

Research Article

A Fusion Multiobjective Empire Split Algorithm

Liang Liang 

School of Accounting College, Zibo Vocational Institute, Zibo 255000, Shandong, China

Correspondence should be addressed to Liang Liang; algorithms123@126.com

Received 23 July 2020; Revised 13 November 2020; Accepted 30 November 2020; Published 14 December 2020

Academic Editor: Kalyana C. Veluvolu

Copyright © 2020 Liang Liang. 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 last two decades, swarm intelligence optimization algorithms have been widely studied and applied to multiobjective optimization problems. In multiobjective optimization, reproduction operations and the balance of convergence and diversity are two crucial issues. Imperialist competitive algorithm (ICA) and sine cosine algorithm (SCA) are two potential algorithms for handling single-objective optimization problems, but the research of them in multiobjective optimization is scarce. In this paper, a fusion multiobjective empire split algorithm (FMOESA) is proposed. First, an initialization operation based on opposition-based learning strategy is hired to generate a good initial population. A new reproduction of offspring is introduced, which combines ICA and SCA. Besides, a novel power evaluation mechanism is proposed to identify individual performance, which takes into account both convergence and diversity of population. Experimental studies on several benchmark problems show that FMOESA is competitive compared with the state-of-the-art algorithms. Given both good performance and nice properties, the proposed algorithm could be an alternative tool when dealing with multiobjective optimization problems.

1. Introduction

With the continuous innovation of technology and the rapid development of industrial production, the multiobjective optimization problem (MOP) has gradually become a research focus in the current scientific and engineering fields [1].

The subobjectives in a multiobjective optimization problem are usually mutually constrained. There is no absolute optimal solution, and only the dominance of the solutions can be adopted to evaluate the pros and cons of the solutions. Therefore, intelligent algorithms are usually used to approximate to the Pareto front, and a set of Pareto optimal solution sets of algorithm is received. The swarm intelligence evolution algorithm is optimized in the form of swarms. In the past few decades, swarm intelligence evolution algorithm has been continuously developed and extended, such as genetic algorithm (GA) [2], differential evolution (DE) [3], evolution strategy (ES) [4], ant colony algorithm (ACO) [5], and particle swarm algorithm (PSO) [6]. They have been proven to be very effective single-objective optimization algorithms and are widely used in engineering practice, such as water cycle [7], distributed

assembly permutation flow shop problem [8], a fuel cell/battery/supercapacitor hybrid power train [9], and workforce planning [10]. One of their most significant features is that they can obtain multiple solutions in one iteration. Therefore, they have the advantages of simple implementation, low calculation cost, and high efficiency. Recently, the employment of swarm intelligence evolutionary algorithms to deal with multiobjective optimization problems has been widely studied, and a large number of swarm intelligence-based algorithms have been proposed. According to different candidate solution selection mechanisms, multiobjective evolutionary algorithms (MOEAs) can be roughly divided into the following three categories.

The first type is domination-based algorithms. This type of algorithm is represented by methods such as NSGAII [11] and SPEA2 [12]. Their basic idea is to use Pareto dominance relationships to classify groups and then calculate the density of individuals in each category. Based on the dominant relationship and density estimation, the population is fully sorted, and the relatively superior individuals are selected to enter the next generation. Crowding distance, k-nearest neighbor, ϵ -domination, and grid-divided are often used to estimate the density of individuals. Since the MOEA

algorithm based on the Pareto dominance relationship has the advantages of simple principles, easy understanding, and fewer parameters, this type of algorithm has attracted many researchers' in-depth research and extensive applications.

The second type is the MOEA algorithm based on decomposition. MOEA based on decomposition (MOEA/D) [13] is another classic MOEA algorithm framework. MOEA/D decomposes MOP into a series of subproblems and organizes these subproblems to solve at the same time to obtain an approximation of the Pareto solution set. Because the algorithm runs only once, it can achieve high computing efficiency. In MOEA/D, each subproblem is a single-objective optimization problem, so the MOEA/D algorithm framework can accommodate various types of single-objective optimization methods and local search methods. Thus, it naturally establishes the connection between heuristic algorithms and traditional mathematical programming methods. The MOEA/D framework has attracted more and more researchers' attention [14–16]. However, this algorithm requires a predefined set of weight vectors. Without prior knowledge, uniformly defined weight vectors cannot solve Pareto front with special shapes, which add unnecessary difficulty to solve MOPs [17].

The third type of algorithm is the MOEA algorithm based on evaluation indicators. This type of algorithm is represented by algorithms such as IBEA [18] and HypE [19]. The basic idea is to optimize the original MOP indirectly by directly optimizing the performance indicator of the Pareto approximation set. This is actually converting a multi-objective optimization problem into a single-objective optimization problem. In this type of method, hypervolume evaluation indicators are the most commonly used. This type of MOEA algorithm avoids a large number of comparison problems caused by Pareto dominance, but due to the large amount of calculation of the hypervolume, it is necessary to introduce methods such as Monte Carlo estimation to improve the calculation speed [19]. In addition, with the increase of the number of objectives, the calculation amount of such methods is increasing exponentially.

In the past two decades, a variety of intelligent optimization algorithms have been introduced to deal with multiobjective optimization problems and have achieved the expected results, such as particle swarm optimization, grey wolf optimization [20], imperialist competitive algorithm [21], and sine cosine algorithm [22]. Among them, the imperialist competitive algorithm (ICA) [21] is an evolutionary algorithm based on the imperialist colonial competition mechanism proposed by Atashpaz-Gargari and Lucas in 2007. It belongs to a socially inspired random optimization search method. At present, ICA has been successfully applied to a variety of optimization problems, such as ship design optimization [23], water distribution networks [24], and production scheduling problems [25]. ICA has a fast convergence speed, but it is easy to fall into a local optimum. But in dealing with multiobjective optimization problems, there is not much research on ICA. In solving the single-objective optimization problem, multiple colonies obtain the final optimal solution by approaching the empire. It should be noted that, at the end of the algorithm,

only one empire remains in the population, and the position of the empire is the optimal solution. However, when dealing with multiobjective optimization problems, a set of trade-off solutions is needed at the end of the algorithm. Therefore, inspired by ICA, a reverse ICA algorithm is studied in this paper. While the algorithm is running, the nondominated solutions are considered as the individual empire. In the end, the algorithm gets an approximate Pareto solution set composed of a group of empire individuals.

In order to prevent the population from falling into the local optimal solution, mutation operation is a commonly used strategy in multiobjective evolutionary algorithms. However, the use of mutation operations may change the original trajectory of some excellent solutions. Therefore, in order to improve the global search ability and improve the diversity of the proposed algorithm, a hybrid generation method is introduced. The sine cosine algorithm (SCA) [26] is a new metaheuristic algorithm. It is a numerical optimization calculation method based on self-organization and group intelligence based on the sine cosine function. The self-organizing model of the concept of sine and cosine was first proposed by Mirjalili. In this paper, SCA is used to update iterations of empire individuals to improve their global search capabilities.

In addition, archive strategy is also adopted by many algorithms [27–29]. The archiving strategy can better save the historical information during the algorithm running process, so as to obtain the final better candidate solution. However, the update operation of the archive strategy often occupies additional computing resources and increases the computational burden of the algorithm. In this paper, archive strategy is abandoned. The excellent solution of each generation of the population is regarded as the empire individual, and the evolutionary direction of the colony is updated. In other words, the candidate solutions after each iteration directly guide the evolution of the next generation.

In this paper, a fusion multiobjective empire split algorithm (FMOESA) is proposed. Inspired by the imperialist competitive algorithm, when dealing with multiobjective optimization, the excellent solutions are saved through the behavior of empire splitting. The approach of colonial individuals to empire individuals was also exploited when producing offspring individuals. In addition, in order to balance convergence and diversity, a new individual power assessment mechanism is also proposed. Finally, in order to verify the effectiveness of the proposed algorithm, FMOESA is compared with four state-of-the-art MOEAs on various well-known benchmark MOPs. The experimental results show that FMOESA is capable of obtaining high-quality solutions and the power assessment mechanism works well on maintaining good balance of population diversity and convergence.

The main contributions of this paper are highlighted as follows.

- (i) A new reproduction of offspring is introduced, which combines three operators: ICA, SCA, and GA. For empire individuals, the offspring are generated by performing sine and cosine operators

with other empire individuals. For colonial individuals, the offspring are generated through crossover and mutation operations with their empire. In other words, the idea of colonial individuals approaching empire individuals through crossover and mutation operations is similar to that in ICA.

- (ii) A novel power evaluation mechanism is proposed to identify individual performance. When evaluating the performance of an individual, convergence and diversity are considered simultaneously. This evaluation mechanism is not only used to distinguish between imperial and colonial individuals but also to eliminate redundant individuals in the population. This strategy can balance the convergence and diversity of the entire population.
- (iii) Archive strategy is not adopted in this paper. The outstanding individuals selected after each iteration constitute a candidate solution set, and, at the same time, they are used as the parent population of the next iteration to directly guide the process of the next iteration. This operation saves computing resources and reduces unnecessary computational complexity.

The remainder of this paper is organized as follows. Section 2 presents the main concepts of multiobjective optimization and basic content of ICA and SCA. The proposed algorithm is described in Section 3. Section 4 presents the experimental design and results. Finally, Section 5 presents conclusions and future research directions.

2. Basic Concepts and Motivation

2.1. Multiobjective Optimization Problems. In general, multiobjective optimization problems (MOP) can be defined as

$$\begin{aligned} \text{Min } F(X) &= \{f_1(X), f_2(X), f_3(X), \dots, f_m(X)\}, \\ \text{s.t. } X &\in \Omega^D, \end{aligned} \quad (1)$$

where f_i ($i \in 1, 2, 3, \dots, m$) are conflicting objective functions. Ω^D is the decision variable space. Denote a solution X by $X = (x_1, x_2, x_3, \dots, x_D)$, where D is the number of decision variables.

Definition 1. $X < X'$, which solution X is dominate solution X' , if $f_i(X) < f_i(X')$, $\forall i \in m$.

Definition 2. A solution X is Pareto optimal if and only if there is no other X' , $X' < X$.

Definition 3. A set PS of all Pareto optimal solution is called Pareto optimal set, $PS = \{X \mid \exists X' < X\}$.

Definition 4. A set that contains all the Pareto optimal objective vectors is called the Pareto front, denoted by $PF = \{f(X) \mid X \in PS\}$.

2.2. The Traditional Imperialist Competitive Algorithm. The traditional imperialist competitive algorithm (ICA) is mainly used to deal with single-objective optimization problems. It is a sociopolitical evolutionary algorithm proposed by Atashpaz-Gargari and Lucas [21] in 2007, which is a simulation of the colonial competition process of human society. The main process of the Empire competition algorithm is as follows.

- (i) Step 1: form an empire. The imperialist competitive algorithm first generates an initial country by a random method, and each country represents a solution to the problem. These countries are divided into two categories: colonial countries and colonies according to the size of the power (the quality of the solution). The former several countries became colonial countries. Then the remaining countries were assigned as colonies to the colonial countries in order according to the power of the colonial countries. The more powerful the colonial countries, the more colonies were assigned. The colonial countries and the colonies they belong to are collectively called empires.
- (ii) Step 2: assimilation and revolution. After the formation of the empire, the economic, cultural, and language attributes of the colonies will inevitably tend to belong to the colonial countries. This process is called assimilation. The goal of assimilation is to improve the quality of the solution of all countries, which can increase the influence of colonial countries on colonies. In order to be consistent with history, the process of colonial tending to empire always has a certain deviation. In extreme cases, there may even be a reverse deviation, that is, the colonial revolution. If the colony is undergoing assimilation and revolution, in the process, the power surpassed the colonial country; then the colony will replace the colonial country and establish a new empire.
- (iii) Step 3: colonial competition. There is competition between empires. The decline of the power between empires will make the colonies of weak empires deprived of powerful empires until the weak empires disappear. At the same time, new empires will continue to appear in the competition. After generations of assimilation, revolution, and competition, ideally only one empire will remain, and all countries are members of that empire. Through such a series of evolutionary operations, the algorithm finally finds the global optimal solution.

2.3. The Standard Sine Cosine Algorithm. The standard sine cosine algorithm (SCA) was proposed by Australian scholar Mirjalili in 2016 [26]. The algorithm starts with a set of random solutions and continuously approaches the global optimal solution through the search and development phases. First, the position of the solution is initialized, and the algorithm is updated as shown in

$$X_{ij}^{t+1} = \begin{cases} X_{ij}^t + r_1 \times \sin(r_2) \times |r_3 P_{gj}^t - X_{ij}^t|, & r_4 < 0.5, \\ X_{ij}^t + r_1 \times \cos(r_2) \times |r_3 P_{gj}^t - X_{ij}^t|, & r_4 \geq 0.5, \end{cases} \quad (2)$$

where X_{ij}^t is the position of the i -th individual in the j -th search space in the t -th iteration; r_1 is a linear decreasing function, and it is updated as shown in

$$r_1 = a - a \frac{t}{T_{\max}}. \quad (3)$$

In (3), a is a constant and $a = 2$; T_{\max} is the maximum number of iterations; r_2 is a random number in the range $[0, 2\pi]$; r_3 is a random number in the range $[-2, 2]$; P_{gj} is the global extremum of the population.

As shown in (2), there are four main parameters in the SCA, that is, r_1 , r_2 , r_3 , and r_4 . The parameter r_1 determines the area (or moving direction) of the next position. It can be either the area between the current solution and the target solution, or it can be an area other than the two. Parameter r_2 defines the distance of the current solution toward or away from the target solution; parameter r_3 randomly assigns a weight to the target solution, the purpose of which is to strengthen ($r_3 > 1$) or weaken ($r_3 < 1$) the distance defined by the target solution. The parameter r_4 indicates that the probability of updating the sine and cosine in (2) is equal.

The theoretical advantages of SCA in solving the global optimal solution of an optimization problem with sufficient candidate solution set size and number of iterations are as follows:

- (i) SCA is the same as other swarm intelligence optimization algorithms. Compared with the algorithm of a single candidate solution, the candidate solution set of a certain size achieves a stronger search ability and the ability to escape the local optimal trap, and SCA has sufficient random search ability.
- (ii) The optimal candidate solution is always retained during the iteration process and used as a basis for updating the candidate solution set, and there is a tendency to move to the optimal solution space during the search process.
- (iii) The randomness of its global search and local development has stronger adaptability and stability than searching and developing in two stages.

3. Proposed Algorithm

In this section, the details of the proposed fusion multi-objective empire split algorithm (FMOESA) are documented. Particularly, the framework of the proposed algorithm is presented first in Section 3.1. In the following, the different key stages in the algorithm are described.

3.1. Framework of the Proposed Algorithm. The first part of the algorithm is the initialization phase (lines 2–8 in Algorithm 1). Consistent with most optimization algorithms, N individuals are randomly generated. Immediately after,

the opposition-based learning is performed to obtain a reverse population. Then, the power of the existing $2N$ individuals is evaluated and the empire and colony are divided according to the value of power (lines 7–8 in Algorithm 1). The power assessment mechanism is described in detail in Section 3.4. At the end of the first phase, the N individuals with the greatest power value constitute the final initial population.

The second part of the algorithm is the iterative update phase (lines 10–26 in Algorithm 1). If the maximum number of fitness evaluations (FE_{max}) is not reached, the iterative cycle is not terminated. First of all, the offspring population is generated through a fusion reproduction strategy (line 11 in Algorithm 1). The detailed steps of the fusion reproduction strategy are described in Section 3.3. After fitness and power evaluation, the empire and the colony are divided (lines 12–16 in Algorithm 1). If there are less than N empire individuals, the colony individuals are arranged in reverse order of power value, and $(N - \text{empire})$ individuals are selected as candidate solutions (lines 17–19 in Algorithm 1). If the number of empire individuals is equal to N , we assign all empire individuals to X as the parent population of the next iteration (line 24 in Algorithm 1). If the number of empire individuals is greater than N , the population is deleted to obtain the final N empires with better performance (lines 21–23 in Algorithm 1). Detailed empire reduction strategy is exhibited in Section 3.6.

3.2. Initialization Based on Opposition-Based Learning.

Most of the traditional algorithms adopt the initialization method to generate the initial population. Due to the lack of prior knowledge, the probability of searching the population to a better area is greatly reduced. Opposition-based learning (OBL) [30] is an important method to enhance the performance of stochastic optimization algorithms, which greedily selects the fitness value of the objective function between the current solution and the backward solution population. In this way, the diversity of the population is enhanced, and the ability of the algorithm to approach the global optimal solution is improved.

In this paper, a reverse population is obtained based on OBL after randomly generating an initial population. In the original population and the reverse population, the optimal N candidate solutions are selected to form the initial population of the algorithm. This strategy can obtain more suitable initialization candidate solutions without prior knowledge, thereby increasing the probability of the population exploring a better region. For an individual $X_i, j = X_{i1}, X_{i2}, \dots, X_{iD}$ in the population, its reverse solution is generated according to the following:

$$X'_{ij} = a_j + b_j - X_{ij}, \quad (4)$$

where $j = 1, 2, \dots, D$ and a_j and b_j are the maximum and minimum values in the j -th dimension decision variable, respectively. All the solutions in the original population X are reversed to generate a reverse population X' . In these two populations, N individuals are selected to form the final initial population based on individual power assessment.

```

Input:  $N, FE \max, K$ 
Output:  $X$ 
(1) Part 1: Initialization
(2)  $evaluation = 0$ 
(3) Initialize  $N$  population ( $X$ )
(4) Execute the opposition-based learning operation
(5) Evaluate the fitness value of each individual
(6)  $evaluation = evaluation + 2N$ 
(7) Evaluate the power of each individual
(8) Select empires and then assign colonies to them
(9) Part 2: Iteration and Update
(10) while  $evaluation < FE \max$  do
(11)  $X_{new} = \text{Reproduction}(X)$ 
(12) Evaluate the fitness value of each individualV
(13)  $evaluation = evaluation + N$ 
(14)  $UnitX = X \cup X_{new}$ 
(15) Evaluate the power of each individual in UnitX
(16) Redistributing the empire and their colonies according to the power
(17) if  $|empire| < N$ 
(18) Sort the colonies in reverse order
(19)  $X = empire \cup (N - |empire|)colonies$ 
(20) else
(21) if  $|empire| > N$ 
(22) Implementing empire reduction strategy until the number of empires is  $N$ 
(23) end if
(24)  $X = empire$ 
(25) end if
(26) end while
(27) Return  $X$ 

```

ALGORITHM 1: Framework of the proposed FMOESA.

The detailed operation of individual power assessment is in Section 3.4.

3.3. Fusion Reproduction Strategy. In the algorithm, population X consists of two parts, the empire individuals, and their colony individuals. In general, the individual empire is superior to the individual colony. Therefore, these two parts need to be updated with different evolution operators. In this paper, sine and cosine operator (SCA) is employed to generate offspring individuals for empire individuals. For colonial individuals, their goal is to move closer to the empire and to gain greater power. Therefore, the colony obtains better convergence through crossover operations with the empire, and then performs mutation operations to increase diversity. The crossover operation adopts simulated binary crossover (SBX) [31], and the mutation operation uses polynomial mutation (PM) [3]. The detailed operation steps are shown in Algorithm 2. It should be noted that each colony belongs to an empire. In SBX operation, the colony crosses its corresponding empire (line 6 in Algorithm 2).

Through the fusion reproduction strategy, sufficient information was exchanged not only between empires but

also between colonies and empires. The former focuses on diversity, while the latter focuses on convergence.

3.4. Empire Power Evaluation Mechanism. The empire power evaluation mechanism includes two different evaluation methods. The first case is the assessment of the empire's power during the initialization phase. The second case is the assessment of the empire's power during the update iteration.

In the initialization phase, convergence is generally poor, so the assessment of individual power is mainly based on convergence. For the i -th individual X_i in the population X , its power is calculated according to the following:

$$\text{power}_i = f_i^1 + f_i^2 + \dots + f_i^m, \quad (5)$$

where m is the number of objective functions.

In the second stage, not only does the entire population need to improve convergence, but also diversity is important. Therefore, when evaluating individual power, it is necessary to consider both convergence and diversity. During the update iteration phase, for the i -th individual X_i in the population X , its power is calculated according to

```

Input:  $X, N$ 
Output:  $X_{\text{new}}$ 
(1) for each  $X_i \in X$ 
(2)   if  $X_i \in \text{empire}$ 
(3)     Randomly choose an to  $P_g$ 
(4)     Produce offspring  $X_{\text{new}_i}$  according to equation (2)
(5)   else
(6)      $X_{\text{temp}_i} = \text{SBX}(X_i, \text{empire})$ 
(7)      $X_{\text{new}_i} = \text{PM}(X_{\text{temp}_i})$ 
(8)   end if
(9) end for
(10) Return  $X_{\text{new}}$ 

```

ALGORITHM 2: $X_{\text{new}} = \text{reproduction}(X)$.

$$\text{power}_i = -C_i + D_i,$$

$$C_i = \frac{C'_i - C_{\min}}{C_{\max} - C_{\min}}, \quad (6)$$

$$D_i = \frac{D'_i - D_{\min}}{D_{\max} - D_{\min}},$$

where $C'_i = f_i^1 + f_i^2 + \dots + f_i^m$ and C_{\max} and C_{\min} are the maximum and minimum values of C , respectively. Similarly, D_{\max} and D_{\min} are the maximum and minimum values of D , respectively. D'_i is defined by the following equation:

$$D'_i = \min(\text{angle}(X_i, \text{empire})), \quad (7)$$

where $\text{angle}(A, B)$ means the angle between A and B . D'_i means the minimum angle between X_i and all empires.

It is worth noting that before calculating individual power, the convergence index (C) and the diversity index (D) need to be unified to the same order of magnitude. In other words, the value range of C_i and D_i is $[0, 1]$; thus, the value range of power is $[0, 2]$. The greater power of X_i means that X_i has better convergence and diversity.

3.5. Empire Distribution Strategy. In the population, each individual has his own role: empire or colony. In the FMOESA, individuals are assigned to corresponding roles according to their nondominated level and power. First, all nondominated solutions in the population are selected as empires. Then, the remaining individuals are sorted in descending order according to the value of power. Finally, allocate colonial individuals among the remaining populations. The specific method is as follows: we pick the individual with the largest power at a time and then assign it to the nearest empire, until $(N - \text{empire})$ individuals are selected and assigned tasks are completed.

It should be noted that the number of colonies contained in each empire is not fixed. That is, it is possible that some empires do not have colonial individuals, and some empires contain more than one colony individual.

3.6. Empire Reduction Strategy. In the later stages of the algorithm, the number of empires may exceed N . At this point, reduction strategies are essential. The main goal at this time is to select N better-performing solutions as the parent population of the next iteration. In this paper, N empires with high power are selected. The power of empire is evaluated according to (6). The empire with the lowest power is deleted one by one until the number of empires is N . The detailed steps are shown in Algorithm 3.

4. Experimental and Discussion

In order to demonstrate the effectiveness of the proposed framework, we compare its results with respect to those obtained by MOEA/D [13], dMOPSO [32], NSGAI [11], MOEA/D-STM [33], and MOEA/D-ACD [34]. Firstly, the selected performance metrics and benchmark are introduced and then the experimental setup is presented. The corresponding parameter settings are shown in Table 1. Then, the performance comparison and convergence analysis on these MOEAs are demonstrated.

4.1. Experimental Settings. We adopted eight test problems whose Pareto fronts have different characteristics. In these functions, convexity, concavity, disconnections, and multifrontality are included. The 2-objective test suite of Zitzler-Deb Thiele (ZDT) [35] (ZDT1, ZDT2, ZDT3, and ZDT6) is also adopted. For 3-objective functions, four functions are adopted which is taken from the Deb-Thiele-Laumanns Zitzler (DTLZ) test suite [36] (DTLZ1, DTLZ2, DTLZ4, and DTLZ7). We used 30 decision variables for ZDT1, ZDT2, and ZDT3. ZDT6 was tested using 10 decision variables. For DTLZ2 and DTLZ4, 12 variables were adopted. DTLZ1 and DTLZ7 were tested with 7 and 22 decision variables, respectively. The analysis on ZDT and DTLZ benchmark is presented at last.

The general parameter settings are as follows: ZDT: $m = 2, N = 100, FE_{\max} = 30000$; DTLZ: $m = 3, N = 210, FE_{\max} = 210000$. All the algorithms run 30 times independently for each test function.

```

Input:  $empire$ 
Output:  $empire_{new}$ 
(1)  $empire_{new} = empire$ 
(2) while  $|empire_{new}| > N$  do
(3) Calculate the empire's power according to equation (6).
(4)  $del = \min(power)$ 
(5) Remove  $empire_{new}(del)$  from  $empire_{new}$ 
(6) end while
(7) Return  $empire_{new}$ 

```

ALGORITHM 3: $empire_{new} = \text{reduction}(empire)$.

TABLE 1: Parameter settings of all compared algorithms.

Algorithm	Parameter settings
MOEA/D	$T = 20, pc = 0.5, pm = 1/n, \eta_c = \eta_m = 20$
dMOPSO	$T = 20, age_{max} = 2$
NSGAI	$pc = 0.9, pm = 1/n, \eta_c = \eta_m = 20$
MOEA/D-STM	$T = 20, pc = 0.5, pm = 1/n, \eta_c = \eta_m = 20$
MOEA/D-ACD	$CR = 1, pc = 0.5, pm = 1/n, \eta_c = \eta_m = 20, n_r = 2$

4.2. *Performance Metric.* In our experimental study, the widely used metrics inverted generational distance (IGD) [37] and hypervolume (HV) are chosen to evaluate the performance of each algorithm. They are used as performance metrics because they provide a joint measurement of both the convergence and diversity of the obtained solutions.

The IGD metric is defined as follows. If P^* is a set of nondominated points uniformly distributed along the true Pareto front in the objective space and P is the obtained approximation set of nondominated solutions in the objective space from a MOEA, the IGD value for the approximation set is calculated by

$$IGD(A, P^*) = \frac{\sum_{z^* \in P^*} \text{dist}(z^*, P)}{|P^*|}, \quad (8)$$

where $\text{dist}(z^*, P)$ is the minimum Euclidean distance between z^* and points in P and $|P^*|$ is the cardinality of P^* .

If $|P^*|$ is large enough to represent the Pareto front, both diversity and convergence of the approximated set P could be measured using $IGD(P, P^*)$. If the approximated set P is close enough to the true Pareto front and cannot miss any part of the whole PF, the smaller values of this metric are obtained. The advantages of IGD measure include two aspects: one is its computational efficiency and the other is its generality. For an MOEA, a smaller IGD value is desirable. The low value of IGD indicates that the obtained solution set is close to the true Pareto front.

The hypervolume (HV) metric is defined as the volume of the hypercube enclosed in the objective space by the reference point and every vector in the Pareto approximation set P . This is mathematically defined as follows:

$$HV = \{ \cup_i \text{VOL}_i | s_i \in P \}, \quad (9)$$

where s_i is a nondominated vector from Pareto approximation set P , VOL_i is the volume from the hypercube formed by the reference point and the nondominated vector s_i , and reference point is z^{ref} in the objective space.

The HV metric is applied to access both convergence and maximum spread of the solutions for the Pareto approximation set obtained with any MOESs. In addition, larger values of this measure value that the solutions are closer to the true PF and that the solutions cover a wider extension of it. In this paper, the HV metric is calculated with respect to a given reference point $z^{\text{ref}} = (z_1^{\text{ref}}, \dots, z_m^{\text{ref}})$. $z^{\text{ref}} = (2, 2)$ is designed for ZDT test functions, $z^{\text{ref}} = (1, 1, 1)$ is used for DTLZ1 and $z^{\text{ref}} = (2, 2, 2)$ is used for DTLZ2 and DTLZ4, while $z^{\text{ref}} = (2, 2, 7)$ is designed for DTLZ7.

4.3. *Results and Analysis.* Table 2 shows the mean and standard deviation (std) results on all the four ZDT and four DTLZ test instances in terms of the IGD metric. The Wilcoxon rank-sum test is recorded as p value in Table 2, and the results with confidence level 0.95 have been conducted based on the IGD values to assess the statistical significance. In Table 2, the symbols “+,” “=,” and “-,” respectively, indicate that FMOESA performances are statistically better than, equivalent to, and slightly worse than the compared algorithms.

As shown, FMOESA is the most effective algorithm in terms of the number of the best results it obtains. MOEA/D performs very competitively to FMOESA.

For the two-objective test function, FMOESA obtains the best performance on ZDT2 and ZDT3. The performance of dMOPSO is poor compared to that of its competitors, and it performs best only on ZDT1 and ZDT6. Our proposed

TABLE 2: IGD results on the eight problems.

Function		FMOESA	MOEA/D	dMOPSO	NSGAI	STM	ACD
ZDT1	Mean	4.664E-03	7.806E-03	3.928 E-03	4.922E-03	1.907E-02	6.429E-03
	Std	5.834E-04	8.352E-04	3.765 E-05	2.585E-04	3.583E-03	5.734E-03
	p value		3.013E-15	4.363E-05	2.147E-06	2.827E-19	6.083E-13
ZDT2	Mean	4.205 E-03	2.984E-02	6.926E-02	4.354E-03	1.847E-02	1.548E-02
	Std	2.475E-04	1.038E-01	1.352E-01	2.082 E-04	3.452E-03	3.154E-02
	p value		1.193E-02	8.068E-03	5.093E-14	1.748E-20	9.273E-05
ZDT3	Mean	4.947 E-03	1.312E-02	1.094E-02	7.223E-03	2.245E-02	1.072E-02
	Std	3.073 E-04	5.587E-03	7.298E-04	7.734E-03	5.827E-03	6.863E-03
	p value		9.834E-04	4.002E-26	1.105E-03	9.641E-17	8.267E-03
ZDT6	Mean	2.315E-03	1.928E-03	1.788 E-03	2.606E-03	1.509E-02	2.101E-03
	Std	1.894E-04	6.547E-05	4.863 E-05	2.521E-04	1.696E-02	4.028E-05
	p value		1.945E-05	1.356E-17	9.137E-05	3.905E-02	7.302E-05
DTLZ1	Mean	4.119E-02	1.824 E-02	3.201E+00	1.920E-02	8.202E+00	2.868E-02
	Std	1.439E-02	4.737 E-04	1.810E+01	6.101E-04	2.037E+00	5.827E-04
	p value		2.938E-10	8.076E-08	4.466E-10	3.082E-18	6.286E-10
DTLZ2	Mean	4.380 E-02	4.579E-02	4.751E-02	5.424E-02	7.442E-02	4.661E-02
	Std	1.326E-03	1.218 E-03	1.498E-03	2.337E-03	3.001E-03	2.319E-03
	p value		2.270E-05	5.386E-09	7.156E-17	1.090E-29	3.917E-09
DTLZ4	Mean	1.486E-01	5.106 E-02	1.517E-01	5.373E-02	1.582E-01	5.704E-02
	Std	1.435E-01	1.984E-03	7.267E-02	1.772 E-03	3.916E-02	2.397E-03
	p value		5.681E-04	3.569E-01	8.324E-04	1.729E-01	9.143E-04
DTLZ7	Mean	4.943 E-02	1.926E-01	1.256E-01	6.809E-02	5.321E-02	1.275E-01
	Std	2.686E-03	5.016E-03	6.278E-03	5.011E-02	1.682 E-03	6.764E-03
	p value		3.672E-36	2.579E-30	6.788E-03	5.156E-01	3.582E-3
+/-/-			5/0/3	5/1/2	6/0/2	6/2/0	5/0/3

TABLE 3: HV results on the eight problems.

Function		FMOESA	MOEA/D	dMOPSO	NSGAI	STM	ACD
ZDT1	Mean	9.153E-01	9.127E-01	9.171E-01	9.158E-01	9.154E-01	9.136E-01
	Std	1.332E-03	1.163E-03	1.302E-03	7.138E-04	1.429E-03	1.227E-03
	p value		1.112E-02	2.627E-07	2.347E-10	1.352E-01	1.945E-02
ZDT2	Mean	8.329E-01	8.204E-01	8.226E-01	8.328E-01	8.317E-01	8.247E-01
	Std	1.128E-03	5.862E-02	2.643E-03	1.207E-03	1.745E-03	2.729E-03
	p value		4.927E-03	3.669E-11	3.031E-07	1.340E-03	4.262E-03
ZDT3	Mean	9.498E-01	9.458E-01	9.426E-01	9.495E-01	9.491E-01	9.477E-01
	Std	7.704E-04	1.752E-02	1.372E-03	7.607E-04	1.168E-03	1.387E-03
	p value		3.684E-08	3.061E-11	7.797E-01	6.908E-04	7.265E-10
ZDT6	Mean	5.643E-01	7.362E-01	9.114E-01	9.001E-01	8.692E-01	7.996E-01
	Std	4.163E-01	9.525E-02	3.214E-02	2.613E-03	8.137E-02	5.208E-02
	p value		5.979E-05	1.581E-06	6.842E-03	1.237E-02	4.253E-07
DTLZ1	Mean	9.798E-01	9.801E-01	9.756E-01	9.728E-01	0.00E+00	9.767E-01
	Std	1.974E-04	7.796E-04	5.778E-04	7.884E-04	0.00E+00	3.662E-04
	p value		8.596E-01	3.032E-11	4.968E-07	1.261E-12	2.311E-11
DTLZ2	Mean	9.297E-01	9.255E-01	9.285E-01	9.258E-01	8.795E-01	9.247E-01
	Std	1.135E-03	8.147E-04	1.074E-03	9.258E-04	6.853E-03	6.286E-03
	p value		1.456E-04	2.670E-08	5.148E-07	3.062E-11	7.214E-08
DTLZ4	Mean	9.273E-01	9.274E-01	9.260E-01	9.265E-01	8.807E-01	9.268E-01
	Std	8.437E-04	1.056E-03	9.468E-04	1.003E-03	9.221E-03	3.109E-03
	p value		1.831E-01	2.449E-06	8.516E-04	3.029E-11	6.287E-06
DTLZ7	Mean	4.867E-01	4.757E-01	4.744E-01	4.790E-01	4.821E-01	4.779E-01
	Std	7.964E-03	3.742E-03	1.980E-02	2.376E-03	2.536E-03	2.817E-03
	p value		7.142E-09	3.645E-09	4.928E-11	3.082E-11	2.011E-09
+/-/-			5/2/1	6/0/2	5/2/1	6/1/1	7/0/1

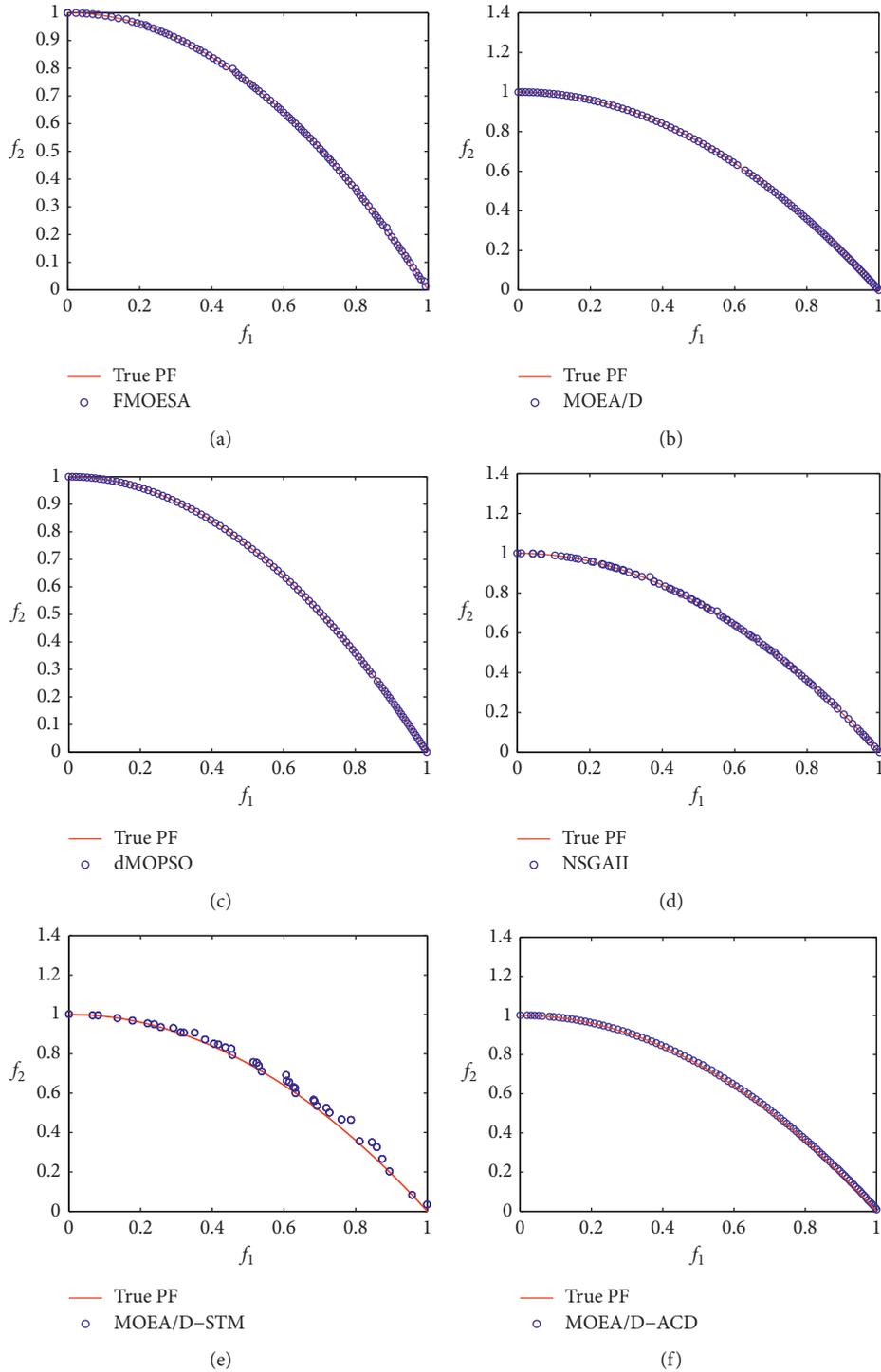


FIGURE 1: The approximate front of FMOESA, MOEA/D, dMOPSO, NSGAI, MOEA/D-STM, and MOEA/D-ACD on ZDT2. (a) FMOESA, (b) MOEA/D, (c) dMOPSO, (d) NSGAI, (e) MOEA/D-STM, and (f) MOEA/D-ACD.

FMOESA obtains significantly better performance than NSGAI and MOEA/D-STM. Except for ZDT6, the performance of MOEA/D and MOEA/D-ACD is worse than FMOESA.

For the three-objective test function, FMOESA and MOEA/D have comparable performance. FMOESA shows clear improvements over the other four compared

algorithms on DTLZ2 and DTLZ7. For MOEA/D, it obtains the best performance on DTLZ1 and DTLZ4. FMOESA performs slightly worse than NSGAI on DTLZ1 and DTLZ4. Except for the similar performance on DTLZ4 and DTLZ7, FMOESA performs better than MOEA/D-STM on other functions. MOEA/D-ACD and MOEA/D have similar performance on most of the test functions.

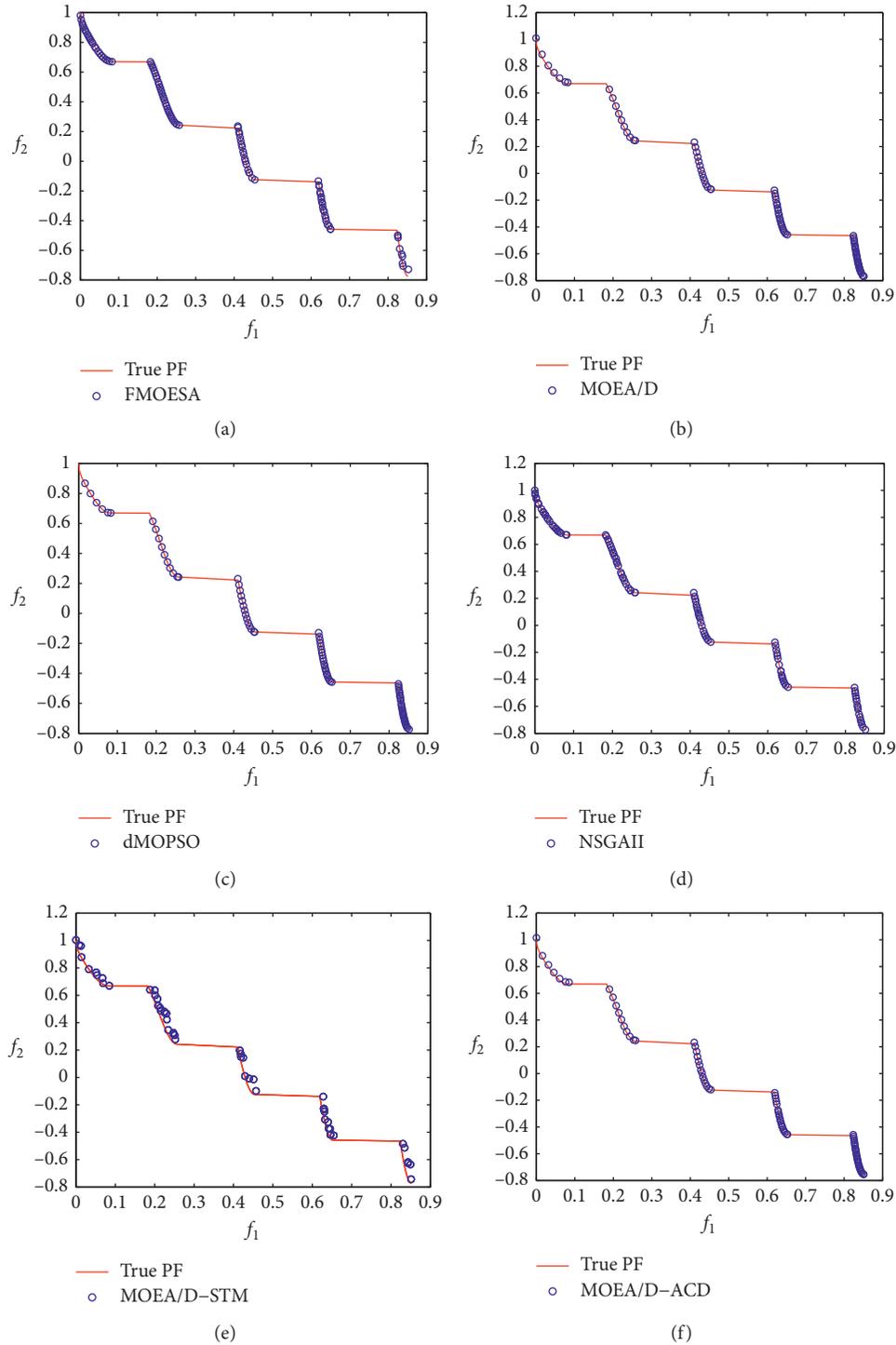


FIGURE 2: The approximate front of FMOESA, MOEA/D, dMOPSO, NSGAI, MOEA/D-STM, and MOEA/D-ACD on ZDT3. (a) FMOESA, (b) MOEA/D, (c) dMOPSO, (d) NSGAI, (e) MOEA/D-STM, and (f) MOEA/D-ACD.

As shown in Table 3, the experimental results of HV value are similar to that of IGD value. FMOESA obtains the best value on four test functions, that is, ZDT2, ZDT3, DTLZ2, and DTLZ7. FMOESA performs slightly worse than dMOPSO on ZDT1 and ZDT6. On DTLZ1 and DTLZ4, the performance of MOEA/D and FMOESA is very competitive.

Except for ZDT6, the performance of MOEA/D-ACD is worse than FMOESA.

To further compare the difference between FMOESA and the other compared algorithms, the Pareto fronts of some adopted test problems have been shown in Figures 1–4 with clearer discrimination. In these figures, the blue dots

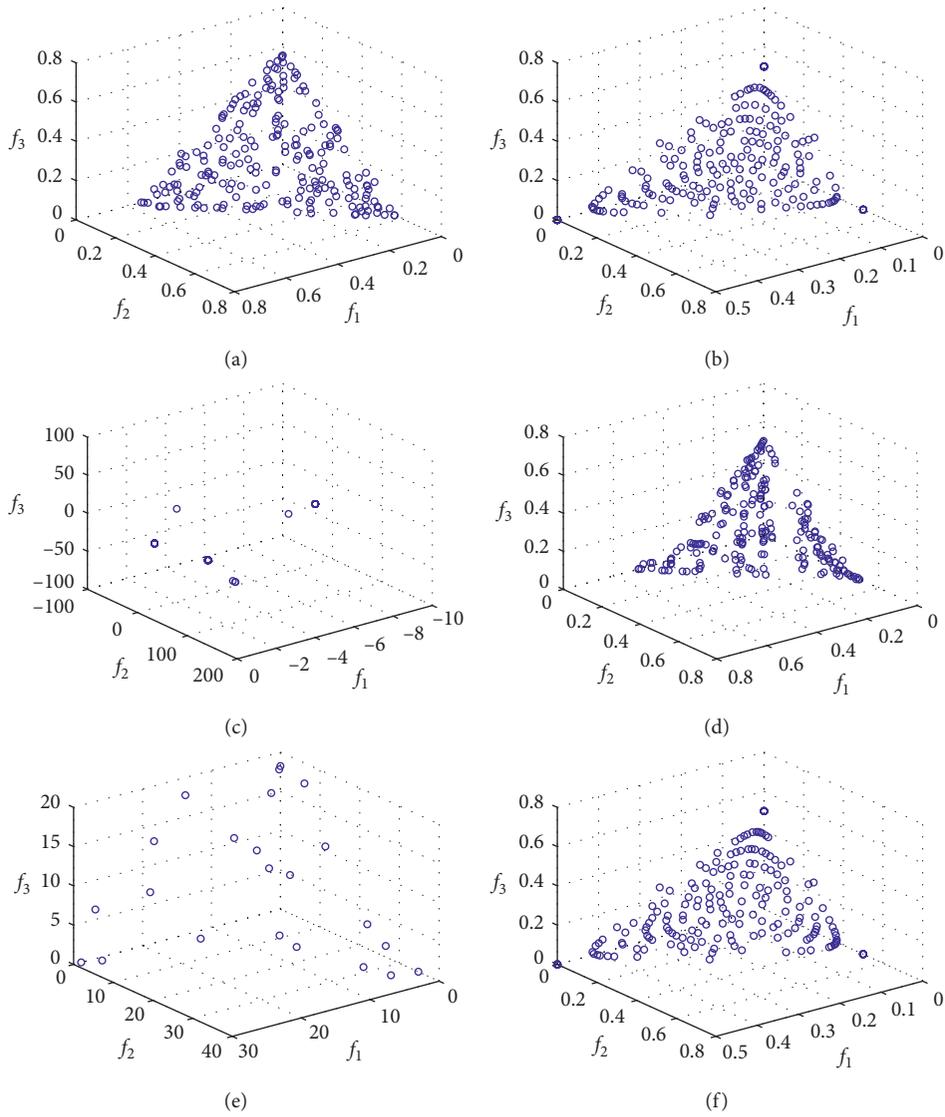


FIGURE 3: The approximate front of FMOESA, MOEA/D, dMOPSO, NSGAII, MOEA/D-STM, and MOEA/D-ACD on DTLZ1. (a) FMOESA, (b) MOEA/D, (c) dMOPSO, (d) NSGAII, (e) MOEA/D-STM, and (f) MOEA/D-ACD.

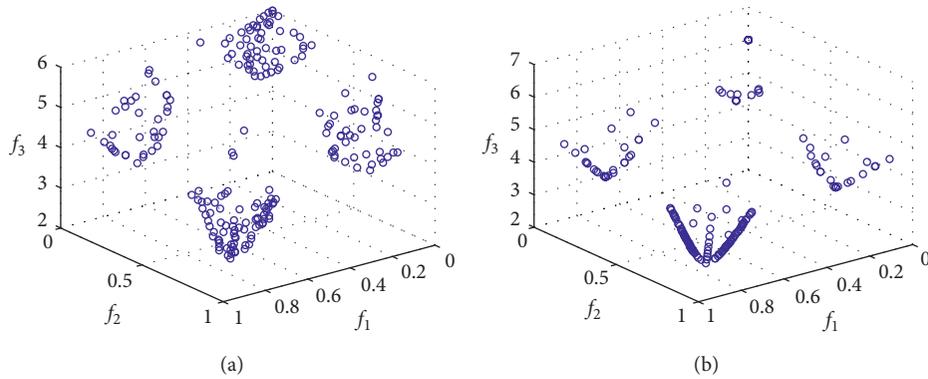


FIGURE 4: Continued.

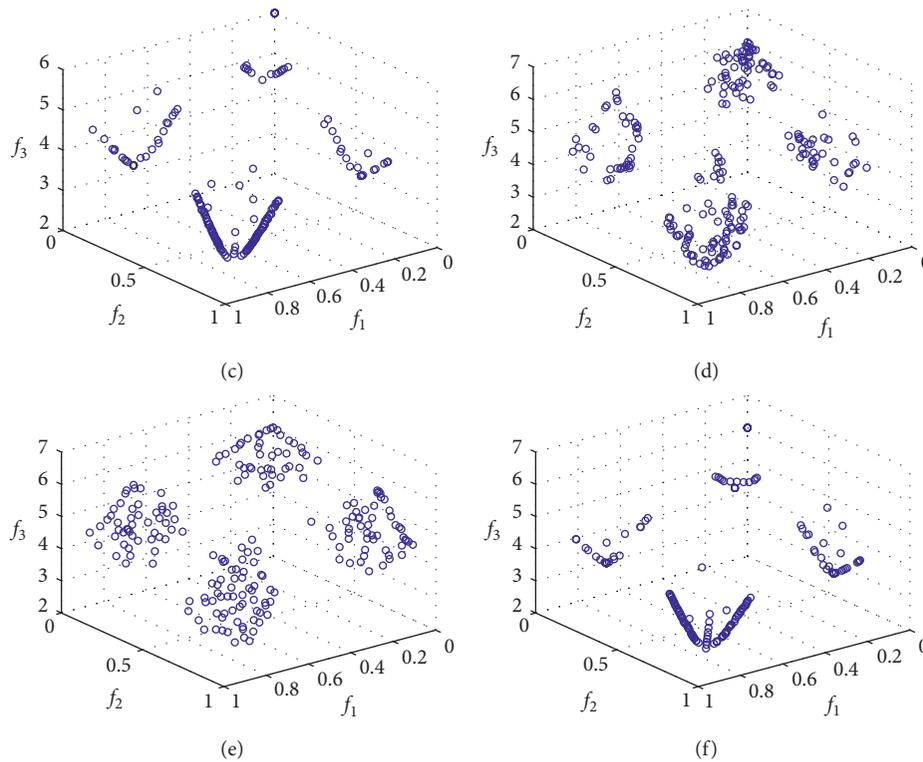


FIGURE 4: The approximate front of FMOESA, MOEA/D, dMOPSO, NSGAI, MOEA/D-STM, and MOEA/D-ACD on DTLZ7. (a) FMOESA, (b) MOEA/D, (c) dMOPSO, (d) NSGAI, (e) MOEA/D-STM, and (f) MOEA/D-ACD.

represent the approximate Pareto front obtained by each algorithm, while the solid red line represents the true PF of the test function. These figures demonstrate the abilities of those algorithms in converging to the true Pareto front.

As seen from the performance diagram of ZDT3, DTLZ1, and DTLZ7, our algorithm can well converge to its true PF, especially in DTLZ1, dMOPSO, and MOEA/D-STM may be trapped in local optimization and cannot well converge to the true PF. Besides, it can be observed that FMOESA produces better distribution than other algorithms, especially in ZDT2, ZDT3, and DTLZ7.

5. Conclusion and Future Work

In this paper, a fusion multiobjective empire split algorithm has been proposed. In the proposed algorithm, the opposition-based learning is adapted in the initialization phase. In this way, the diversity of the initial population is enhanced, and the ability of the algorithm to approach the global optimal solution is enhanced. Then, a fusion reproduction strategy is utilized to produce high-quality offspring population. Inspired by the ICA, individuals with better convergence and diversity are identified as empire individuals in the selection mechanism. This process marked the demise of the old empires and the birth of new empires. Finally, a novel type of power evaluation mechanism was proposed to select candidate solutions with superior performance. For quantifying the performance of the proposed algorithm, a series of well-designed experiments are performed against peer competitors, which include four state-of-the-art MOEAs

competitors and four multiobjective evolutionary algorithm competitors, upon a couple of widely used benchmark test suites with 2-, and 3-objective. It is demonstrated by the experimental results that FMOESA is quite competitive on the majority of all the test instances. FMOESA especially shows significant improvement over other peer algorithms in terms of the distribution of the obtained solutions.

In future research, we will investigate into more effective mechanisms for the selection process. Also, we will put efforts into the MOPs with irregularly shaped Pareto fronts.

Data Availability

The data used to support the findings of this study are included within the article. The test data are also available from the corresponding author upon request via email.

Conflicts of Interest

The author declares no conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

- [1] M. Sreedhar, S. A. N. Reddy, S. A. Chakra, T. S. Kumar, S. S. Reddy, and B. V. Kumar, "A review on advanced optimization algorithms in multidisciplinary applications," in *Recent Trends in Mechanical Engineering*, pp. 745–755, Springer, Singapore, 2020.
- [2] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

- [3] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [4] N. Hansen, "The CMA evolution strategy: a comparing review," in *Towards a New Evolutionary Computation*, pp. 75–102, Springer, Berlin, Germany, 2006.
- [5] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [6] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science MHS'95*, pp. 39–43, IEEE, San Francisco, CA, USA, 1995.
- [7] S. Khalilpourazari and S. Khalilpourazary, "An efficient hybrid algorithm based on water cycle and moth-flame optimization algorithms for solving numerical and constrained engineering optimization problems," *Soft Computing*, vol. 23, no. 5, pp. 1699–1722, 2019.
- [8] H.-Y. Sang, Q.-K. Pan, J.-Q. Li et al., "Effective invasive weed optimization algorithms for distributed assembly permutation flowshop problem with total flowtime criterion," *Swarm and Evolutionary Computation*, vol. 44, pp. 64–73, 2019.
- [9] H. Jiang, L. Xu, J. Li, Z. Hu, and M. Ouyang, "Energy management and component sizing for a fuel cell/battery/supercapacitor hybrid powertrain based on two-dimensional optimization algorithms," *Energy*, vol. 177, pp. 386–396, 2019.
- [10] S. Fidanova, O. Roeva, G. Luque, and M. Paprzycki, "Inter-Criteria analysis of different hybrid ant colony optimization algorithms for workforce planning," in *Recent Advances in Computational Optimization*, pp. 61–81, Springer, Cham, Switzerland, 2020.
- [11] 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.
- [12] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," TIK-report 103, 2001.
- [13] 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.
- [14] Z. Wang, Q. Zhang, H. Li, H. Ishibuchi, and L. Jiao, "On the use of two reference points in decomposition based multi-objective evolutionary algorithms," *Swarm and Evolutionary Computation*, vol. 34, pp. 89–102, 2017.
- [15] S. Ying, L. Li, Z. Wang, W. Li, and W. Wang, "An improved decomposition-based multiobjective evolutionary algorithm with a better balance of convergence and diversity," *Applied Soft Computing*, vol. 57, pp. 627–641, 2017.
- [16] S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 421–437, 2016.
- [17] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169–190, 2017.
- [18] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proceedings of the International Conference on Parallel Problem Solving from Nature*, pp. 832–842, Springer, Birmingham, UK, September 2004.
- [19] 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.
- [20] S. Gupta and K. Deep, "A memory-based grey wolf optimizer for global optimization tasks," *Applied Soft Computing*, vol. 93, p. 106367, 2020.
- [21] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, pp. 4661–4667, IEEE, Singapore, September 2007.
- [22] S. Gupta and K. Deep, "A hybrid self-adaptive sine cosine algorithm with opposition based learning," *Expert Systems with Applications*, vol. 119, pp. 210–230, 2019.
- [23] D. Peri, "Hybridization of the imperialist competitive algorithm and local search with application to ship design optimization," *Computers & Industrial Engineering*, vol. 137, p. 106069, 2019.
- [24] R. Moasheri and M. Jalili-Ghazizadeh, "Locating of probabilistic leakage areas in water distribution networks by a calibration method using the imperialist competitive algorithm," *Water Resources Management*, vol. 34, no. 1, pp. 35–49, 2020.
- [25] S. Mohammadi, R. Kakaie, M. Ataei, and E. Pourzamani, "Determination of the optimum cut-off grades and production scheduling in multi-product open pit mines using imperialist competitive algorithm (ICA)," *Resources Policy*, vol. 51, pp. 39–48, 2017.
- [26] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-based Systems*, vol. 96, pp. 120–133, 2016.
- [27] K. Nag, T. Pal, and N. R. Pal, "ASMiGA: an archive-based steady-state micro genetic algorithm," *IEEE Transactions on Cybernetics*, vol. 45, no. 1, pp. 40–52, 2015.
- [28] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2014.
- [29] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.
- [30] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, pp. 695–701, IEEE, Sydney, NSW, Australia, November 2005.
- [31] A. Kumar and K. Deb, "Real-coded genetic algorithms with simulated binary crossover: studies on multimodal and multiobjective problems," *Complex Systems*, vol. 9, no. 6, pp. 431–454, 1995.
- [32] S. Zapotecas Martínez and C. A. Coello Coello, "A multi-objective particle swarm optimizer based on decomposition," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pp. 69–76, ACM, Dublin, Ireland, July 2011.
- [33] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 909–923, 2014.
- [34] L. Jiao, A. Zhou, L. Wang, M. Gong, and Q. Zhang, "Constrained subproblems in a decomposition-based multi-objective evolutionary algorithm," *IEEE Transactions on*

- Evolutionary Computation: A Publication of the IEEE Neural Networks Council*, vol. 20, no. 3, pp. 475–480, 2016.
- [35] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multi-objective evolutionary algorithms: empirical results,” *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [36] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable multi-objective optimization test problems,” in *Proceedings of the 2002 Congress on Evolutionary Computation CEC’02*, pp. 825–830, IEEE, Hilton Hawaiian Village Hotel, HI, USA, May 2002.
- [37] F. Neri and C. Cotta, “Memetic algorithms and memetic computing optimization: a literature review,” *Swarm and Evolutionary Computation*, vol. 2, pp. 1–14, 2012.