current improved methods for fixed neighborhoods have improved the performance of traditional MOEA/D algorithms, the neighborhood adaptive strategies used by these algorithms do not consider the impact of population evolution on neighborhoods. Literature [22] proposes a neighborhood adaptive adjustment mechanism based on population evolution stage and individual fitness value, so that every individual has a corresponding neighborhood value at different evolution stages, but its neighborhood adjustment method does not consider the evolution status of the subproblems. Although the MOEA/D-ANS algorithm proposed in literature [23] adopts the ANS mechanism that adaptively adjusts the size of the neighborhood according to the evolution state of the population and subproblems, it can balance the convergence and distribution of population evolution, but it does not give a clear method on the statistical evolution of the number of better subquestions.

3. Method

3.1. Optimization of Fire Resource Allocation Model. The multitarget firepower resource allocation optimization problem in the edge environment can be described as follows: on the basis of satisfying the maximum damage effect and the minimum combat cost, determine the number of various weapons and equipment used to strike specific targets to obtain a feasible combat plan.

Set the target set of the enemy's combat system as $D = \{D_1, D_2, \dots, D_M\}$; D_i represents the i -th target. There are a total of N types of weapons available for use.

$B = \{B_1, B_2, \dots, B_N\}$, and B_j represents the j -th types of weapons. If there is a total of M_j class to choose from the j -th type of weapons, then $B_j = \{B_j^1, B_j^2, B_j^3, \dots, B_j^{M_j}\}$. Select the j -th weapon in the weapon set B to strike the i -th target in the target set D ; the probability of the target being destroyed is p_{ij} and the cost of each use of the j -th weapon is C_j . Suppose that the damage ability of the j -th type of weapons to target D_i is

$$P_i = 1 - \prod_{n=1}^{M_j} (1 - m_{ij}^n p_{ij}). \quad (1)$$

Among them, only when the j -th type of weapons of class n weapon is used to strike target D_i , there is $m_{ij}^n = 1$; otherwise, $m_{ij}^n = 0$. The purpose of firepower resource allocation is to maximize the damage effect under limited conditions. It is necessary to consider the priority of attacking the targets with high importance. Therefore, the calculation model of damage capability can be defined as

$$\max f(x) = \sum_{i=1}^M \sum_{j=1}^N \omega_i \left[1 - \prod_{n=1}^{M_j} (1 - m_{ij}^n p_{ij}) \right]. \quad (2)$$

Among them, ω_i is the importance of the i -th target.

In addition, the minimum operational cost calculation model is defined as follows:

$$\min C(x) = \sum_{i=1}^M \sum_{j=1}^N \sum_{n=1}^{M_j} m_{ij}^n C_j. \quad (3)$$

The constraints of the model are as follows:

- (1) *Damage lower bound constraint:* if the target is to be destroyed to a certain extent so that it will lose certain combat capability, it is necessary to reach its damage lower bound. If the damage lower bound of target i is defined as β_i , then

$$P_i = 1 - \prod_{n=1}^{M_j} (1 - m_{ij}^n p_{ij}) \geq \beta_i. \quad (4)$$

- (2) *Constraints on the number and types of weapons used:* it is stipulated that one weapon can only attack one target at most:

$$\sum_{i=1}^M m_{ij}^n \leq 1, \quad j = 1, 2, 3, \dots, N, n = 1, 2, 3, \dots, M_j. \quad (5)$$

It is stipulated that one type of weapon can only attack one type of target:

$$\sum_{j=1}^N m_{ij}^n \leq 1, \quad i = 1, 2, 3, \dots, M, n = 1, 2, 3, \dots, M_j. \quad (6)$$

In summary, the multiobjective firepower resource allocation optimization model can be defined as

$$\left\{ \begin{array}{l} \max f(x) = \sum_{i=1}^M \sum_{j=1}^N \omega_i \left[1 - \prod_{j=1}^{M_j} (1 - m_{ij}^n p_{ij}) \right], \\ \min C(x) = \sum_{i=1}^M \sum_{j=1}^N \sum_{n=1}^{M_j} m_{ij}^n C_j, \\ \text{s.t.} \\ \sum_{i=1}^M m_{ij}^n \leq 1, \quad j = 1, 2, 3, \dots, N, n = 1, 2, 3, \dots, M_j, \\ \sum_{j=1}^N m_{ij}^n \leq 1, \quad i = 1, 2, 3, \dots, M, n = 1, 2, 3, \dots, M_j, \\ P_i = 1 - \prod_{n=1}^{M_j} (1 - m_{ij}^n p_{ij}) \geq \beta_j. \end{array} \right. \quad (7)$$

3.2. Detailed Introduction of NAAM-MOEA/D Algorithm

3.2.1. Construction and Decomposition of Subproblem. The core of constructing the subproblem of the MOEA/D algorithm is to construct the weight vector of an objective function subproblem. Suppose the weight vector of the subproblem of the objective function is

$$\varphi^r = \left(\frac{r-1}{N-1}, \frac{N-r}{N-1} \right). \quad (8)$$

In the formula, N is the number of subproblems after decomposition, $r = 1, 2, \dots, N$.

The core of the MOEA/D algorithm is the decomposition operation, usually using aggregate functions to decompose the multiobjective constraint problem into single-objective subproblems. Commonly used decomposition methods are weighted sum method, Chebyshev method, and boundary crossing method based on penalty. This paper adopts the Chebyshev method, and its decomposition principle is

$$g^{te}(x|\varphi^r, Z^*) = \max_{1 \leq i \leq m} \{ \varphi_i^r | f_i(x) - Z_i^* \}. \quad (9)$$

Among them, $\varphi^* = \{ \varphi_1^*, \varphi_2^*, \dots, \varphi_m^* \}$ is the weight vector corresponding to the subproblem r . $Z^* = \{ Z_1^*, Z_2^*, \dots, Z_m^* \}$ is the ideal point. $f_i(x)$ is the i -th objective function, and φ_i^r is the i -th component of the weight vector φ^r . Z_i^* is the i -th component of the ideal point Z^* .

The single-objective optimization function of the i -th subproblem of objective function constructed by the Cheby-

shev method can be expressed as follows:

$$g^{te}(\mu|\gamma^r, Z) = \max_{1 \leq i \leq m} \left\{ \gamma_i^r \left| \frac{f_i(x) - Z_i}{c_i} \right| \right\}. \quad (10)$$

In the formula, $f_i(x)$ is the i -th objective function, Z is the reference vector, Z_i is the i -th component of the reference vector Z , γ^r is the weight vector, and γ_i^r is the i -th component of the weight vector γ^r .

3.2.2. Defects of Traditional MOEA/D Algorithm. The MOEA/D algorithm maintains the power of population evolution from the update strategy of the neighborhood. The parent gene of an individual comes from the neighborhood, and it adopts a coevolution model based on neighborhood update. The evolution of an individual is carried out on the basis of the neighborhood. While evolving by itself, it drives the evolution of other neighborhoods by optimizing other individuals in the neighborhood. The MOEA/D algorithm uses a fixed neighborhood strategy. For different subproblems, the MOEA/D algorithm divides it into a neighborhood of the same size. In fact, the computational complexity of each subproblem in the objective function is different. The subproblems have different requirements for the size of the neighborhood at different stages. The size of the neighborhood has a very important impact on the evolution of the subproblems. When the size of the neighborhood is large, the probability of other individuals in the neighborhood being replaced by offspring individuals increases, and the population convergence speeds up, but the distribution of the population will become worse as the neighborhood size increases, making it easy for the algorithm to fall into local find the best. When the size of the neighborhood is small, the probability of other individuals in the neighborhood being replaced by offspring individuals decreases, the population convergence speed slows, the algorithm convergence decreases, and the overall evolution speed of the population decreases accordingly.

3.2.3. Judging Mechanism of Population Evolution State. From the previous analysis, we can see that in the MOEA/D algorithm, subproblems and populations have different requirements for neighborhood size at different evolution stages. Then, how to judge the evolution state of the population and whether it can find a mechanism that can effectively evaluate the evolution stage of the population is the core problem that the new algorithm needs to solve.

Some scholars propose to use the individual density of subproblems to assess the degree of population evolution. The individual density of the subproblem is equivalent to the number of individuals in the subinterval. If the individual density of the subproblem is smaller, the surrounding individuals are denser, the better the degree of evolution of the individual is, and the greater the probability of the problem being solved. If the individual density of the subproblem is smaller, the surrounding individuals are sparser, then the degree of evolution of the individual is smaller, and the problem is less likely to be solved.

Input: the threshold $d_m, \omega_m, \varepsilon_m$; JM_{wei}^i is the attribution judging mechanism of weight vector; JM_{sub}^i is the subproblem evolution degree judgment mechanism; JM_{pop}^i is the population evolution degree judgment mechanism; d_{wi} is the distance between the weight vector and the individual; ω is the number of individuals owned by the weight vector; ε represents the number of subproblems with better evolution.

Output: the population evolution state

1 Determine the attribution of the weight vector;

2 **for** $d_{wi} \leq d_m$, **do**

$JM_{wei}^i = 1$; determine that the individual belongs to the weight vector

else do $JM_{wei}^i = 0$

3 Calculate the number of individuals owned by the weight vector: $\omega = \sum_{i=1}^K JM_{wei}^i$;

4 Determine the degree of evolution of subproblems;

5 **for** $\omega \geq \omega_m$, **do**

$JM_{sub}^i = 1$ and determine the degree of evolution of subproblems is better

else do $JM_{sub}^i = 0$;

6 Calculate the number of subproblems with better evolution: $\varepsilon = \sum_{i=1}^K JM_{sub}^i$

7 Determine the evolution stage of the population;

8 **for** $\varepsilon > \varepsilon_m$, **do**

$JM_{pop}^i = 2$ and determine that the current population evolution degree is too fast, belonging to an overevolution state;

else for $\varepsilon < \varepsilon_m$, **do**

$JM_{pop}^i = 0$ and determine that the current population is slowly evolving and belongs to a state of lagging evolution

else for $\varepsilon = \varepsilon_m$, **do**

$JM_{pop}^i = 1$ and determine that the current population has a good evolution speed and belongs to a normal evolutionary state

9 **end**

ALGORITHM 1: Judgment mechanism of population evolution status.

Input: the initial population $X = \{x_1, x_2, \dots, x_n\}$; the size of the initial neighborhood corresponding to the subproblem within the population $T_s = \{T_{s1}, T_{s2}, \dots, T_{sn}\}$; the population initial neighborhood T_i

Output: the size of the neighborhood corresponding to the subproblem within the population $T_s = \{T_{s1}, T_{s2}, \dots, T_{sn}\}$, the current population size T^* .

1 Initialize: $T^* = T_i = T_s$.

2 evolution

3 **for** $x_i \in X$ **do**

Take x_i corresponding to the individuals in neighborhood $B(i) = \{i_1, i_2, \dots, i_T\}$ to perform crossover and mutation operations to obtain offspring individuals;

4 Determine the degree of evolution of the subproblems and the evolution status of the population according to the mechanism provided in Section 3.2.3;

5 Neighborhood adaptive adjustment

6 **for** $JM_{sub}^i = 1$, **do**

The evolution of the previous generation is fast and uses formula (11) to appropriately reduce the current neighborhood size T ;

else for $JM_{sub}^i = 0$, **do**

The evolution of the previous generation is slow and uses formula (11) to appropriately increase the current neighborhood size T ;

7 **for** $JM_{pop}^i = 1$, **do**

The evolution rate of the previous generation population is moderate;

else for $JM_{pop}^i = 2$, **do**

The evolution rate of the previous generation population is fast and uses formula (12) to appropriately reduce the current population size T^* ;

else for $JM_{pop}^i = 0$, **do**

The evolution rate of the previous generation population is slow and uses formula (12) to appropriately increase the current population size T^* ;

8 Output $T_s = \{T_{s1}, T_{s2}, \dots, T_{sn}\}$ and T^* .

ALGORITHM 2: Neighborhood adaptive adjustment mechanism.

Input: optimal model of multiobjective fire resource allocation; termination criteria; population size N ; population crossover probability p_c ; probability of population variation p_m ;
Output: optimal plan for firepower resource allocation
1 Initialize
2 EP = \emptyset
3 Initialize population individuals x_1, x_2, \dots, x_N ;
4 Generate weight vector $\varphi_1, \varphi_2, \dots, \varphi_N$;
5 Calculate the Euclidean distance between any two weight vectors. For each weight vector, find the T nearest weight vectors to form its neighborhood. $i = 1, 2, \dots, N$ and $B(i) = \{i_1, i_2, \dots, i_T\}$. Among them, $\varphi_{i_1}, \varphi_{i_2}, \dots, \varphi_{i_T}$ are the T weight vectors closest to φ_i ;
6 Initialize the ideal point $Z^* = (Z_1, Z_2, \dots, Z_m)^T$;
7 Evolution
8 for $i = 1, 2, \dots, N$, do
9 Crossover and mutation: randomly select two individuals in $B(i)$ to perform crossover and mutation operations to obtain offspring individual y ;
10 Update ideal point Z^*
11 for each $j = 1, 2, \dots, m$.
if $Z_j < f_j(y)$, then $Z_j = f_j(y)$
end
12 Set adaptive neighborhood
13 Judge the subproblems and the evolution status of the population through the criteria provided in Section 2.1;
14 Use the method provided in Section 2.2 to obtain the population neighborhood T^* and the neighborhood T corresponding to the subproblem;
15 Update neighborhood $B(i)$
for each $j \in B(i)$
if $g^{ws}(y' \lambda_j, z) \leq g^{ws}(x^j \lambda_j, z)$ then
$x^j = y', FV^j = F(y')$
end
end
16 Remove all individuals dominated by $F(y')$ in EP and add individuals not dominated to EP at the same time;
17 end
18 Stop operation
19 After the algorithm evolves to the maximum algebra G_{\max} , it stops and outputs the optimal solution. If the stopping condition is not met, it returns to Step 7.

ALGORITHM 3: The framework of NAAM-MOEA/D algorithm.

TABLE 1: Optimization model parameters of firepower allocation resources.

Project	W1	W2	W3	W4	Importance
T1	0.82	—	—	—	0.22
T2	—	0.95	—	—	0.31
T3	—	—	0.87	—	0.28
T4	—	—	—	0.85	0.19
Unit cost	5	10	8	4	—
Total number	10	10	10	10	—

Some scholars propose that if the distance between the subproblem and a certain solution in space is used as the evaluation criterion, if the distance between them is relatively close, it can be judged that the solution belongs to the subproblem. In the spatial coordinate system, the solution corresponds to the individual in the coordinate system, and the subproblem corresponds to the weight vector. Therefore, the problem of determining the attribution of the solution can be transformed into the problem of finding the distance between the weight vector and the individual.

This paper proposes a mechanism for evaluating the evolutionary stage of a population (see Algorithm 1):

3.2.4. Neighborhood Adaptive Adjustment Mechanism. In order to meet the needs of balancing the convergence and distribution of the MOEA/D algorithm, according to the population evolution state judgment mechanism in Section 3.2.3, this paper proposes a neighborhood strategy that adaptively adjusts the population size based on the different evolution stages of the population, which can also be called a neighborhood adaptive adjustment mechanism (NAAM) as in Algorithm 2.

The setting adjustment formula is as follows:

$$T_s = \begin{cases} T^* \left[1 - \frac{1}{T^*} \omega \left(\frac{mT^*}{N} \right)^\theta \right], & JM_{\text{sub}}^i = 1, \\ T^* \left[1 + \frac{1}{T^*} \omega \left(\frac{mT^*}{N} \right)^\theta \right], & JM_{\text{sub}}^i = 0, \end{cases} \quad (11)$$

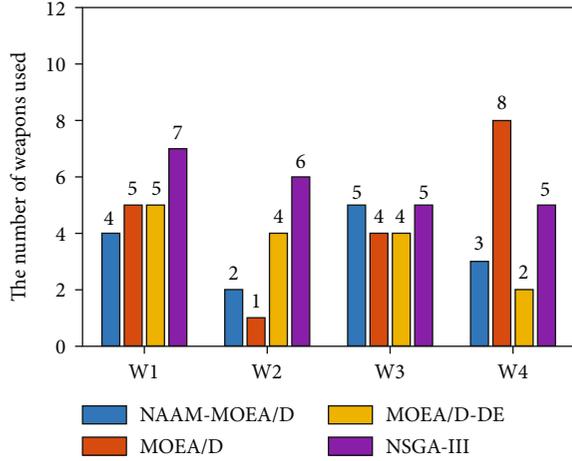


FIGURE 1: Number of weapons of each type used by the 4 algorithms.

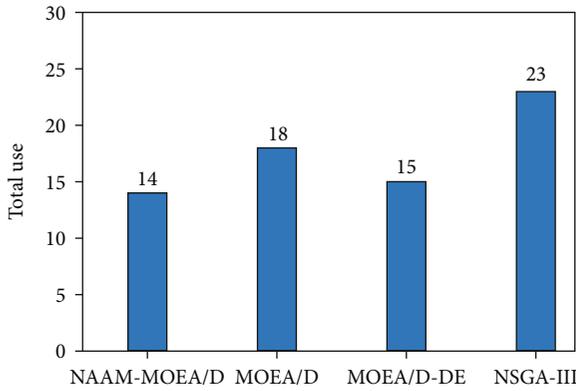


FIGURE 2: Total number of weapons used by the 4 algorithms.

$$T^* = \begin{cases} T_i \left[1 - \frac{1}{T_i} \omega \left(\frac{mT_i}{N} \right)^\theta \right], & JM_{\text{pop}}^i = 2, \\ T_i, & JM_{\text{pop}}^i = 1, \\ T_i \left[1 + \frac{1}{T_i} \omega \left(\frac{mT_i}{N} \right)^\theta \right], & JM_{\text{pop}}^i = 0. \end{cases} \quad (12)$$

3.2.5. *The Framework of NAAM-MOEA/D Algorithm.* The framework of the NAAM-MOEA/D algorithm can be described as in Algorithm 3:

4. Experiment and Simulation

4.1. *Example Analysis of Algorithm.* There are 4 types of weapons to strike at 4 targets in the enemy's combat system. Combining the content of Section 3.1, we assume that the model satisfies various constraints, and the model parameters are given in Table 1.

MATLAB 2020 is selected to write the algorithm program. The running environment is a Windows 7 R64-bit operating system, 4 GB memory, Intel Pentium processor. The NAAM-MOEA/D algorithm, MOEA/D algorithm,

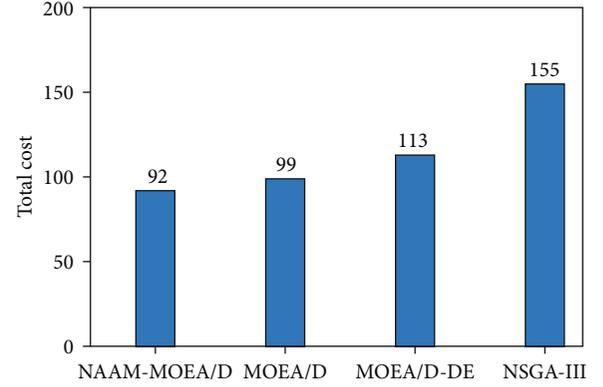


FIGURE 3: Total computational cost of the 4 algorithms.

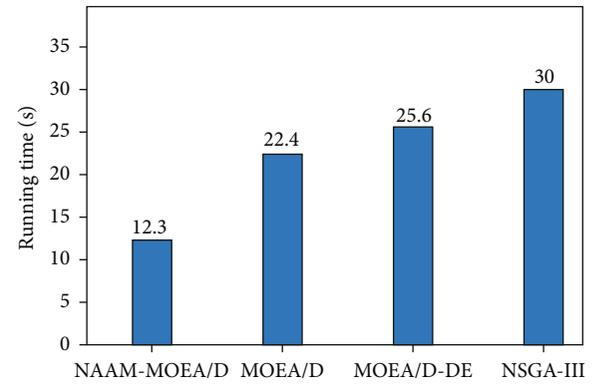


FIGURE 4: Calculation time spent of the 4 algorithms.

TABLE 2: Statistics of fire resource distribution.

Project	W1 T1	W2 T2	W3 T3	W4 T4	Total use	Total cost	Running time
NAAM-MOEA/D	4	2	5	3	14	92	12.3 s
MOEA/D	5	1	4	8	18	99	22.4 s
MOEA/D-DE	5	4	4	2	15	113	25.6 s
NSGA-III	7	6	5	5	23	155	30 s

MOEA/D-DE algorithm, and NSGA-III algorithm are selected for the simulation operation.

Figure 1 counts the number of various weapons used by the four algorithms. It can be seen from Figure 1 that the NSGA-III algorithm uses 7 W1 weapons, which is more than the MOEA/D algorithm and the MOEA/D-DE algorithm; both algorithms use 5 W1 weapons, and the NAAM-MOEA/D algorithm uses 4 W1 weapons. The NSGA-III algorithm uses 6 W2 weapons; the MOEA/D-DE algorithm and the NAAM-MOEA/D algorithm use 4 and 2 W2 weapons, respectively; while the MOEA/D algorithm uses the least number of W2 weapons and only one is used. The NAAM-MOEA/D algorithm and the NSGA-III algorithm both use 5 W3 weapons, which is more than the MOEA/D algorithm and the MOEA/D-DE algorithm. Both algorithms use 4 W3 weapons. The MOEA/D algorithm uses 8 W4

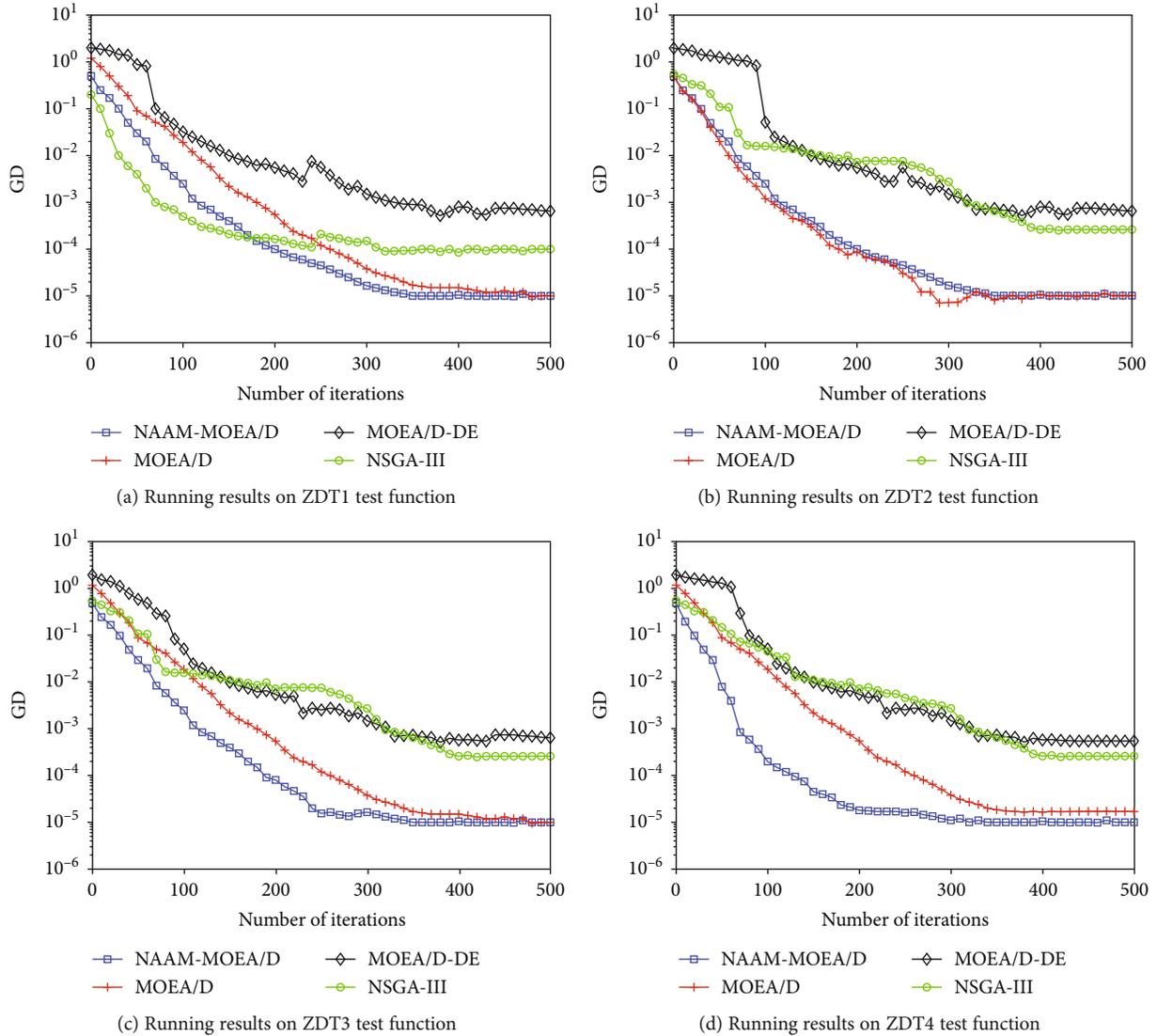


FIGURE 5: Variation curve of GD with algorithm iteration number.

weapons, which is more than the NSGA-III algorithm. The NAAM-MOEA/D algorithm and the MOEA/D algorithm use 3 and 2 W4 weapons, respectively.

Figure 2 counts the total number of weapons used by the four algorithms. It can be seen from Figure 2 that the NSGA-III algorithm uses the largest number of weapons, using 23 weapons in total. The MOEA/D algorithm and the MOEA/D-DE algorithm use 18 and 15 weapons, respectively, and the NAAM-MOEA/D algorithm uses the least amount of weapons—only 14 weapons are used.

Figure 3 compares the total cost of weapon use of the four algorithms. It can be seen from Figure 3 that the NSGA-III algorithm costs the most weapons, with a total cost of 155, followed by the MOEA/D-DE algorithm, with a total cost of 113, while the MOEA/D algorithm and NAAM-MOEA/D algorithm had the least weapon use cost, costing 99 and 92, respectively.

Figure 4 compares the computing time of the four algorithms. It can be seen from Figure 4 that the NAAM-MOEA/D algorithm has the least computing time, which takes only 12.3 s, and the NSGA-III algorithm has the most

computing time, which takes 30 s. The computing times of the MOEA/D algorithm and the MOEA/D-DE algorithm are, respectively, 22.4 s and 25.6 s.

The statistics of firepower resource allocation obtained through simulation calculation are shown in Table 2.

It can be seen from Table 2 that the number of weapons used and the total cost obtained by the NAAM-MOEA/D algorithm are better than those of the other three algorithms. The number of weapons used by the MOEA/D-DE algorithm is close to the number of weapons used by the NAAM-MOEA/D algorithm, but the total cost is about 23% higher. The total cost calculated by the MOEA/D-DE algorithm is close to the total cost calculated by the NAAM-MOEA/D algorithm, but 4 more weapons are used. The number of weapons used and the total cost obtained by the NSGA-III algorithm are significantly more than those of the other three algorithms, indicating that the algorithm has the worst performance. In addition, the running time of the NAAM-MOEA/D algorithm is 12.3 s, which is reduced by 82.1%, 108.1%, and 153.6% compared with the MOEA/D algorithm,

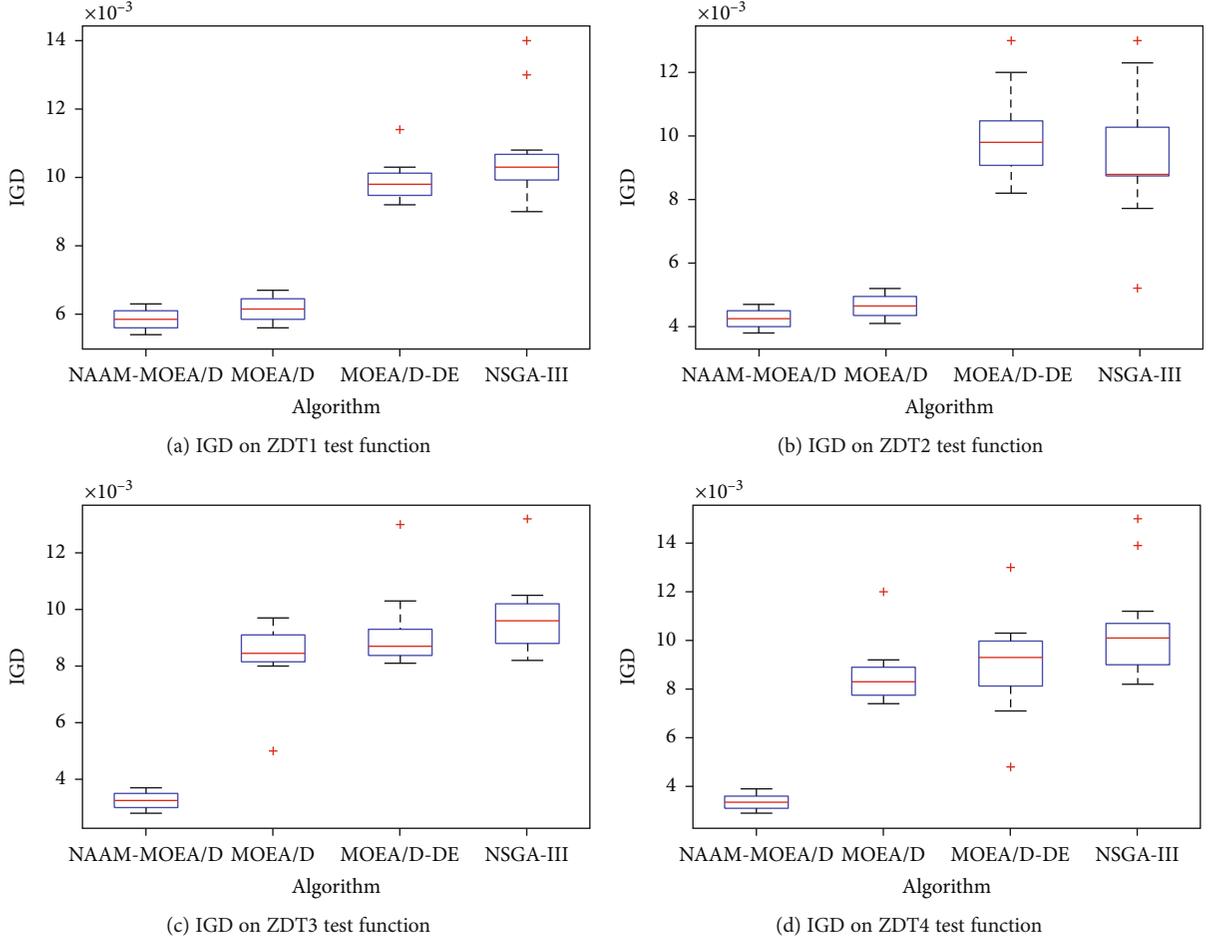


FIGURE 6: IGD box plot of the algorithm under different test functions.

TABLE 3: Comparison of IGD indicators of algorithms on ZDT series functions.

Test function	NAAM-MOEA/D Mean (std)	MOEA/D Mean (std)	MOEA/D-DE Mean (std)	NSGA-III Mean (std)
ZDT1	5.83E-03 (4.73E-04)	5.96E-03 (5.13E-04)	9.57E-03 (1.31E-02)	1.01E-02 (1.65E-02)
ZDT2	4.17E-03 (4.85E-04)	4.51E-03 (8.13E-04)	9.33E-03 (1.07E-04)	8.75E-03 (2.27E-03)
ZDT3	3.17E-03 (2.05E-04)	8.48E-03 (9.30E-04)	8.73E-03 (5.95E-04)	9.55E-03 (2.68E-04)
ZDT4	3.25E-03 (2.72E-04)	8.21E-03 (4.03E-04)	9.12E-03 (7.15E-04)	1.04E-02 (4.52E-03)

the MOEA/D-DE algorithm, and the NSGA-III algorithm, respectively, indicating that the NAAM-MOEA/D algorithm has obvious advantages in computing speed.

4.2. Performance Test of the Algorithm. In order to verify the performance of the NAAM-MOEA/D algorithm, ZDT series of test functions are selected to test the performance of the NAAM-MOEA/D algorithm with the MOEA/D algorithm, MOEA/D-DE algorithm, and NSGA-III algorithm.

In order to ensure the fairness and rationality of the algorithm evaluation, the population size and initial neighborhood size of the four algorithms are set to the same (population size $N = 100$, initial neighborhood $T = 100$). All algorithms adopt simulated binary crossover (crossover probability $p_c = 0.9$) and polynomial mutation (mutation

probability $p_m = 1/n$, n is the dimension of decision variables). Each algorithm runs 20 times independently, and the evaluation times are set to 10000. Inverse generation distance (IGD) and generation distance (GD) were used as evaluation indexes. Each test function is run 20 times independently and averaged every 10 generations. The variation curve of GD with the number of iterations (0-500 generations) of the algorithm is shown in Figure 1.

As shown in Figure 5(a), the NAAM-MOEA/D algorithm tends to be stable on the test function ZDT1, and the convergence speed is slower than the NSGA-III algorithm and faster than the MOEA/D algorithm and the MOEA/D-DE algorithm.

As shown in Figure 5(b), on the test function ZDT2, the convergence speed of the NAAM-MOEA/D algorithm is

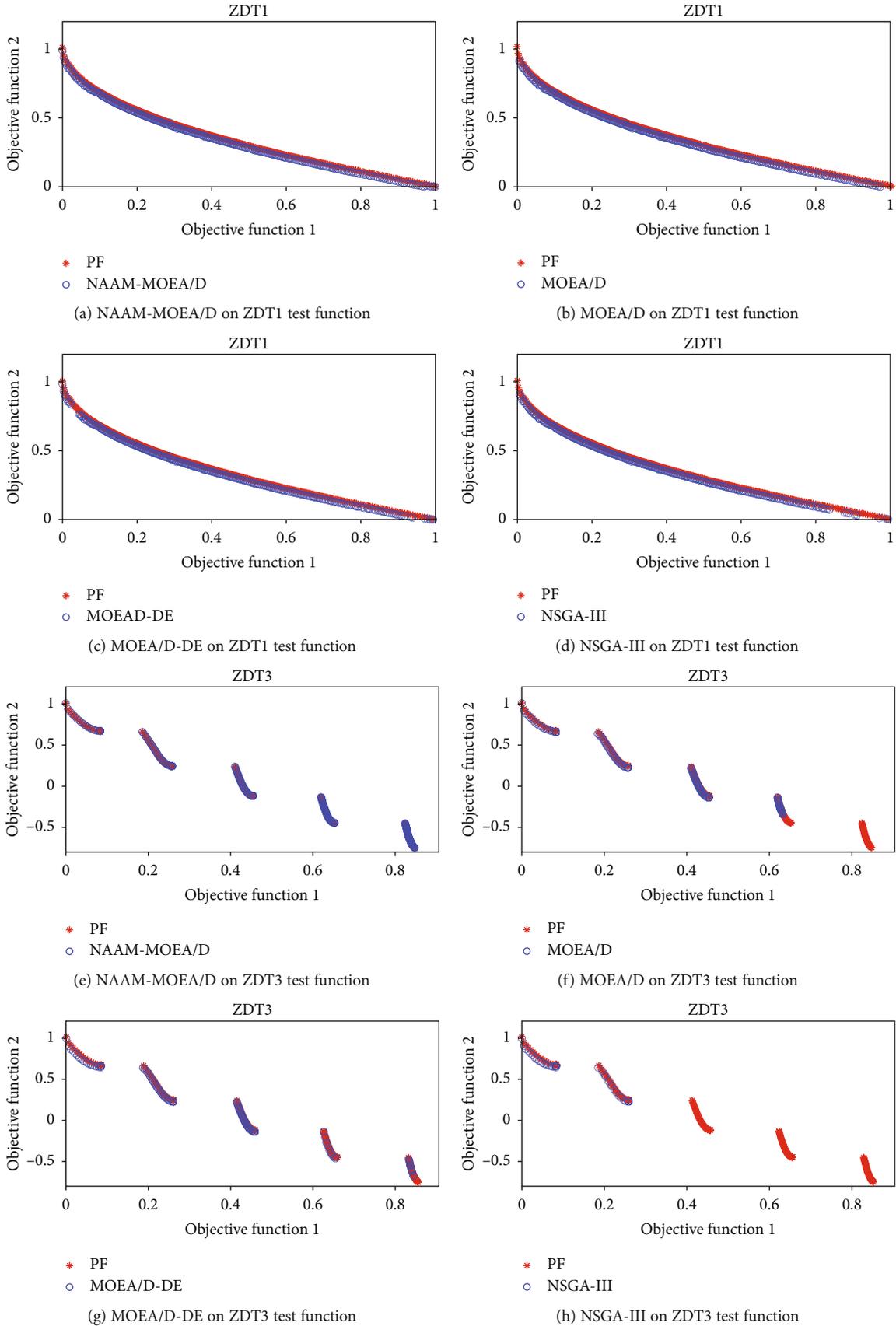


FIGURE 7: Continued.

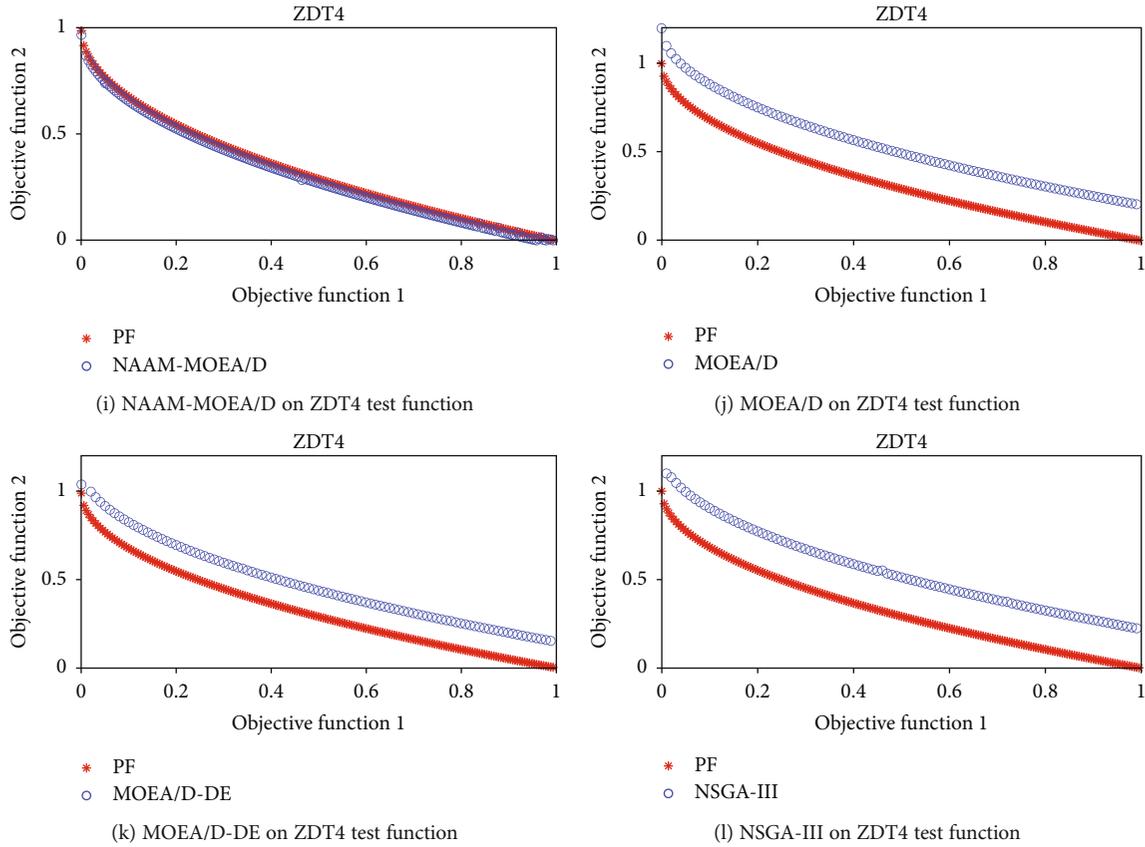


FIGURE 7: Comparison of Pareto front and ideal Pareto front on ZDT test function.

faster than that of the MOEA/D-DE algorithm and the NSGA-III algorithm. Although it is slightly slower than the MOEA/D algorithm, the population degradation degree of the MOEA/D algorithm is higher than that of the NAAM-MOEA/D algorithm.

As shown in Figure 5(c), on the test function ZDT3, the NAAM-MOEA/D algorithm converges faster than the other algorithms.

As shown in Figure 5(d), on the test function ZDT4, the NAAM-MOEA/D algorithm has a faster population convergence speed due to the advantages of the adaptive neighborhood adjustment mechanism adopted, and the algorithm convergence performance is significantly better than the MOEA/D algorithm, MOEA/D-DE algorithm, and NSGA.

Therefore, the NAAM-MOEA/D algorithm not only ensures that the algorithm has a faster convergence rate but also solves the population degradation problem that occurs during the algorithm operation and ensures the stability of the algorithm operation, so that the algorithm can have more resources to improve the diversity of the population.

As shown in Figure 6, comparing the IGD box plots of various algorithms on the ZDT series test functions in the comparison Table 3, we can see that the NAAM-MOEA/D algorithm's mean, minimum, median (at the position of the red line in the figure), and interquartile range (key indicators such as box length) are lower than those of the MOEA/D algorithm, MOEA/D-DE algorithm, and NSGA-III algorithm. The probability and size of the abnormal value of the

NAAM-MOEA/D algorithm are also lower than those of the other three algorithms, which show that the stability and quality of the NAAM-MOEA/D algorithm is higher.

On the test functions ZDT1 and ZDT2, the comprehensive performance of the NAAM-MOEA/D algorithm is slightly better than that of the MOEA/D algorithm and significantly better than that of the MOEA/D-DE algorithm and the NSGA-III algorithm. On the test functions ZDT3 and ZDT4, the comprehensive performance of the NAAM-MOEA/D algorithm is significantly better than that of the MOEA/D algorithm, the MOEA/D-DE algorithm, and the NSGA-III algorithm. This is because there are many discontinuous regions in the target space of test function ZDT3. These regions adopt the fixed neighborhood setting method, but do not use the adaptive neighborhood allocation strategy to reasonably allocate the algorithm, which leads to the waste of algorithm resources and the slowdown of population evolution speed.

Figure 7 shows the comparison of the Pareto front and the ideal Pareto front obtained by the four algorithms on the ZDT test function. Among them, the red meter character represents the ideal PF, and the blue circle represents the optimal solution of the Pareto frontier obtained by the various algorithms.

On the test function ZDT3, the improved MOEA/D solution set is more evenly distributed on the ideal Pareto front. In the other three algorithms, some leading edges are not completely found, and the solution set is missing to a certain extent. Among them, the MOEA/D algorithm and the MOEA/D-DE algorithm have a little poor distribution of

solution set, while the NSGA-III algorithm has the least distribution. This is because the other algorithms spend limited computing resources in the discrete region of test function ZDT3 and produce too many nondominated solutions, which hinders the evolution of the population.

On the test function ZDT4, the NAAM-MOEA/D algorithm has converged to the ideal, while the other algorithms have fallen into the local optimization state to varying degrees. It can be seen that the NAAM-MOEA/D algorithm has more advantages in reasonable allocation of computing resources and can better ensure the convergence of the algorithm.

Through the comparison, we can see that the Pareto frontier solution set obtained by the NAAM-MOEA/D algorithm almost uniformly converges to the PF of the ideal Pareto. However, the other three algorithms have different degrees of missing or uneven distribution of solution sets in various test functions. The NAAM-MOEA/D algorithm shows some performance advantages when dealing with simple test problems such as ZDT1, but the advantages are not obvious. However, the NAAM-MOEA/D can allocate computing resources reasonably and take into account the convergence and distribution of the algorithm due to its flexible neighborhood update strategy when dealing with relatively complex test problems such as ZDT3 and ZDT4.

5. Discussion

In this section, we establish a firepower resource allocation optimization model for edge environment based on given specific data, conduct simulation experiments, and test and evaluate the performance of the algorithm combined with the ZDT series of functions. However, several additional points should be pointed out and further analyzed in detail, which are specified as below.

- (1) The types of weapons and the number of samples given in Section 4.1 are not large enough (both are 4). Therefore, in the future simulation experiments, we should focus on large sample data sets to verify the performance of the method under the condition of large sample data
- (2) In Section 4.2, the ZDT series functions are selected to test the performance of the algorithm. The simulation results show that the performance of the NAAM-MOEA/D algorithm is better than that of the other three algorithms. However, only one kind of test function verification is not convincing enough, so DLTZ, WFG, and other test functions should be selected to evaluate the algorithm, so as to provide more sufficient reference for the improvement of algorithm performance

6. Conclusion

This paper constructs a multiobjective firepower resource allocation optimization model for edge environment with limited computing resources, based on maximizing damage effect and minimizing combat cost. Aiming at the

defects of the traditional MOEA/D algorithm fixed neighborhood update mechanism, a MOEA/D algorithm based on neighborhood adaptive adjustment mechanism is proposed and the model is solved. It can be seen from the simulation experiment that the MOEA/D algorithm based on the neighborhood adaptive adjustment mechanism has significantly improved its stability, convergence, and distribution.

In the next step, current work will continue to be improved by considering security and privacy issues [24–33]. In addition, more complex multiobjective solutions with more context factors [34–41] will be considered.

Abbreviations

MOEA/D:	Multiobjective evolutionary algorithm based on decomposition
NAAM-MOEA/D:	Neighborhood adaptive adjustment mechanism-multiobjective evolutionary algorithm based on decomposition
MODPSO-GSA:	Multiobjective discrete particle swarm optimization-gravitational search algorithm
WMOM/D:	Weapon-target assignment multiobjective model based on decomposition
GD-MOEA/D:	Gauss mutation and differential evolution based on a multiobjective evolutionary algorithm based on decomposition
MOEA/D-DE:	Multiobjective evolutionary algorithm based on decomposition-differential evolution
MOEA/D-DRA:	Multiobjective evolutionary algorithm based on decomposition-dynamical resource allocation
ENS-MOEA/D:	Ensemble neighborhood size-multiobjective evolutionary algorithm based on decomposition
ADEMO/D-ENS:	Adaptive differential evolution for multiobjective problems-ensemble neighborhood size
MOEA/D-AGR:	Multiobjective evolutionary algorithm based on decomposition-adaptive global replacement
MOEA/D-NMO:	Multiobjective evolutionary algorithm based on decomposition-neighborhood mutation operator
MOEA/D-ANS:	Multiobjective evolutionary algorithm based on decomposition-adaptive neighborhood strategy
NSGA-III:	Nondominated sorted genetic algorithm-III.

Data Availability

The experiment dataset is generated randomly through simulation.

Conflicts of Interest

We declare that there is no conflict of interest regarding this submission.

Authors' Contributions

Liyuan Deng finished the English writing, review, and editing of the paper. Liyuan Deng, Ping Yang, and Weidong Liu finished the experiments. Lina Wang, Sifeng Wang, and Xiumei Zhang finished the algorithm design.

Acknowledgments

This work was supported by Xi'an Research Institute of High-Technology.

References

- [1] L. I. Ping and L. I. Changwen, "Modeling and algorithm of weapon target cooperative fire assignment," *Command Control & Simulation*, vol. 37, no. 2, pp. 36–40, 2015.
- [2] J. Zhang, Z. X. Wang, L. Chen, Z. B. Wu, and J. F. Lu, "Modeling and optimization on antiaircraft weapon-target assignment at multiple interception opportunity," *Acta Armamentarii*, vol. 35, no. 10, pp. 1644–1650, 2014.
- [3] D. Chao-yang, L. Yao, and W. Qing, "Improved genetic algorithm for solve firepower distribution," *Acta Armamentarii*, vol. 37, no. 1, pp. 97–102, 2016.
- [4] C. L. Fan, Q. H. Xing, and M. F. Zheng, "Weapon-target allocation optimization algorithm based on IDPSO," *Systems Engineering and Electronics*, vol. 37, no. 2, pp. 336–342, 2015.
- [5] X. Hao, X. Qinghua, and W. Wei, "WTA for air and missile defense based on fuzzy multi-objective programming," *Systems Engineering and Electronics*, vol. 40, no. 3, pp. 563–570, 2018.
- [6] L. Qingguo, L. Xinxue, W. Jian, L. Yaxiong, and C. Hao, "Optimization of fire distribution for multiple SGSW based on improved NSGA-III," *Systems Engineering and Electronics*, vol. 42, no. 9, pp. 1995–2002, 2020.
- [7] J. J. Gu, J. J. Zhao, J. Yan, and X. Chen, "Cooperative weapon-target assignment based on multi-objective discrete particle swarm optimization-gravitational search algorithm in air combat," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 41, no. 2, pp. 252–258, 2015.
- [8] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transaction on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [9] S. Zhao, P. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 442–446, 2012.
- [10] 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.
- [11] L. Chen and Y. Ma, "Anti-submarine firepower optimization of aircraft carrier formation based on GD-MOEA/D algorithm," *Computer Simulation*, vol. 35, no. 10, pp. 33–38, 2018.
- [12] C. Hui and Y. Ma, "Model of target assignment in joint fire strike operations," *Journal of Systems Simulation*, vol. 30, no. 8, pp. 2942–2949, 2018.
- [13] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.
- [14] Z. H. A. N. G. Qingfu, L. I. U. Wudong, and L. I. Hui, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *2009 IEEE Congress on Evolutionary Computation*, pp. 203–208, Washington D.C., USA, 2009.
- [15] L. Li, D. Liu, and X. Wang, *Multi-objective permutation flow shop scheduling problem based on improved MOEA/D algorithm*, Computer Integrated Manufacturing Systems, 2020.
- [16] X. Zhou, W. Xuewu, and X. Gu, "MOEA/D based on constrained approach and differential evolution," in *Proceedings of the 38th Chinese Control Conference*, pp. 2034–2039, Guangzhou Baiyun International Convention Center, China, 2019.
- [17] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Use of biased neighborhood structures in multiobjective memetic algorithms," *Soft Computing*, vol. 13, no. 8-9, pp. 795–810, 2009.
- [18] S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 442–446, 2012.
- [19] H. Xia, J. Zhuang, and D. Yu, "Combining crowding estimation in objective and decision space with multiple selection and search strategies for multi-objective evolutionary optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 3, pp. 378–393, 2013.
- [20] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Adaptive replacement strategies for MOEA/D," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 474–486, 2016.
- [21] L. Liu and L. Zheng, "MOEA/D algorithm based on combinatorial optimization of neighborhood and mutation operator," *Computer Engineering*, vol. 43, no. 3, pp. 232–240, 2017.
- [22] E. Li and R. Chen, "Improved MOEA/D algorithm based on adaptive mutation operator and neighborhood size," *Computer Engineering and Applications*, vol. 55, no. 9, pp. 49–55, 2019.
- [23] H. Geng, W. Han, Y. Ding, and S. Zhou, "Improved MOEA/D algorithm based on adaptive neighborhood strategy," *Computer Engineering*, vol. 45, no. 5, pp. 161–168, 2019.
- [24] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2016.
- [25] Z. Sun, Y. Wang, Z. Cai, T. Liu, X. Tong, and N. Jiang, "A two-stage privacy protection mechanism based on blockchain in mobile crowdsourcing," *International Journal of Intelligent Systems*, 2021.
- [26] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, "Blockchain empowered arbitrable data auditing scheme for network storage as a service," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 289–300, 2020.
- [27] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

- [28] L. Qi, C. Hu, X. Zhang et al., "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4159–4167, 2020.
- [29] T. Liu, Y. Wang, Y. Li, X. Tong, L. Qi, and N. Jiang, "Privacy protection based on stream cipher for spatiotemporal data in IoT," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7928–7940, 2020.
- [30] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, 2019.
- [31] Y. Xu, C. Zhang, G. Wang, Z. Qin, and Q. Zeng, "A blockchain-enabled deduplicatable data auditing mechanism for network storage services," *IEEE Transactions on Emerging Topics in Computing*, p. 1, 2020.
- [32] W. Zhong, X. Yin, X. Zhang et al., "Multi-dimensional quality-driven service recommendation with privacy-preservation in mobile edge environment," *Computer Communications*, vol. 157, pp. 116–123, 2020.
- [33] Q. Liu, Y. Tian, J. Wu, T. Peng, and G. Wang, "Enabling verifiable and dynamic ranked search over outsourced data," *IEEE Transactions on Services Computing*, p. 1, 2019.
- [34] L. Wang, X. Zhang, R. Wang, C. Yan, H. Kou, and L. Qi, "Diversified service recommendation with high accuracy and efficiency," *Knowledge-Based Systems*, vol. 204, article 106196, 2020.
- [35] J. Li, T. Cai, K. Deng, X. Wang, T. Sellis, and F. Xia, "Community-diversified influence maximization in social networks," *Information Systems*, vol. 92, article 101522, 2020.
- [36] H. Liu, H. Kou, C. Yan, and L. Qi, "Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph," *Complexity*, vol. 2020, Article ID 2085638, 15 pages, 2020.
- [37] S. Zhang, Q. Liu, and Y. Lin, "Anonymizing popularity in online social networks with full utility," *Future Generation Computer Systems*, vol. 72, no. 7, pp. 227–238, 2017.
- [38] Z. Chunjie, L. Ali, H. Aihua, Z. Zhiwang, Z. Zhenxing, and W. Fusheng, "Modeling methodology for early warning of chronic heart failure based on real medical big data," *Expert Systems with Applications*, vol. 151, article 113361, 2020.
- [39] T. Cai, J. Li, A. S. Mian, R. Li, T. Sellis, and J. X. Yu, "Target-aware holistic influence maximization in spatial social networks," *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2020.
- [40] Q. Liu, P. Hou, G. Wang, T. Peng, and S. Zhang, "Intelligent route planning on large road networks with efficiency and privacy," *Journal of Parallel and Distributed Computing*, vol. 133, pp. 93–106, 2019.
- [41] Y. Wang, G. Yang, Y. Li, and X. Tong, "A worker-selection incentive mechanism for optimizing platform-centric mobile crowdsourcing systems," *Computer Networks*, vol. 107, article 107144, 2020.