

## Research Article

# A Multiobjective Brain Storm Optimization Algorithm Based on Decomposition

Cai Dai  and Xiujuan Lei 

School of Computer Science, Shaanxi Normal University, Xi'an 710119, China

Correspondence should be addressed to Xiujuan Lei; xjlei@snnu.edu.cn

Received 1 October 2018; Revised 18 December 2018; Accepted 14 January 2019; Published 22 January 2019

Academic Editor: Matilde Santos

Copyright © 2019 Cai Dai and Xiujuan Lei. 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.

Brain storm optimization (BSO) algorithm is a simple and effective evolutionary algorithm. Some multiobjective brain storm optimization algorithms have low search efficiency. This paper combines the decomposition technology and multiobjective brain storm optimization algorithm (MBSO/D) to improve the search efficiency. Given weight vectors transform a multiobjective optimization problem into a series of subproblems. The decomposition technology determines the neighboring clusters of each cluster. Solutions of adjacent clusters generate new solutions to update population. An adaptive selection strategy is used to balance exploration and exploitation. Besides, MBSO/D compares with three efficient state-of-the-art algorithms, e.g., NSGAI and MOEA/D, on twenty-two test problems. The experimental results show that MBSO/D is more efficient than compared algorithms and can improve the search efficiency for most test problems.

## 1. Introduction

Multiobjective optimization problems (MOPs) [1–4] which have conflicting objectives exist widely in the real world. Different from single-objective optimization problems, there are a series of optimal solutions for a MOP. Since the number of optimal solutions may be infinite, which leads to achieving impractically all optimal solutions at once, the good diversity and convergence of got solutions are the main goals of a good algorithm for MOPs. Although metaheuristics [5–7] can deal with MOPs, the evolutionary algorithms (EAs) [8, 9] naturally make up the set of the objective vectors of Pareto optimal solutions [10] with a population of nondominated solutions.

Multiobjective evolutionary algorithms (MOEAs) make use of the population evolution to achieve a set of nondominated solutions which are optimal solutions of the current population, which are effective methods of solving MOPs. In decades, many MOEAs [11–30] have been proposed, such as multiobjective genetic algorithms [11], multiobjective particle swarm optimization algorithms [12–14], multiobjective differential evolution algorithms [15, 16], multiobjective immune clone algorithms [17], group search optimizer [18],

and evolutionary algorithms based on decomposition [19–22]. The MOEA based on decomposition (MOEA/D) [11] may be the most widely studied in recent years. In MOEA/D, a set of uniformly distributed weight vectors and aggregate functions decomposes a MOP into a series of single or multiobjective subproblems. All subproblems are simultaneously optimized in a population evolution. Moreover, MOEA/D uses the neighbor information to improve the search efficiency and maintains the diversity of got solutions by the uniformly distributed weight vectors and aggregate functions.

In 2011, Shi proposes the brain storm optimization (BSO) [24] algorithm which is a simple and promising algorithm. It simulates the collective behavior of human being which is the brainstorming [24]. Since the invention of BSO algorithm in 2011, it had attracted many attentions in the evolutionary algorithms research community. Many works [25, 26] develop and apply the BSO algorithm. In BSO, the solutions are diverged into several clusters. One individual or more individuals are selected to generate new solutions by some genetic operators. Some multiobjective BSO algorithms [27–29] are proposed to solve MOPs. In these multiobjective

BSO algorithms, population is updated by newly solutions after solutions are clustered, which may low the speed of convergence.

In this paper, the decomposition technology is fused together with multiobjective brain storm optimization algorithm (MBSO/D) to solve MOPs. The main motivation of MBSO/D is improve the performance of multiobjective BSO algorithms (MOBSOs) by using the decomposition technology. The MBSO/D mainly includes three parts to solve effectively MOPs. First, the current solutions are automatically clustered as each subproblem is optimized and the size of each cluster is the same; second, the decomposition technology determines the neighboring clusters of each cluster, and parent solutions are selected from adjacent clusters to produce the new solutions, which can improve the search efficiency; third, an adaptive selection strategy is used to balance exploration and exploitation.

We have done some work [31, 32] on evolutionary algorithms based on decomposition recently. In the paper [31], a crossover operator based on uniform design and a selection strategy based on decomposition are designed to improve the performance of decomposition-based multiobjective evolutionary algorithms. In the paper [32], we design an initialization method of weight vectors and a new adaptive weight vector adjustment strategy. Compared to our previous works [31, 32], the main highlight originality of this manuscript is that the decomposition technology is firstly used in multiobjective brain storm optimization algorithm to improve the performance of MBSOs; the main difference is that an adaptive selection strategy in which the selection probability is updated adaptively according to the success rate of the selection strategy is used to balance exploration and exploitation.

The organization of the remaining paper is reproduced below. The related works about the MOP and BSO are briefly reviewed in Section 2. Section 3 states and describes MBSO/D in detail. Experimental results and discussions are contained in Section 4. Section 5 gives the conclusions and some possible paths for the future work.

## 2. Related Work

The related definitions of MOP, BSO, and aggregation function are stated in this section.

*2.1. Multiobjective Optimization.* The description of a MOP by mathematic formulation is as follows [30]:

$$\begin{aligned} \min \quad & F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{s.t.} \quad & x \in \Omega \end{aligned} \quad (1)$$

where  $x = (x_1, \dots, x_n) \in X \subset R^n$  is a  $n$ -dimensional decision variable whose feasible region is  $\Omega$  for (1); a MOP includes  $m$  objective functions  $f_i(x)$  ( $i = 1, \dots, m$ ). In the following, some important definitions of MOPs are displayed. For two solutions  $x, z \in \Omega$ , if each  $f_i(x) \leq f_i(z)$  and  $\|F(x) - F(z)\|_2 \neq 0$ ,  $x$  dominates  $z$  (or denoted  $x < z$ ). If the solution vector  $x$  is not dominated by any other solutions,  $x$  is called a Pareto optimal solution. The set of Pareto optimal solutions (PS) is formed by all Pareto optimal solutions. The Pareto optimal

front (PF) is the set of the objective vectors of all Pareto optimal solutions.

*2.2. Brain Storm Optimization Algorithm.* Humans may work together to solve some problems which cannot be solved by one person. Shi proposes the brain storm optimization algorithm [24] based on this human idea generation process. This algorithm is presented as follows. In a BSO algorithm, a clustering algorithm clusters the solutions into several clusters. One or three cluster(s) is (or are) randomly selected to produce new offspring. The new offspring are only compared with the best solution of the population at the same cluster to update the population. When parents are chosen from one cluster or neighboring clusters, the local search ability is strengthened. When the new offspring are generated by parents that are selected from three random clusters, the global search ability is heightened. The pseudocode of the BSO algorithm is as shown in Pseudocode 1.

## 3. The Proposed Algorithm

This paper combines the decomposition technology and multiobjective brain storm optimization algorithm (MBSO/D) to address MOPs. The major parts of this algorithm are that three strategies based on decomposition (cluster strategy, update strategy selection strategy) are developed. Three strategies will be pointed out in this section.

*3.1. Motivation.* The main motivation of this paper is to design a MOEA to achieve a set of solutions which evenly distribute on the true PF. BSO algorithm uses the group information to address the problems, which can help MOEAs to improve their performance. MOEAs use the neighboring information to optimal population, which can improve the performance of MOEAs. In this paper, this optimization idea is utilized to enhance the performance of multiobjective BSO algorithm. First, new solutions update the population by using the updated strategy of MOEA/D [21]; then, solutions are automatically clustered as each subproblem is optimized. Second, the decomposition technology determines the neighboring clusters of each cluster, and parent solutions are selected from adjacent clusters. Third, an adaptive selection strategy is used to balance exploration and exploitation.

The MOEA/D decomposes a MOP into a series of subproblems by weight vectors and aggregate functions. For each solution, it and some of its neighbor solutions are selected as parents to generate a new solution. Then, some of its neighbor solutions are updated by aggregate functions and the new solution. So, all subproblems are simultaneously optimized in a population evolution. Each aggregate function makes some solutions converge to the corresponding weight vector, which can improve the convergence of the algorithm. In addition, the diversity of solutions is maintained by the uniformly distributed weight vectors. The main advantages of MOEA/D are that the diversity of obtained solutions can be determined by the given weight vectors; neighbor's information is used to generate new offspring, which can improve the search efficiency.

**(I) Initialization:** Generate an  $N$  solutions randomly or by a problem-specific method and evaluate these solutions  
**While the stopping criterion is not met do**  
**Clustering:** cluster  $N$  solutions into  $M$  clusters by a clustering algorithm;  
**Generating new solutions:** randomly select one or two cluster(s) to produce new solutions.  
**Updating:** compare the new solution with the existing solution with the same cluster; the better one is kept and recorded as the new solution.

PSEUDOCODE 1: The pseudo code of the BSO algorithm.

3.2. *Cluster and Update.* In this work, each given weight vector fixes a cluster, and the size of each cluster is made the same. The best solution of each cluster is determined by corresponding weight vector  $\lambda^i = (\lambda_1^i, \dots, \lambda_m^i)$  and the aggregation function [31]. The formula is as follows:

$$x_{best}^i = \min \{g^{TE}(x | \lambda^i, Z^*), x \in P^i\} \quad (2)$$

where  $g^{TE}(x | \lambda^i, Z^*) = \max_{1 \leq j \leq m} \{ |f_j(x) - z_j^*| / \lambda_j^i \}$  is a aggregation function,  $Z^* = (z_1^*, \dots, z_m^*)$  is the reference point, and  $x_{best}^i$  is the current best solution of  $i$ -th cluster  $P^i$ . After a new solution is produced by some cluster, one solution of neighboring clusters of this cluster may be replaced by this new solution. If the best solution is not better than the new solution, randomly select a solution (not the best solution) from the cluster, which is replaced by this new solution, and the current best solution of this cluster will be redefined by (2). This update strategy can reduce the computing effort. Moreover, in this paper, one new solution will replace two solutions at most, which can balance the convergence and diversity. One solution is replaced and the diversity of the population is maintained, but the rate of convergence may be lower. Many solutions are replaced and the rate of convergence can be enhanced, but the diversity of the population is reduced.

3.3. *Selection Strategy and Crossover Operator.* An effective selection strategy can help crossover operators to perform search work more effectively. In this BSO algorithm, according to an adaptive selection probability, a new solution will be produced by one cluster or three clusters. The adaptive selection probability is calculated by the following formulation:

$$p^t = \frac{s_{one}}{s_{one} + s_{th} + \delta} + 0.1p^{t-1} \quad (3)$$

where  $t$  is the evolution generation;  $p^{t-1}$  is the selection probability of the  $t-1$ -generation;  $s_{one}$  and  $s_{th}$  are the number of kept new solutions which are generated by one cluster and three clusters, respectively;  $\delta$  is to make the denominator not 0 and is set to a small number. If a random number is smaller than  $p^t$ , parents are selected from one cluster to produce a new solution, which may enhance the exploitation; otherwise, parents are selected from three clusters to produce a new solution, which can strengthen the exploration.

Based on the selection probability  $p^t$  and the decomposition technology, a selection strategy is designed to balance

local search and global search. For each weight vector, work out its  $T$  closet weight vectors according to the Euclidean distances of any two weight vectors. For each  $i = 1, \dots, N$ , set  $B(i) = \{K * i_1, \dots, K * i_T\}$  where  $\lambda^{i_1}, \dots, \lambda^{i_T}$  are the  $T$  closet weight vectors to  $\lambda^i$  and  $K$  is the size of each cluster. Then set

$$P = \begin{cases} \{(i-1) * K + 1, \dots, i * K\}, & \text{if rand1} < p^t \\ B(i), & \text{if rand2} < J \\ \{K * 1, \dots, K * N\}, & \text{otherwise,} \end{cases} \quad (4)$$

where  $\text{rand1} \in [0, 1]$  and  $\text{rand2} \in [0, 1]$  are two random numbers;  $J$  is set to 0.9 as the same in [21];  $x^{K*i}$  is the current best solution of  $i$ -th cluster. When the set  $P$  is set, randomly select two indexes  $r2$  and  $r3$  from  $P$  and generate one solution from  $x^{K*i}$ ,  $x^{r2}$ , and  $x^{r3}$  by the following formula:

$$x_j^{new} = \begin{cases} x_j^{K*i} + L(x_j^{r2} - x_j^{r3}), & \text{if rand}(0, 1) < CR \\ x_j^{K*i} & \text{otherwise} \end{cases} \quad (5)$$

where  $L \in [0, 2]$  is a scale factor which controls the length of the exploration vector ( $x^{r2} - x^{r3}$ );  $CR$  is a constant value namely crossover rate;  $j = 1, \dots, n$  and  $x_j^{r2}$  indicates the  $j$ -th component of  $x^{r2}$ ;  $x^{new} = (x_1^{new}, x_2^{new}, \dots, x_n^{new})$ . If the solution  $x^{K*i}$  dominates  $x^{new}$ ,  $x^{new}$  is set as the following formula:

$$x_j^{new} = \begin{cases} x_j^{K*i} - L(x_j^{r2} - x_j^{r3}) & \text{if } x_j^{new} \neq x_j^{K*i} \\ x_j^{K*i} & \text{otherwise} \end{cases} \quad (6)$$

If the new solution generated by (5) is dominated by  $x^{K*i}$ , the new solution generated by (6) may be better than  $x^{K*i}$  with big probability. Equation (6) serves to enhance the search efficiency. When two indexes  $r2$  and  $r3$  are randomly selected from  $\{(i-1) * K + 1, \dots, i * K\}$  or  $B(i)$ , the local search can be carried out to improve the convergence; when two indexes  $r2$  and  $r3$  are randomly selected from  $\{K * 1, \dots, K * N\}$ , the global search can be carried out.

3.4. *Pseudocode of the Proposed Algorithm.* In this subsection, the pseudocode for the multiobjective brain storm optimization algorithm based on decomposition (MBSO/D) is displayed as shown in Pseudocode 2.

```

Input:
  MOP (1)
  A stopping criterion
  N: the number of direction vectors (the clusters)
  K: the size of the cluster
  T: the number of the neighborhood,  $0 < T < N$ 
   $\lambda^1, \lambda^2, \dots, \lambda^N$ : a set of  $N$  uniformly distributed weight vectors
Output: Objective vectors:  $\{F(x^K), F(x^{2*K}), \dots, F(x^{N*K})\}$ 
Initialization: Generate an initial population  $x^1, x^2, \dots, x^{N*K}$  determine  $Z = (z_1, \dots, z_m)$ ; randomly cluster
the initial population into  $N$  clusters with size  $K$  and determine the
best solution of each cluster; determine  $B(i) = \{i_1, \dots, i_T\}$ , ( $i = 1, \dots, N$ ), where
 $\lambda^{i_1}, \dots, \lambda^{i_T}$  are the  $T$  closest weight vectors to  $\lambda^i$ ; set  $p^t = 0.5$ 
While the function evaluation times are less than the maximum function evaluation times do
  Set  $s_{one} = 0$  and  $s_{th} = 0$ 
  For  $i = 1, \dots, N$ , do
    According to Eq. (4), randomly select two indexes  $r2$  and  $r3$ , use  $x^{K*i}, x^{r2}$  and  $x^{r3}$  to generate offspring
     $x^{new} = (x_1^{new}, x_2^{new}, \dots, x_n^{new})$  by Eq. (5).
    If  $x^{K*i}$  dominates  $x^{new}$ ,  $x^{new}$  is regenerated by Eq. (6).
    Update of  $Z$ : For  $j = 1, \dots, m$ , if  $z_j < f_j(x^{new})$ , then set  $z_j = f_j(x^{new})$ 
    Update of Neighboring Solutions: set  $H = 0$  and  $Q = B(i)$ 
    While  $H \leq 2$  and  $Q \neq \emptyset$ 
       $k = Q(1)$ ;
      If  $g^{TE}(x^{new} | \lambda^k, z) < g^{TE}(x^{k*K} | \lambda^k, z)$ 
        randomly select a number  $h$  from  $\{(k-1) * K + 1, \dots, k * K - 1\}$ , set  $x^h = x^{k*K}$ ,
         $F(x^h) = F(x^{k*K})$ ,  $x^{k*K} = x^{new}$  and  $F(x^{k*K}) = F(x^{new})$ .
         $H = H + 1$ .
        If  $x^{new}$  generated by one cluster
           $s_{one} = s_{one} + 1$ ;
        Else
           $s_{th} = s_{th} + 1$ ;
        End if
      End if
       $Q(1) = []$ ;
    End while
  End for
  Use Eq. (3) to update  $p^t$ .
end for
end while

```

PSEUDOCODE 2: The pseudo code of the algorithm MBSO/D.

In MBSO/D, firstly, solutions in initial population are randomly clustered into  $N$  clusters with size  $K$  and the best solution of each cluster is determined by (2); secondly, for the best solution of each cluster (the for loop in pseudocode), some solutions are selected according to (3)-(4), the best solution and selected solutions generate a new solution by (5) or (6), and these produce good offspring; thirdly, some neighboring solutions of the best solution are updated by the

new solution and aggregation function (the while loop in pseudocode), to help the algorithm obtain a set of solutions with good diversity and convergence; finally, we update (3) to balance exploitation and exploration.

In this MBSO/D, the MOPs are solved by updating the neighboring solutions of each solution. We use the aggregation function  $g^{TE}(x | \lambda^i, Z^*) = \max_{1 \leq j \leq m} \{|f_j(x) - z_j^*| / \lambda_j^i\}$  to update the neighboring solutions of each solution, so the

population is being updated at the same time. This updated strategy makes solution converge to subproblems  $g^{TE}(x | \lambda^i, Z^*)$ ; this ensures the convergence of the algorithm. In addition, the weight vectors are uniformly distributed; this ensures the diversity of the algorithm. So, this update strategy can solve MOPs. Moreover, in original BSO, solutions are clustered by some clustering algorithms. But, in this MBSO, each subproblem is considered a class; the solutions are automatically clustered while the population is updated. The clustering in this MBSO does not require additional computation.

This multiobjective BSO algorithm compares with other multiobjective BSO algorithms, e.g., MBSO-C [28] (MBSO with Cauchy mutation) and MBSO-DE [27] (MBSO with differential evolution); the major differences are as follows.

In MBSO-C and MBSO-DE, after the solutions are cluttered by some clustering methods, some solutions of the population are replaced by the newly generated solutions. In these MBSOs, the population is firstly updated by the newly generated solutions; then, solutions are automatically clustered as each subproblem is optimized. In MBSO-C and MBSO-DE, the newly generated solutions compare with other solutions of the same cluster to update this cluster, which may reduce the pressure of convergence; in this algorithm, aggregate function is used to update the population, which may enhance pressure of convergence.

## 4. Experimental Studies

In this section, the performance of MBSO/D will be verified by comparing it with existing multiobjective optimization algorithms, e.g., NSGAI [11] and MOEA/D [21]. Two test suites are used in this experiment: ten test problems of the CEC 2009 competition [33] and seven DTLZ problems [34]. The number of decision variables is placed to 30, 7, 21, 12, and 22 for F1-F10, DTLZ1 and DTLZ3, DTLZ2 and DTLZ4-DTLZ6, DTLZ7, respectively.

*4.1. Performance Metrics.* Three performance metrics are adopted to quantify the performances of algorithms: generational distance (GD) [35], inverted generational distance (IGD) [35], and hypervolume indicator (HV) [36]. GD can evaluate the convergence performance of an algorithm. Both IGD and HV can evaluate the diversity and convergence of solutions obtained by an algorithm. Roughly 2000 points uniformly sampled on the Pareto fronts are used in the calculation of IGD and GD for each test problem. In the calculation of HV, the reference point  $Z^* = (Z_1^*, \dots, Z_m^*)$  is set to  $Z_i^* = 1.1 * \max\{f_i(x) | x \in PS\}$ ,  $i = 1 \dots, m$ . Moreover, the Wilcoxon Rank-Sum test [37] is employed at a significance level of 0.05. It tests whether the performance of MBSO/D is significantly better (“+”), statistically similar (“=”), or significantly worse (“-”) than/as that obtained by the compared algorithms.

*4.2. Parameters Setting.* All algorithms are implemented by using the MATLAB language and run independently for thirty times with the maximal number of function evaluations 100 000 on all test problems. For fair comparisons, the

population size and the maximal number of function evaluations of the compared algorithms are the same as this work, and other parameters of NSGAI and MOEA/D are the same as the original literature. In MBSO/D,  $CR = 0.5$  and  $L = 0.5$ ; the size of each cluster is set to 5; the population size is set to 105 for all compared algorithms on each test problem; the size of neighborhood list T is set to 0.1N; the probability of choosing mate subproblem from its neighborhood J is set to 0.9.

*4.3. Algorithm Performance Analyses.* The statistical results of the GD, IGD, and HV metrics obtained by each MOEA are posted in Tables 1 and 2. The performance of MBSO/D will be checked by these statistical results. In these tables, the highlighted bold results indicate the best results.

*4.3.1. Comparisons of MBSO/D with NSGAI and MOEA/D.* This subsection presents the comparison results on IGD, GD, and HV in seventeen problems. Table 1 gives the mean and standard deviation values for three comparison algorithms. We can obtain from Table 1 that, in the form of the HV and IGD metrics, the results obtained by MBSO/D are better than those obtained by NSGAI and MOEA/D on more than fourteen problems, which indicates that the final solutions obtained by MBSO/D have a better diversity than those obtained by NSGAI and MOEA/D and have a good convergence; the mean values of GD metric contained by MBSO/D are bigger than those obtained by MOEA/D and NSGAI on more than nine problems; these imply that MBSO/D can obtain a set of solutions with better convergence than MOEA/D and NSGAI on most test problems.

Moreover, MBSO/D outperforms NSGAI and MOEA/D in solving DTLZ1 and DTLZ3; this emphasizes that the selection strategy and crossover operators have the advantage in solving multiple local fronts problems. NSGAI is better at solving DTLZ5. The reason for the higher mean IGD value for MBSO/D is because the update strategy is not suitable for MOPs with degenerated PF. According to the median values of IGD metric, Figure 1 plots the nondominated solutions obtained by MBSO/D. It is observed that the obtained objective vectors of nondominated solutions uniformly cover the full true PF on these seventeen test problems except for problems F5, F6, F9, and F10. MBSO/D cannot converge to the true PF on problems F5, F6, F9, and F10, because the maximal number of function evaluations is small and the sizes of decision variables are big.

The main goal of MOEAs is to obtain a set of solutions with good diversity and convergence. To test the algorithms' ability to accomplish this goal, Figure 2 shows the evolution of the average IGD metric values of the current population with the number of function evaluations for F1, F2, DTLZ1, and DTLZ3. Figure 2 indicates that MBSO/D converges, in terms of the number of function evaluations, much faster than MOEA/D and NSGAI in minimizing the IGD metric value for these four test problems. In other words, for MOPs with many local PFs (DTLZ1 and DTLZ3), MBSO/D can quickly converge to true PF and maintain the diversity of obtained solutions; for two-objective problems (F1 and F2), MBSO/D can find a set of solutions better than MOEA/D and NSGAI. These comparisons suggest that the selection

TABLE 1: IGD, GD, and HV obtained by IMOEA/DA, MOEA/D, and NSGAI.

Problems		IGD		GD		HV	
		mean	std	mean	std	mean	std
F1	MBSO/D	<b>0.0172</b>	0.0068	<b>0.0108</b>	0.0020	<b>0.8448</b>	0.0073
	MOEA/D	0.0750(+)	0.0513	0.0150(+)	0.0082	0.7622(+)	0.0515
	NSGAI	0.0951(+)	0.0355	0.0135(+)	0.0028	0.7298(+)	0.0517
F2	MBSO/D	<b>0.0050</b>	0.0001	<b>0.0025</b>	0.0003	<b>0.8679</b>	0.0004
	MOEA/D	0.0137(+)	0.0032	0.0050(+)	0.0006	0.8538(+)	0.0036
	NSGAI	0.0095(+)	0.0005	0.0101(+)	0.0006	0.8614(+)	0.0007
F3	MBSO/D	<b>0.0326</b>	0.0101	0.0168	0.0043	0.8231	0.0123
	MOEA/D	0.1194(+)	0.0939	0.0145(-)	0.0067	0.7538(+)	0.0604
	NSGAI	0.0351(+)	0.0255	<b>0.0130(-)</b>	0.0015	<b>0.8328(-)</b>	0.0195
F4	MBSO/D	<b>0.0038</b>	0.0001	<b>0.0004</b>	0.0001	<b>0.5382</b>	0.0001
	MOEA/D	0.0084(+)	0.0009	0.0040(+)	0.0002	0.5300(+)	0.0023
	NSGAI	0.0056(+)	0.0003	0.0050(+)	0.0003	0.5348(+)	0.0004
F5	MBSO/D	<b>0.4074</b>	0.0718	0.4799	0.0896	0.0767	0.0599
	MOEA/D	0.4342(+)	0.1547	0.3598(-)	0.1601	<b>0.1199(-)</b>	0.0926
	NSGAI	0.4643(+)	0.1087	<b>0.3094(-)</b>	0.1325	0.0782(=)	0.0817
F6	MBSO/D	<b>0.1334</b>	0.0677	0.3879	0.9174	<b>0.4485</b>	0.0811
	MOEA/D	0.2116(+)	0.1383	0.1467(-)	0.0554	0.3911(+)	0.0946
	NSGAI	0.1935(+)	0.0897	<b>0.0079(-)</b>	0.0100	0.3900(+)	0.0889
F7	MBSO/D	<b>0.0168</b>	0.0024	0.0222	0.0097	<b>0.6770</b>	0.0048
	MOEA/D	0.0794(+)	0.1676	0.0166(-)	0.0127	0.6165(+)	0.1386
	NSGAI	0.0782(+)	0.1203	<b>0.0067(-)</b>	0.0016	0.6066(+)	0.1002
F8	MBSO/D	<b>0.0903</b>	0.0080	0.0443	0.0093	0.6444	0.0198
	MOEA/D	0.0939(+)	0.0114	<b>0.0136(-)</b>	0.0025	<b>0.6516(=)</b>	0.0158
	NSGAI	0.1482(+)	0.0242	0.7099(+)	0.6017	0.5616(+)	0.0427
F9	MBSO/D	<b>0.0770</b>	0.0135	0.1484	0.1113	<b>0.9797</b>	0.0276
	MOEA/D	0.1039(+)	0.0448	<b>0.0857(-)</b>	0.0375	0.9067(+)	0.0638
	NSGAI	0.1666(+)	0.0709	0.9917(+)	0.8967	0.7647(+)	0.1496
F10	MBSO/D	<b>0.3442</b>	0.0772	7.3988	3.3987	<b>0.3175</b>	0.1025
	MOEA/D	0.3597(+)	0.2113	<b>0.1555(-)</b>	0.1425	0.3100(+)	0.1442
	NSGAI	0.3505(+)	0.0682	3.4759(-)	3.5608	0.1811(+)	0.0624
DTLZ1	MBSO/D	<b>0.0186</b>	0.0001	<b>0.0071</b>	0.0001	<b>0.1404</b>	0.0001
	MOEA/D	0.0314(+)	0.0016	0.0075(+)	0.0002	0.1295(+)	0.0011
	NSGAI	0.0356(+)	0.0500	0.0183(+)	0.0611	0.1335(+)	0.0204
DTLZ2	MBSO/D	<b>0.0522</b>	0.0037	<b>0.0184</b>	0.0009	<b>0.7380</b>	0.0027
	MOEA/D	0.0813(+)	0.0053	0.0209(+)	0.0010	0.6673(+)	0.0107
	NSGAI	0.0692(+)	0.0021	0.0234(+)	0.0013	0.7011(+)	0.0057
DTLZ3	MBSO/D	<b>0.0624</b>	0.0515	<b>0.0177</b>	0.0017	<b>0.7327</b>	0.0577
	MOEA/D	0.0807(+)	0.0048	0.0204(+)	0.0009	0.6709(+)	0.0115
	NSGAI	0.0692(+)	0.0024	0.0231(+)	0.0146	0.7113(+)	0.0062
DTLZ4	MBSO/D	<b>0.0530</b>	0.0027	<b>0.0174</b>	0.0008	<b>0.7421</b>	0.0018
	MOEA/D	0.0822(+)	0.0053	0.0202(+)	0.0010	0.6788(+)	0.0155
	NSGAI	0.1299(+)	0.1628	0.0218(+)	0.0037	0.6748(+)	0.0853
DTLZ5	MBSO/D	0.0186	0.0015	0.0077	0.0047	0.4281	0.0012
	MOEA/D	0.0121(-)	0.0030	<b>0.0006(-)</b>	0.0001	0.4174(+)	0.0083
	NSGAI	<b>0.0053(-)</b>	0.0003	0.0011(-)	0.0002	<b>0.4378(-)</b>	0.0003
DTLZ6	MBSO/D	0.0207	0.0003	0.0035	0.0040	<b>0.4260</b>	0.0002
	MOEA/D	<b>0.0118(-)</b>	0.0038	<b>0.0004(+)</b>	0.0001	0.4190(+)	0.0092
	NSGAI	0.0555(+)	0.0260	0.0668(+)	0.0249	0.3745(+)	0.0291
DTLZ7	MBSO/D	<b>0.0784</b>	0.0005	<b>0.0071</b>	0.0005	<b>1.2913</b>	0.0013
	MOEA/D	0.1558(+)	0.0239	0.0079(+)	0.0010	0.9319(+)	0.0031
	NSGAI	0.1124(+)	0.0935	0.0157(+)	0.0132	1.0256(+)	0.0024

“+” means that MBSO/D outperforms its competitor algorithm, “-” means that MBSO/D is worse than its competitor algorithm, and “=” means that the competitor algorithm has the same performance as MBSO/D.

TABLE 2: GD and  $\Delta$  obtained by MBSO/D, MBSO-DE, and MBSO-C on five ZDT problems.

Problems		GD		$\Delta$	
		best	mean	best	mean
ZDT1	MBSO/D	0.0009	0.0009	0.0563	0.0569
	MBSO-DE	<b>0.0010</b>	0.0011	<b>0.1008</b>	0.1257
	MBSO-C	0.0695	0.0912	0.5105	0.5529
ZDT2	MBSO/D	<b>0.0007</b>	0.0008	<b>0.0270</b>	0.0274
	MBSO-DE	<b>0.0007</b>	0.0008	0.0997	0.1253
	MBSO-C	0.0725	0.0905	0.4898	0.5588
ZDT3	MBSO/D	<b>0.0010</b>	0.0012	<b>0.4126</b>	0.4187
	MBSO-DE	0.0011	0.0012	0.4126	0.4188
	MBSO-C	0.0443	0.0589	0.5708	0.6364
ZDT4	MBSO/D	<b>0.0022</b>	0.0023	<b>0.1405</b>	0.1460
	MBSO-DE	2.9322	13.8379	1.2958	1.3968
	MBSO-C	6.4966	15.2905	0.8362	0.9699
ZDT6	MBSO/D	<b>0.0008</b>	0.0009	<b>0.0062</b>	0.0082
	MBSO-DE	0.0037	0.0040	0.5322	0.5346
	MBSO-C	0.0580	0.0813	0.6969	0.7425

strategy of MBSO/D is good at balancing exploration and exploitation; the updated strategy can maintain the diversity.

*4.3.2. Comparisons of MBSO/D with MBSO-C and MBSO-DE.* In this subsection, we compare MBSO/D with two MBSOs (MBSO-C [28] and MBSO-DE [27]) to testify the performance of MBSO/D. MBSO/D compares with MBSO-C and MBSO-DE on five ZDT [23] problems (ZDT1-ZDT4 and ZDT6). The experimental results of MBSO-C and MBSO-DE are directly obtained from the original literatures. For fair comparisons, the population size of MBSO/D is set to 100 for five problems; the maximal number of function evaluations is set to 300 000 which is smaller than the number of MBSO-C and MBSO-DE; other parameters settings are adopted in Section 4.2.

The mean and the best values of GD and  $\Delta$  [11] metrics obtained by MBSO/D, MBSO-C, and MBSO-DE are displayed in Table 2. It can be obtained from this table that, in the form of these two metrics, the results obtained by MBSO/D are better than those obtained by MBSO-C and MBSO-DE, which illustrate that MBSO/D can achieve a set of solutions with better diversity and converge than MBSO-C and MBSO-DE on all these five problems. These comparisons imply that MBSO/D is better at balancing exploration and exploitation than MBSO-C and MBSO-DE on these problems. According to the median values of GD metric, the objective vectors of nondominated solutions obtained by MBSO/D on the five ZDT test problems are plotted in Figure 3, which can visually show the superior performance of MBSO/D. These indicate that MBSO/D can effectively approach the true PFs.

*4.3.3. Comparisons of MBSO/D with MOHS/D.* In this subsection, MOHS/D is used to compare with MBSO/D [38].

MOHS/D hybridizes MOEA/D with harmony search algorithm to solve MOPs. MBSO/D compares with MOHS/D on five ZDT problems and seven DTLZ problems. The experimental results of MOHS/D are directly got from the original literature to make a fair comparison, and the population sizes of MBSO/D are set to 100 and 200 for two-objective and three-objective problems, respectively. For each problem, the maximal number of function evaluations is the same as MOHS/D [38]; other parameters settings are the same as in Section 4.2.

Table 3 summarizes the best, median, and worst values of IGD metric obtained by MBSO/D and MOHS/D on these twelve test problems. It can be seen from this table that the median values of IGD metric obtained by MBSO/D are smaller than those obtained by MOHS/D on five ZDT problems, DTLZ3 and DTLZ6 and that the median values of IGD metric obtained by MBSO/D are bigger than those obtained by MOHS/D on other five DTLZ problems. Those indicate that we can obtain a set of solutions with better coverage and diversity than MOHS/D on most problems of these twelve test problems.

## 5. Conclusions

In this paper, we proposed a multiobjective brain storm optimization algorithm, called MBSO/D, based on the idea of decomposition. In this approach, the update strategy of MOEA/D [21] is used in the MBSO, which can well balance the diversity and convergence. Simultaneously, a selection strategy is adopted to improve the search efficiency of this algorithm. Moreover, MBSO/D compares with NSGAI, MOEA/D, MBSO-C, MBSO-DE, and MOHS/D on some test sets with complicated PS or many local PFs. According to the performance analyses, MBSO/D shows competitive

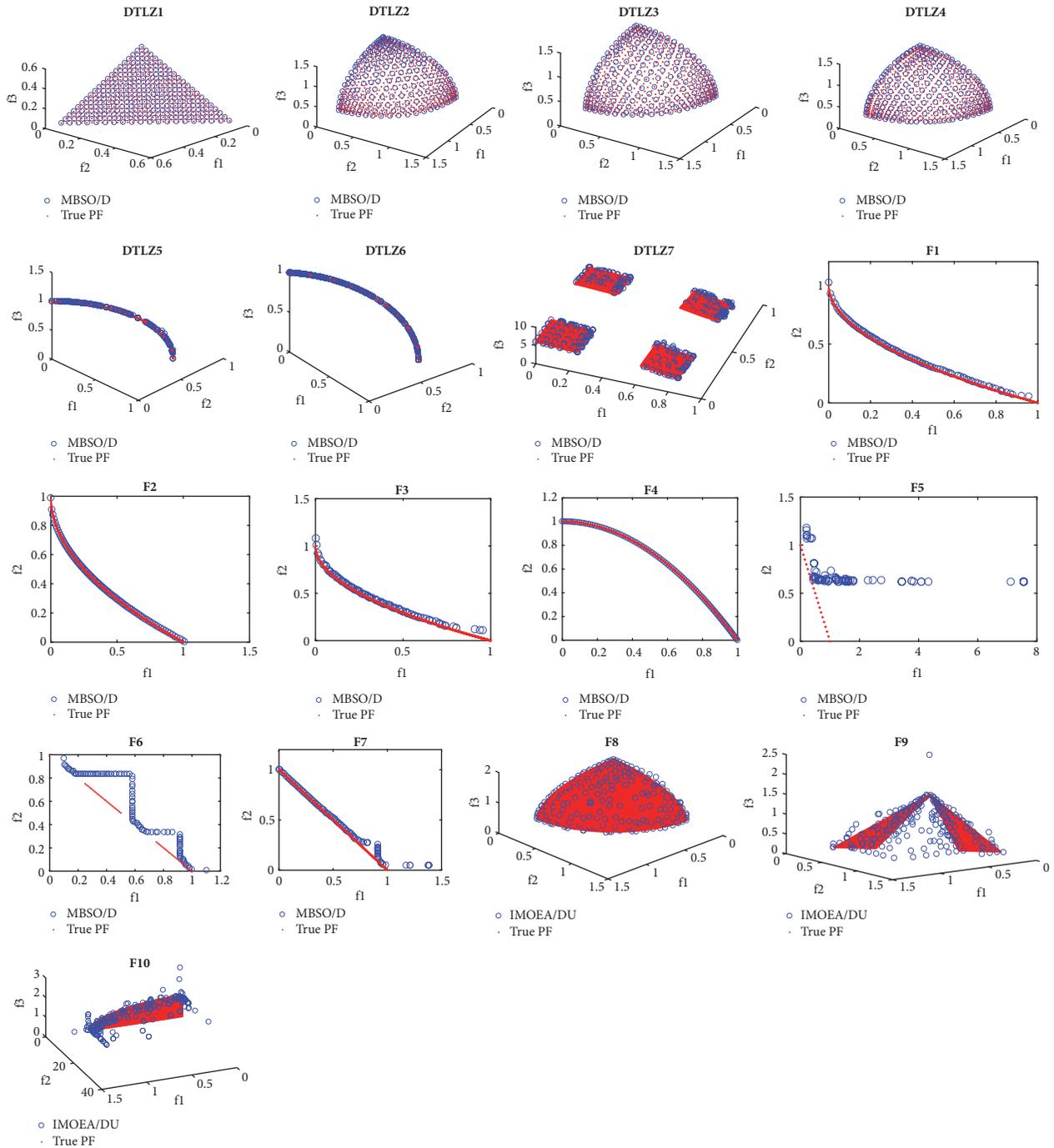


FIGURE 1: According to the median values of IGD metric, nondominated solutions obtained by MBSO/D on seven DTLZ problems and ten F1-F10 problems.

performances on most test problems against for comparisons MOEAs. These results imply that the update strategy and selection strategy in MBSO/D can help MBSO to obtain a set of solutions with good diversity and convergence. However, for a few benchmark functions, the proposed algorithm shows shortcoming because the update strategy is not suitable for some MOPs with degenerated PF. In the future, we will study that this algorithm is used to solve real-world problems.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

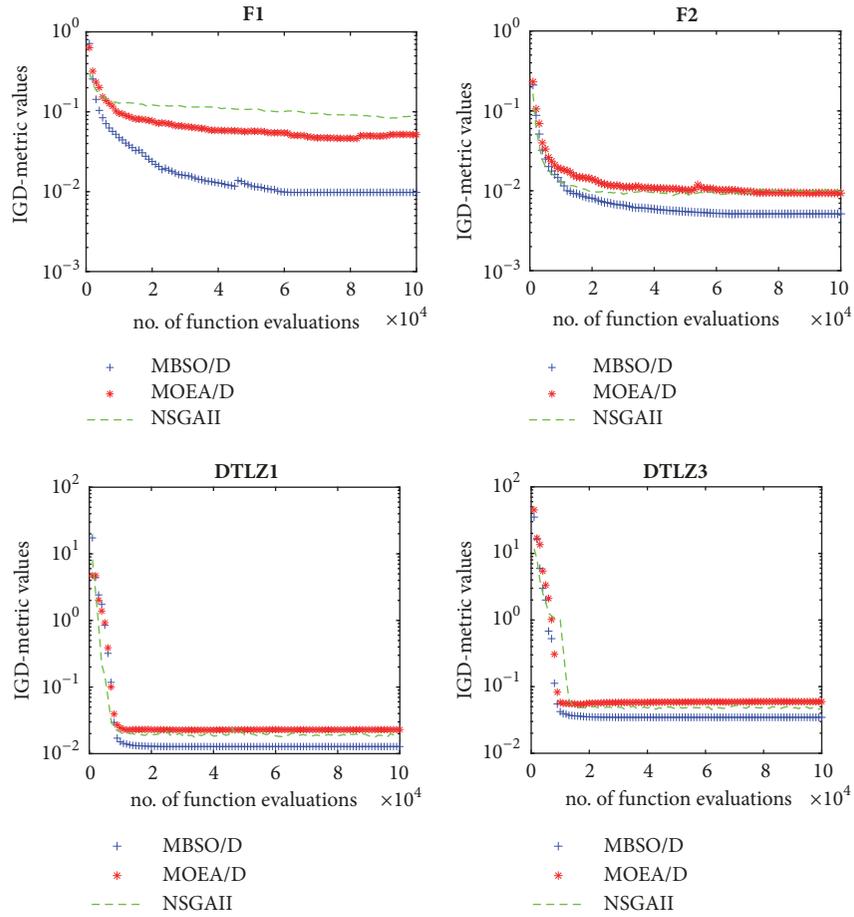


FIGURE 2: Evolution of the mean of IGD metric values for F1, F2, DTLZ1, and DTLZ3.

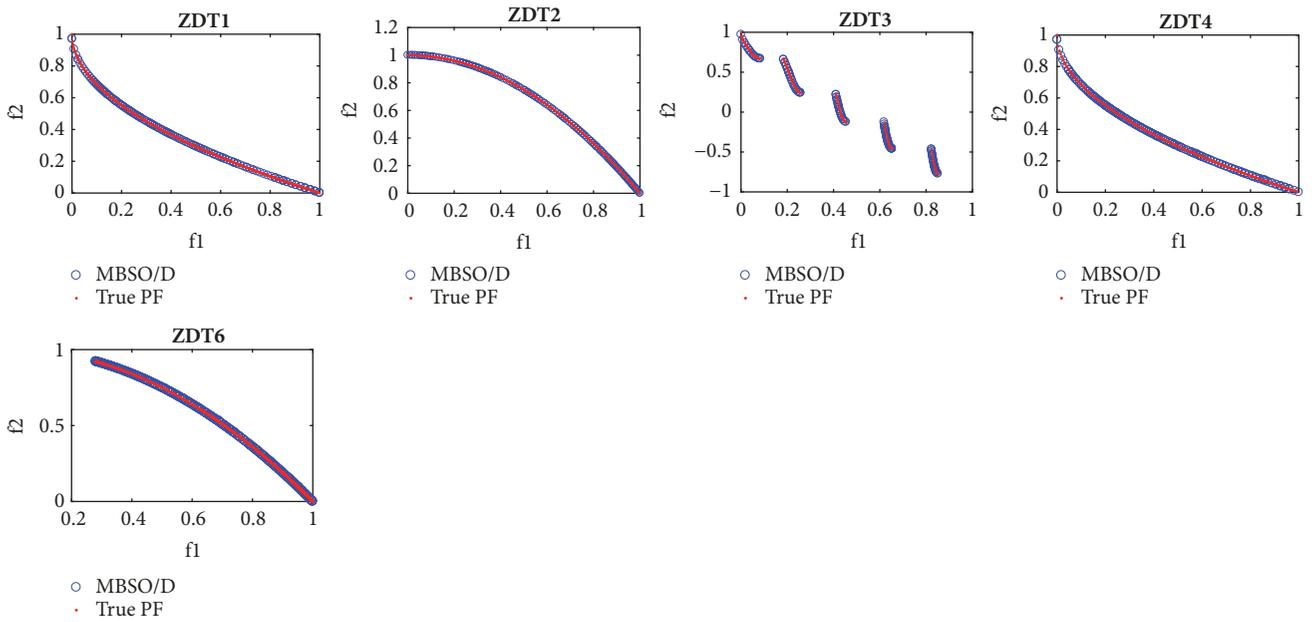


FIGURE 3: According to the median values of GD metric, nondominated solutions obtained by MBSO/D on five ZDT problems.

TABLE 3: IGD obtained by MBSO/D and MOHS/D on five ZDT problems and seven DTLZ problems.

Problems		IGD		
		Best	Median	Worst
ZDT1	MBSO/D	1.62e-3	<b>1.71e-3</b>	2.04e-3
	MOHS/D	1.37e-3	1.86e-3	2.18e-3
ZDT2	MBSO/D	8.42e-4	<b>1.02e-3</b>	2.45e-3
	MOHS/D	2.26e-3	2.26e-3	3.01e-3
ZDT3	MBSO/D	6.14e-4	<b>9.52e-4</b>	1.26e-3
	MOHS/D	9.15e-4	1.19e-3	1.76e-3
ZDT4	MBSO/D	1.86e-4	<b>1.62e-4</b>	5.21e-3
	MOHS/D	1.76e-4	1.64e-4	4.23e-3
ZDT6	MBSO/D	1.62e-4	<b>1.87e-4</b>	3.41e-4
	MOHS/D	1.59e-4	1.91e-4	2.42e-4
DTLZ1	MBSO/D	1.77e-2	1.81e-2	1.92e-2
	MOHS/D	4.69e-03	<b>1.28e-02</b>	4.16e-2
DTLZ2	MBSO/D	5.12e-2	5.22e-2	5.44e-2
	MOHS/D	3.23e-3	<b>4.08e-3</b>	4.71e-3
DTLZ3	MBSO/D	6.13e-2	<b>6.24e-2</b>	8.26e-2
	MOHS/D	8.62e-2	1.33e-1	2.00e-1
DTLZ4	MBSO/D	6.23e-3	5.30e-2	5.74e-2
	MOHS/D	8.43e-3	<b>9.76e-3</b>	1.06e-2
DTLZ5	MBSO/D	5.49e-3	1.86e-2	2.61e-2
	MOHS/D	1.34e-3	<b>1.40e-3</b>	1.45e-3
DTLZ6	MBSO/D	1.57e-2	<b>2.07e-2</b>	2.51e-2
	MOHS/D	1.21e-2	2.21e-2	3.35e-2
DTLZ7	MBSO/D	6.25e-2	7.85e-2	9.21e-2
	MOHS/D	2.89e-2	<b>3.00e-2</b>	3.07e-2

## Acknowledgments

This work was supported by National Natural Science Foundation of China (nos. 61502290, 61806120, 61672334, 61673251, and 61401263), the Fundamental Research Funds for the Central Universities (GK201901010), China Postdoctoral Science Foundation (no. 2015M582606), and Natural Science Basic Research Plan in Shaanxi Province of China (no. 2016JQ6045).

## References

- [1] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, and Y. Jin, "Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 2, pp. 97–111, 2017.
- [2] R. Roy and J. Mehnen, "Dynamic multi-objective optimisation for machining gradient materials," *CIRP Annals - Manufacturing Technology*, vol. 57, no. 1, pp. 429–432, 2008.
- [3] X. Xue and Y. Wang, "Using memetic algorithm for instance coreference resolution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 580–591, 2016.
- [4] S. Palaniappan, S. Zein-Sabatto, and A. Sekmen, "Dynamic multiobjective optimization of war resource allocation using adaptive genetic algorithms," in *Proceedings of the 2001 IEEE SoutheastCon*, pp. 160–165, 2001.
- [5] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evolutionary Computation*, vol. 19, no. 4, pp. 561–595, 2011.
- [6] L. Zuo, L. Shu, S. Dong, C. Zhu, and T. Hara, "A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing," *IEEE Access*, vol. 3, pp. 2687–2699, 2015.
- [7] Y. Zhang, D.-W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 64–75, 2017.
- [8] W. K. Mashwani, A. Salhi, M. Asif, R. Adeeb, and M. Sulaiman, "Enhanced version of multi-algorithm genetically adaptive for multiobjective optimization," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 12, pp. 279–287, 2015.
- [9] W. K. Mashwani, A. Salhi, O. Yeniay, H. Hussian, and M. Jan, "Hybrid non-dominated sorting genetic algorithm with adaptive operators selection," *Applied Soft Computing*, vol. 56, pp. 1–18, 2017.
- [10] X. Ma, F. Liu, Y. Qi et al., "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 275–298, 2016.
- [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] M. Kiani and S. H. Pourtakdoust, “State estimation of nonlinear dynamic systems using weighted variance-based adaptive particle swarm optimization,” *Applied Soft Computing*, vol. 34, pp. 1–17, 2015.
- [13] B. Tang, Z. Zhu, H.-S. Shin, A. Tsourdos, and J. Luo, “A framework for multi-objective optimisation based on a new self-adaptive particle swarm optimisation algorithm,” *Information Sciences*, vol. 420, pp. 364–385, 2017.
- [14] H. Han, W. Lu, and J. Qiao, “An adaptive multiobjective particle swarm optimization based on multiple adaptive methods,” *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2754–2767, 2017.
- [15] Q. H. Lin, Q. F. Zhu, P. Huang et al., “A novel hybrid multi-objective immune algorithm with adaptive differential evolution,” *Computer and Operations Research*, vol. 62, pp. 95–111, 2015.
- [16] X. P. Wang and L. X. Tang, “An adaptive multi-population differential evolution algorithm for continuous multi-objective optimization,” *Information Sciences*, vol. 348, pp. 124–141, 2016.
- [17] R. Shang, L. Jiao, F. Liu, and W. Ma, “A novel immune clonal algorithm for MO problems,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 35–50, 2012.
- [18] Z.-H. Zhan, J. Li, J. Cao, J. Zhang, H. S.-H. Chung, and Y.-H. Shi, “Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems,” *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 445–463, 2013.
- [19] H. Zhang, X. Zhang, X.-Z. Gao, and S. Song, “Self-organizing multiobjective optimization based on decomposition with neighborhood ensemble,” *Neurocomputing*, vol. 173, pp. 1868–1884, 2016.
- [20] 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, 2015.
- [21] 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.
- [22] M. Ming, R. Wang, Y. Zha, and T. Zhang, “Pareto adaptive penalty-based boundary intersection method for multi-objective optimization,” *Information Sciences*, vol. 414, pp. 158–174, 2017.
- [23] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: empirical results,” *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [24] Y. Shi, “An optimization algorithm based on brainstorming process,” *International Journal of Swarm Intelligence Research*, vol. 2, no. 4, pp. 35–62, 2011.
- [25] S. Cheng, Q. Qin, J. Chen, and Y. Shi, “Brain storm optimization algorithm: a review,” *Artificial Intelligence Review*, vol. 46, no. 4, pp. 445–458, 2016.
- [26] X. Ma, Y. Jin, and Q. Dong, “A generalized dynamic fuzzy neural network based on singular spectrum analysis optimized by brain storm optimization for short-term wind speed forecasting,” *Applied Soft Computing*, vol. 54, pp. 296–312, 2017.
- [27] X. Guo, Y. Wu, L. Xie, S. Cheng, and J. Xin, “An adaptive brain storm optimization algorithm for multiobjective optimization problems,” in *Advances in Swarm and Computational Intelligence*, vol. 9140 of *lecture notes in computer science*, pp. 365–372, 2015.
- [28] Y. Shi, J. Xue, and Y. Wu, “Multi-objective optimization based on brain storm optimization algorithm,” *International Journal of Swarm Intelligence Research*, vol. 4, no. 3, pp. 1–21, 2013.
- [29] L. Xie and Y. Wu, “A modified multi-objective optimization based on brain storm optimization algorithm,” in *Advances in Swarm Intelligence*, vol. 8795 of *Lecture Notes in Computer Science*, pp. 328–339, 2014.
- [30] D. A. Van Veldhuizen, *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, Air Force Institute of Technology Wright, Ohio, Ohio, USA, 1999.
- [31] C. Dai and X. Lei, “An improvement decomposition-based multi-objective evolutionary algorithm with uniform design,” *Knowledge-Based Systems*, vol. 125, pp. 108–115, 2017.
- [32] C. Dai and X. Lei, “A Decomposition-based multi-objective evolutionary algorithm with adaptive weight adjustment,” *Complexity*, vol. 2018, Article ID 1753071, 20 pages, 2018.
- [33] Q. F. Zhang and P. N. Suganthan, “Final report on CEC’09 MOEA competition,” Tech. Rep., the School of CS and EE, University of Essex, UK and School of EEE, Nanyang Technological University, Singapore, 2009, <http://dces.essex.ac.uk/staff/qzhang/moeacompetition09.htm>.
- [34] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable multi-objective optimization test problems,” in *Proceedings of the Congress on Evolutionary Computation (CEC ’02)*, pp. 825–830, May 2002.
- [35] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, “Performance assessment of multiobjective optimizers: an analysis and review,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.
- [36] K. Deb, A. Sinha, and S. Kukkonen, “Multi-objective test problems, linkages, and evolutionary methodologies,” in *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference (GECCO ’06)*, pp. 1141–1148, Washington, Wash, USA, 2006.
- [37] S. Robert, J. Torrie, and D. Dickey, *Principles and Procedures of Statistics: A Biometrical Approach*, McGraw-Hill, New York, NY, USA, 1997.
- [38] I. A. Doush, M. Q. Bataineh, and M. El-Abd, “The Hybrid Framework for Multi-objective Evolutionary Optimization Based on Harmony Search Algorithm,” in *RTIS*, vol. 756, pp. 134–142, *Lecture Notes in Real-Time Intelligent Systems*, 2017.

