


# Balancing Exploration and Exploitation With Decomposition-Based Dynamic Multi-Objective Evolutionary Algorithm

Qing Zhang, Huanggang Normal University, Hubei, China

Ruwang Jiao, China University of Geosciences, Wuhan, China

Sanyou Zeng, China University of Geosciences, Wuhan, China

 <https://orcid.org/0000-0002-0795-9092>

Zhigao Zeng, Hunan University of Technology, Zhuzhou, China

## ABSTRACT

Balancing exploration and exploitation is a crucial issue in evolutionary global optimization. This paper proposes a decomposition-based dynamic multi-objective evolutionary algorithm for addressing global optimization problems. In the proposed method, the niche count function is regarded as a helper objective to maintain the population diversity, which converts a global optimization problem to a multi-objective optimization problem (MOP). The niche-count value is controlled by the niche radius. In the early stage of evolution, a large niche radius promotes the population diversity for better exploration; in the later stage of evolution, a niche radius close to 0 focuses on convergence for better exploitation. Through the whole evolution process, the niche radius is dynamically decreased from a large value to zero, which can provide a sound balance between the exploration and exploitation. Experimental results on CEC 2014 benchmark problems reveal that the proposed algorithm is capable of offering high-quality solutions in comparison with four state-of-the-art algorithms.

## KEYWORDS

Evolutionary Algorithm, Exploration and Exploitation, Global Optimization, MOEA/D, Multi-Objective Optimization

## 1. INTRODUCTION

In most science, business, and engineering fields, global optimization problems often arise with the purpose of locating the global optimum. Many approaches have been proposed to deal with global optimization problems. Nevertheless, when the objective of global optimization problems is nonlinear, non-convex or non-differentiable, traditional mathematical approaches may become inefficient, or even fail to work. Evolutionary algorithm (EA) is a kind of population based iterative heuristic optimization paradigm. Over the last two decades, EAs have been widely studied and applied for many scientific and real-world optimization problems with promising results (Del, Osaba, & Molina, 2019). However, the basic EAs are easy to fall into some local optima when tackling complicated global optimization problems with a large number of local optima. Apparently, the key to solving

DOI: 10.4018/IJCINI.20211001.0a25

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

global optimization problems using EAs is how to handle the relationship between the exploration and exploitation.

The exploration refers to the capability of an EA to investigate undiscovered regions in the decision space to find more potential solutions, while the exploitation means the ability of an EA to apply the knowledge of discovered promising solutions to further improve their fitness. A loss of exploration might result in stagnation in suboptimal areas, producing the effect known as premature convergence. By contrast, a low exploitation rate cannot make the population convergence to the global optimum. In practice, exploration and exploitation are conflicting with each other. Thus, in order to achieve good performances on global optimization problems, the exploration and exploitation abilities should be well balanced (Singh & Deep, 2019). It is extensively recognized that an EA should put more emphasis on exploration at the early stage of the evolution and high exploitation at the later stage of the evolution.

To balance the exploration and exploitation in EAs, this paper makes an attempt to adopt decomposition-based dynamic MOEA to solve complicated global optimization problems. Firstly, a global optimization problem is converted to a dynamic bi-objective optimization problem by taking the niche count function as a helper objective. Subsequently, a decomposition-based MOEA: MOEA/D-M2M (Liu, Gu, & Zhang, 2014), with DE operator (Storn & Price, 1997) is employed to solve the transformed dynamic bi-objective optimization problem. The dynamic version of MOEA/D-M2M is named DMOEA/D-M2M. DMOEA/D-M2M decomposes the converted bi-objective optimization problem into a number of bi-objective optimization subproblems which are easier to solve. In DMOEA/D-M2M, optimization of the original objective is beneficial to the population convergence while minimization of the niche-count objective is helpful to the population diversity. Throughout the entire evolution stage, the gradually-decreasing-to-zero niche radius is capable of achieving a tradeoff between exploration and exploitation.

The remainder of the paper is organized as follows. Section 2 briefly reviewed related works. Section 3 depicts some preliminary. Section 4 presents the dynamic multi-objective technique, which a global optimization problem is converted to a dynamic bi-objective optimization problem. Section 5 shows the detailed of proposed DMOEA/D-M2M algorithm. Section 6 conducts the experiments and makes a comparison of DMOEA/D-M2M with four competitors. Section 7 makes some further discussions. At last, Section 8 draws conclusions.

## 2. RELATED WORKS

Multi-objective optimization is one of the most active research areas in the field of EAs (Zhou, Qu, & Li, 2011). An enormous amount of efforts has been devoted to handling multi-objective optimization problems (MOPs), which results in a large number of multi-objective EAs (MOEAs). After nearly three decades of development, a number of MOEAs can work very well on bi-objective problems (Ishibuchi, Tsukamoto, & Nojima, 2008). The canonical MOEAs could roughly divide into three classes: methods based on Pareto domination (Deb, Pratap, & Agarwal, 2002), methods based on indicator (Bader & Zitzler, 2011), and methods based on decomposition (Zhang & Li, 2007). Decomposition-based methods are well-known ones in traditional multi-objective optimization design. Decomposition-based MOEAs decompose a MOP into a number of scalar optimization subproblems and optimize them in a collaborative manner using an EA (Zhang & Li, 2007) (Zhou & Zhang, 2015) (Gee, & Arokiasami, 2016) (Trivedi, Srinivasan, & Sanyal, 2017) (Zhang & Zhen, 2017) (Li & Zhang, 2019). In 2006, Zhang and Li proposed MOEA/D (Zhang & Li, 2007), which decomposes an MOP into several scalar optimization subproblems, and these scalar optimization subproblems are formulated by decomposition method using uniformly distributed weight vectors. In MOEA/D, all subproblems are solved simultaneously by employing an EA and evolving a population of individuals. MOEA/D-M2M (Liu, Gu, & Zhang, 2014) is an improved version of MOEA/D. Different from most decomposition-based MOEAs which decompose a MOP into certain single-objective subproblems,

MOEA/D-M2M decomposes a MOP into a number of simple multi-objective subproblems, and then solving these multi-objective subproblems in a cooperative way. Decomposition-based MOEAs have also extended to address many-objective optimization problems (Yuan, Xu, & Wang, 2015) (Chen, & Zhou, 2018)(Chen, Liu, & Tan, 2019) (Zhang, Gong, & Gu, 2019) (Li, Deb, & Zhang, 2019)(Liu, Wang, & Huang, 2020).

MOEAs have been successfully applied to solve single-objective optimization problems, which can be grouped into three types (Segura, Coello, & Miranda, 2016):

- Methods that transform a constrained single-objective optimization problem into an unconstrained multi-objective optimization problem;
- Methods that called multiobjectivization who try to convert a single-objective global optimization problem to a MOP by transforming its fitness landscape;
- Methods that consider diversity as a helper objective.

Dynamic multi-objective technique (Jiao, Zeng, & Alkasassbeh, 2017) is a very recent work which belongs to the third scheme. It transforms a single-objective optimization problem into an equivalent dynamic multi-objective optimization problem (DMOP) which regards the niche count function as a helper objective. The niche count function is controlled by the niche radius. At the early evolution stage, a large niche radius enables the search pay more attention to the exploration to investigate promising regions. At the later evolution stage, the niche radius gradually decreased to zero as the environmental changes, which allows a MOEA concentrates on exploitation to guarantee the convergence.

### 3. BASIC DEFINITIONS

#### 3.1. Global Optimization Problem

Without loss of generality, a minimization global optimization problem can be formulated as:

$$\begin{aligned} & y = f(xx) \\ \min \quad & xx = (x_1, \dots, x_D) \in X \\ \text{where} \quad & X = \{xx \mid ll \leq xx \leq uu\} \\ & ll = (l_1, \dots, l_D), uu = (u_1, \dots, u_D) \end{aligned} \quad (1)$$

where  $x$  represents the solution vector,  $X$  indicates the decision space.  $ll$  and  $uu$  denote the lower bound and upper bound of the decision space. The main purpose of the global optimization problem is to search the global optima solution while satisfy the bound constraints ( $ll \leq xx \leq uu$ ).

#### 3.2. Multi-Objective Optimization Problem

Similarly, a minimization MOP can be expressed as:

$$\begin{aligned} & y = ff(xx) = (f_1) \\ \min \quad & xx = (x_1, \dots, x_D) \in X \\ \text{where} \quad & X = \{xx \mid ll \leq xx \leq uu\} \\ & ll = (l_1, \dots, l_D), uu = (u_1, \dots, u_D) \end{aligned} \quad (2)$$

where  $x$  is the solution vector,  $X$  denotes the decision space,  $ll$  and  $uu$  are the lower and upper bounds of the decision space. The objective vector  $f(xx)$  involves  $m$  objective functions.

For any two individuals  $xx_1$  and  $xx_2$ ,  $xx_1$  is said to dominate  $xx_2$  ( $xx_1 \prec xx_2$ ), if  $f_i(xx_1) \leq f_i(xx_2)$  for all  $i \in \{1, \dots, m\}$ , and  $f_i(xx_1) < f_i(xx_2)$  for at least one  $i \in \{1, \dots, m\}$ . An individual  $xx^*$  is

called Pareto optimal if there is no other individual  $x$  satisfying  $xx \prec xx^*$ . The *Pareto set (PS)* is the set of all the Pareto optimal solutions, the *Pareto front (PF)* is the set of images of solutions in the *PS*. Different from global optimization problem, the purpose of the MOP is to find a set of non-dominated solutions which are evenly distributed on the *PF*. It is worth to note that when  $m=1$ , problem (2) is equivalent to problem (1).

#### 4. TRANSFORMATION OF A GLOBAL OPTIMIZATION PROBLEM INTO A DYNAMIC BI-OBJECTIVE OPTIMIZATION PROBLEM

The idea of employing dynamic multi-objective techniques for handling unconstrained and constrained single-objective optimization problems borrows from (Jiao, Zeng, & Alkasassbeh, 2017) (Zeng, Jiao, & Li, 2017) (Zeng, Jiao, & Li, 2019) (Jiao, Zeng, & Li, 2019). In (Jiao, Zeng, & Alkasassbeh, 2017), a helper objective, niche count function, is introduced to balance the exploration and exploitation and, hence, avoid an EA from being trapped in some local optima.

**Definition 1: Niche-count** Assume that the combination of parent population and offspring population has  $2NP$  individuals  $\mathbf{Q} = xx_1, xx_2, \dots, xx_{2NP}$ , the niche-count function for  $xx \in \mathbf{Q}$ :

$$nc(xx | \mathbf{Q}, \sigma) = \sum_{i=1, xx_i \neq xx}^{2NP} sh(xx, xx_i), \quad (3)$$

where the sharing function between  $x_1$  and  $x_2$ :

$$sh(xx_1, xx_2) = \begin{cases} 1 - \left( \frac{d(xx_1, xx_2)}{\sigma} \right), & d(xx_1, xx_2) \leq \sigma \\ 0, & otherwise \end{cases} \quad (4)$$

$d(xx_1, xx_2)$  is the Euclidean distance between  $x_1$  and  $x_2$ ,  $\sigma$  is the niche radius.

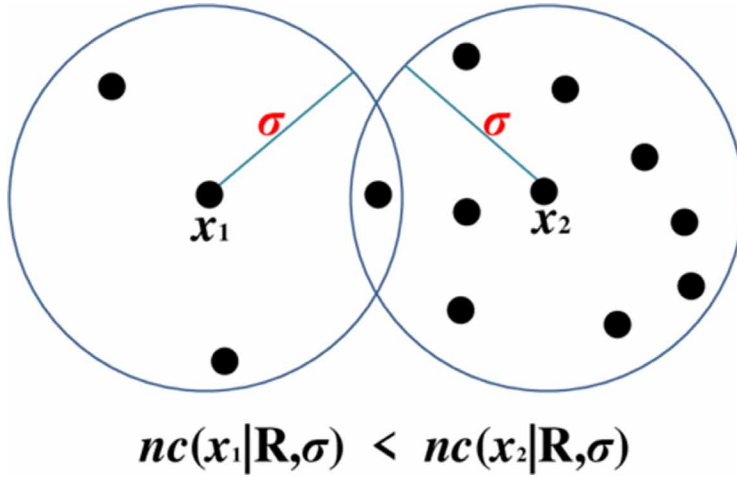
Figure 1 illustrates the relationship between two solutions' niche-count value. As shown in Figure 1, under the fixed niche radius  $\sigma$ , the niche-count value of solution  $x_1$  is smaller than solution  $x_2$ , which means solution  $x_1$  has the better diversity than solution  $x_2$ .

A **dynamic multi-objective optimization problem (DMOP)** is a sequence of MOPs  $\{MOP^{(s)}\}, s = 0, 1, \dots, S$ , defined as

**Definition 2: (DMOP)**

$$\begin{aligned} MOP^{(0)} : & \min_y y = (f(xx), nc(xx | \mathbf{Q}, \sigma^{(0)})) \\ MOP^{(1)} : & \min_y y = (f(xx), nc(xx | \mathbf{Q}, \sigma^{(1)})) \\ & \dots\dots & \dots\dots \\ MOP^{(s)} : & \min_y y = (f(xx), nc(xx | \mathbf{Q}, \sigma^{(s)})) \\ & \text{where } xx \in \mathbf{X} = \{xx | ll \leq xx \leq uu\}, \end{aligned} \quad (5)$$

Figure 1. The niche-count of  $x_1$  is smaller than  $x_2$



where  $\sigma^{(0)} > \sigma^{(1)} > \dots > \sigma^{(s)} = 0$ .  $S$  is the maximum environmental state which is also equivalent to the maximum generation.  $\sigma^{(s)}$  denotes the niche radius value at state  $s$ . An environment change can be seen as the reduction of the niche radius  $\sigma$  from state  $s$  to  $s + 1$ .

The initial niche radius  $\sigma^{(0)}$  is calculated by  $\sigma^{(0)} = \frac{1}{2} \sqrt{\frac{2D \prod_{i=1}^D (u_i - l_i)}{2NP\pi}}$ , which regards the average space occupied by each solution in the initial population. The final niche radius  $\sigma^{(s)} = 0$ , which guarantees the population to converge to the optimum. The reduction of the niche radius  $\sigma$  over time  $s$  adopts the following formula:

$$\sigma^{(s)} = Ce^{-\frac{s}{D}} - \varepsilon \quad (6)$$

where  $\varepsilon$  is a given positive close-to-zero number ( $\varepsilon = 1e-8$ ).  $C$  and  $D$  can be obtained in line with the initial niche radius  $\sigma^{(0)}$  and the last state niche radius  $\sigma^{(s)} = 0$ .

As the niche radius  $\sigma$  decreases from the initial radius  $\sigma^{(0)}$  to the final radius  $\sigma^{(s)} = 0$ , the DMOPs change gradually from the initial  $MOP^{(0)}$  to the final  $MOP^{(s)}$ :

$$MOP^{(s)} : \min \quad yy = (f(xx), 0). \quad (7)$$

It is obvious that the Pareto optimal solutions of  $MOP^{(s)}$  in Eq. (7) is an one-solution set due to the second objective is equal to 0, which equivalents to the original optimum of the global optimization problem in Eq. (1). So we can say the global optimization problem is transformed into the DMOP is an equal conversion.

## 5. THE PROPOSED DMOEA/D-M2M

### 5.1. Algorithm Framework

This section presents the structure of the proposed algorithm, named DMOEA/D-M2M, which relies on a state-of-the-art decomposition-based MOEA: MOEA/D-M2M. The pseudo code of the proposed DMOEA/D-M2M algorithm is presented in Algorithm 1. DMOEA/D-M2M starts with a randomly generated population with  $K*NS$  individuals, where  $K$  is the subpopulation number and  $NS$  is the subpopulation size. In terms of the  $D$ -dimensional hypersphere volume formula, the mean space occupied by each individual in the decision space is regarded as the initial niche radius  $\sigma^{(0)}$ . By regarding the niche count function as a helper objective, the initial population can be divided into  $K$  subpopulations according to objective values.

---

#### Algorithm 1: DMOEA/D-M2M

---

**Input:**  $K$ : the number of subproblems;  $K$  unit direction vectors:  $v^1, \dots, v^K$ ;  $K$  unit direction vectors:  $v^1, \dots, v^K$ ;

**Output:** The nondominated solutions.

- 1 Uniformly and randomly generate  $NP$  ( $K \times NS$ ) solutions;
  - 2 Initializing the niche radius  $\sigma = \sigma^{(0)}$ ,  $MOP^{(0)}$ , environmental state  $s = 0$ ;
  - 3 Computing the objective values of the initial population and then use them to set  $P_1, \dots, P_K$ ;
  - 4 **while** the halting criterion is not met **do**
  - 5     Reducing  $\sigma = \sigma^{(s+1)}$  according to Eq. (6);
  - 6     Updating  $MOP^{(s+1)}$  according to Eq. (5);
  - 7     Set  $R = \emptyset$ ;
  - 8     **for**  $k = 1$  **to**  $K$  **do**
  - 9         Using DE to generate new subpopulation  $P'_k$  by  $P_k$ ;
  - 10          $R = R \cup P'_k$ ;
  - 11     **end**
  - 12      $Q = R \cup (\bigcup_{k=1}^K P_k)$ ;
  - 13     Using  $Q$  to allocate subpopulations  $P_1, \dots, P_K$ ;
  - 14      $s = s + 1$ ;
  - 15 **end**
  - 16 return nondominated solutions.
- 

### 5.2. Generation of Subpopulation

At each generation, the niche radius is dynamically decreased over time in terms of Eq. (6). DMOEA/D-M2M maintains  $K$  subpopulations:  $P_1, \dots, P_K$ . Each subpopulation corresponds to a subproblem, and involves  $NS$  individuals. In this paper, DE/rand/1/bin operator (Storn & Price, 1997) is employed to generate  $NS$  offspring individuals for each subpopulation. The details of offspring reproduction are shown in Algorithm 2.

---

**Algorithm 2:** Generate new subpopulation by DE

---

**Input:** Parent subpopulation  $P_k = \{\mathbf{x}_1, \dots, \mathbf{x}_{NS}\}$ .

**Output:** Offspring subpopulation  $P'_k = \{\mathbf{x}'_1, \dots, \mathbf{x}'_{NS}\}$ .

```

1 for  $i = 1$  to  $NS$  do
2     Select randomly  $\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c$  from  $P_k$ ,  $\mathbf{x}_a \neq \mathbf{x}_i, \mathbf{x}_b \neq \mathbf{x}_i, \mathbf{x}_c \neq \mathbf{x}_i$ ;
3     Performing mutation on  $\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c$ :
4          $\mathbf{v}_i = \mathbf{x}_a + F \times (\mathbf{x}_b - \mathbf{x}_c)$ ;
5     Performing crossover on  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  and  $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ :
6          $j_{rnd} = rndInt(1, D), r_{rnd} = rndReal(0, 1)$ ,
7          $x'_{ij} = \begin{cases} v_{ij}, & \text{if } r_{rnd} < CR \text{ or } j = j_{rnd}, \\ x_{ij}, & \text{if } r_{rnd} \geq CR \text{ and } j \neq j_{rnd}, j = 1, \dots, D. \end{cases}$ 
8 end
9 return offspring subpopulation  $P'_k$ .
```

---

### 5.3. Allocation of Solutions to Each Subpopulation

After generating  $K$  offspring subpopulations, we need to select elite individuals from the union of the parent and offspring populations as the parent subpopulations of next generation. To this end, first, we combine all parent subpopulations and offspring subpopulations to a solution set  $Q$ . Apparently, the solution set  $Q$  has  $2NP$  individuals. For  $k$ th subpopulation, we identify solutions whose objective values are in the  $k$ th subspace based on the bi-objective  $(f(x), nc(xx | \mathbf{Q}, \sigma^{(s)}))$ .

Algorithm 3 guarantees that each subpopulation  $P_k$  has  $NS$  individuals at each generation and, hence, enhances the population diversity during the search (Liu, Gu, & Zhang, 2014). To be specially, if the number of individuals of the  $k$ th subpopulation is less than the subpopulation size  $NS$ , then, we randomly choose  $NS - |P_k|$  solutions from the entire solution set  $Q$ ; otherwise, we choose  $NS$  elite solutions from subspace  $k$  using nondominated sorting (Deb, Pratap, & Agarwal, 2002) and add them to subpopulation  $P_k$ .

---

**Algorithm 3:** Allocation of solutions to each subpopulation

---

**Input:** A set of solutions in  $Q$ .

**Output:** Subpopulations  $P_1, \dots, P_K$ .

```

1 for  $k = 1$  to  $K$  do
2     Initialize  $P_k$  as the solutions in  $Q$  whose objective values are in its subspace;
3     if  $|P_k| \leq NS$  then
4         randomly select  $NS - |P_k|$  solutions from  $Q$  and add them to  $P_k$ ;
5     else
6         Choose  $NS$  solutions from  $P_k$  using nondominated sorting;
7     end
8 end
9 return sub-populations  $P_1, \dots, P_K$ .
```

---

In line with the search of DMOEA/D-M2M, optimizing the original objective  $f(xx)$  is helpful to the population convergence, while minimizing the helper objective, the niche-count  $nc(xx | \mathbf{Q}, \sigma)$ , is beneficial to the population diversity from the decision space. Through the entire evolution process, the proposed DMOEA/D-M2M is capable of providing a sound tradeoff between the exploration and exploitation by dynamically decreasing the niche radius from a large value to zero, from the diverse population to the global optimum.

#### 5.4. Computational Complexity

The computational complexity of DMOEA/D-M2M is mainly governed by calculating the niche-count objective. The computational cost for niche-count is  $O(D*NP^2)$ , where  $D$  denotes the dimension of the solution vector and  $NP$  represents the population size.

### 6. EXPERIMENTAL RESULTS

#### 6.1. Benchmark Problems and Compared Algorithms

The performance of DMOEA/D-M2M is tested by the IEEE CEC2014 test suite (Liang, Qu, & Suganthan, 2013). IEEE CEC2014 test suite consists of thirty test problems. Among them, three of them are unimodal functions (F01-F03), thirteen of them are simple multimodal functions (F04-F16), and the remaining test problems are hybrid or composition functions (F17-F30). It exhibits various complex characteristics, such as strong nonlinearity, rotated landscape. Therefore, it can provide a systematic assessment on the performance of the proposed DMOEA/D-M2M.

Four state-of-the-art global optimization EAs are used to as competitors: MOMPSO (Singh & Deep, 2017), LX-BBO (Garg & Deep, 2016), MERDE (Qu, Liang, & Xiao, 2014), and M-PSO-MA (Singh, Deep, & Nagar). Among these four compared algorithms, MOMPSO and M-PSO-MA are based on particle swarm optimization (PSO), MERDE is based on DE, and LX-BBO is based on biogeography algorithm.

To detect the differences of different algorithms for statistical significance, the Wilcoxon rank sum test with a 0.05 significance level is performed between DMOEA/D-M2M and each competitor. The symbols +, -, and  $\approx$  suggest that the performance of the proposed DMOEA/D-M2M is significantly better than, significant worse than, and no significant difference to the corresponding algorithm, respectively. In addition, the Friedman test is chosen to sort all algorithms on all test functions.

#### 6.2. Parameter Settings

- The dimension of the test problems:  $D = 10$  and  $D = 30$ , respectively;
- The population size:  $NP = 100$ ;
- The subpopulation size and the subpopulation number:  $NS=10$ ,  $K=10$  (Liu, Gu, & Zhang, 2014);
- DE parameters:  $CR = 0.9$ ,  $F$  is randomly chosen in  $[0.0,1.0]$ ;
- The number of independent runs: 31;
- The number of function evolutions:  $FES=D*10,000$ .

#### 6.3. Comparisons with State-of-the-Art Algorithms on CEC 2014 Test Suite with $D=10$ and $D=30$

Tables 1 and 2 summarize the average objective function error values and standard deviations obtained from five compared algorithms over 31 independent runs on solving IEEE CEC 2014 test suite with  $10D$  and  $30D$ , respectively. The objective function error value is calculated by the absolute difference between obtained objective value at the termination of each algorithm and the known optimal value. In these tables, the best results of each test problem were highlighted in the gray background.



**Table 1. The average and standard deviation error value obtained by five algorithms for 10-dimensional IEEE CEC2014 benchmark problems. The best results in each test function are highlighted.**

Pro	MOMPSO	LX-BBO	MERDE	M-PSO-MA	DMOEA/D-M2M
F1	1.41e+04±1.77e+04	1.61e+03±1.14e+03	<b>1.58e+00±7.61e+00</b>	4.01e+01±3.17e+01	3.47e+02±7.14e+02
F2	8.43e+03±3.80e+03	5.80e+03±2.27e+03	<b>6.31e-05±1.12e-04</b>	3.17e-02±4.63e-02	6.06e+02±1.27e+03
F3	1.21e+04±1.18e-04	4.47e+03±5.52e+03	1.35e-03±1.22e-03	<b>0.00e+00±0.00e+00</b>	2.25e-07±6.31e-07
F4	6.18e+00±1.07e+01	1.72e+00±4.19e-03	<b>0.00e+00±0.00e+00</b>	6.18e+00±1.07e+01	1.74e+01±1.98e+01
F5	2.00e+01±3.81e-02	<b>1.01e+00±2.81e-01</b>	1.90e+01±2.71e+00	2.00e+01±5.56e-03	1.57e+01±7.80e+00
F6	3.53e+00±1.77e+00	3.45e+00±1.52e+00	8.93e-01±2.81e-01	1.22e+00±1.32e+00	<b>6.90e-02±1.21e-01</b>
F7	1.17e-01±6.19e-02	2.56e-01±1.40e-01	<b>1.83e-02±1.51e-02</b>	5.37e-02±2.81e-02	6.09e-02±2.62e-02
F8	1.05e+01±5.36e+00	<b>0.00e+00±0.00e+00</b>	<b>0.00e+00±0.00e+00</b>	3.12e-01±1.08e+00	0.00e+00±0.00e+00
F9	1.28e+01±8.22e+00	1.10e+01±4.27e+00	5.58e+00±1.74e+00	<b>5.08e+00±2.02e+00</b>	5.71e+00±2.33e+00
F10	3.14e+02±2.03e+02	9.01e+02±5.45e+02	<b>3.67e-02±3.98e-02</b>	9.68e+01±9.89e+01	2.57e-01±7.24e-02
F11	4.70e+02±2.54e+02	1.12e+03±5.93e+02	7.55e+01±7.63e+01	<b>2.52e+00±1.76e+02</b>	5.40e+01±8.40e+01
F12	1.90e-01±1.23e-01	1.00e-01±4.20e-17	1.17e-01±6.93e-02	<b>5.29e-02±3.61e-02</b>	1.07e-01±9.01e-02
F13	<b>7.31e-02±4.04e-02</b>	3.12e-01±1.50e-01	1.17e-01±4.43e-02	1.02e-01±4.80e-02	7.99e-02±1.97e-02
F14	2.20e-02±8.27e-03	2.39e-01±2.22e-01	9.37e-02±2.73e-02	<b>2.09e-02±7.87e-03</b>	7.13e-02±2.41e-02
F15	6.85e-01±1.93e-01	1.51e+00±7.88e-01	6.72e-01±2.18e-01	<b>6.46e-01±1.58e-01</b>	7.61e-01±2.28e-01
F16	2.69e+00±4.30e-01	2.37e+00±4.16e-01	1.53e+00±4.63e-01	1.47e+00±5.68e-01	<b>7.72e-01±2.83e-01</b>
F17	1.08e+03±4.01e+02	5.66e+03±6.81e+03	<b>7.93e+00±9.59e+00</b>	2.38e+02±1.37e+02	4.54e+01±1.08e+02
F18	7.82e+02±1.20e+03	7.02e+03±7.18e+03	2.72e+00±1.29e+00	4.63e+02±5.90e+02	<b>1.68e+00±9.29e-01</b>
F19	2.75e+00±1.43e+00	3.69e+00±7.27e+00	5.10e-01±1.76e-01	9.29e-01±6.98e-01	<b>1.42e-01±9.44e-02</b>
F20	3.95e+01±3.89e+01	1.61e+04±2.06e+04	1.70e+00±7.50e-01	2.50e+00±1.71e+00	<b>2.34e-01±4.46e-01</b>
F21	3.64e+02±2.63e+02	6.26e+03±7.23e+03	8.54e+00±2.66e+01	6.11e+01±7.63e+01	<b>3.13e-01±2.48e-01</b>
F22	5.35e+01±6.99e+01	7.59e+01±7.45e+01	3.24e+00±3.96e+00	1.50e+01±9.03e+00	<b>1.13e+00±3.24e+00</b>
F23	3.29e+02±1.38e-12	2.44e+02±5.65e+01	3.29e+02±2.68e-11	3.29e+02±1.38e-12	<b>2.18e+02±4.10e+01</b>
F24	1.20e+02±6.74e+00	1.01e+03±8.64e+02	1.15e+02±2.45e+00	<b>1.14e+02±4.30e+00</b>	1.17e+02±3.61e+00
F25	1.98e+02±1.60e+01	1.78e+02±1.50e+01	<b>1.36e+02±8.34e+00</b>	1.54e+02±4.17e+01	1.63e+02±1.79e+00
F26	1.00e+02±3.34e-02	<b>3.71e-03±6.58e-03</b>	1.00e+02±4.18e-02	1.00e+02±3.30e-02	1.05e+02±6.17e-05
F27	3.16e+02±1.71e+02	<b>1.05e+01±7.68e+00</b>	2.87e+01±7.60e+01	2.90e+02±1.57e+02	1.88e+02±4.33e+01
F28	4.73e+02±1.07e+02	5.29e+02±1.14e+02	3.66e+02±7.37e+00	4.52e+02±7.24e+01	<b>3.19e+02±3.70e-01</b>
F29	3.63e+02±9.23e+01	3.53e+05±7.54e+05	3.17e+02±5.48e+01	2.93e+02±4.57e+01	<b>2.05e+02±1.67e+00</b>
F30	7.02e+02±2.63e+02	6.31e+04±6.97e+04	5.34e+02±6.06e+01	6.09e+02±1.78e+02	<b>2.31e+02±4.93e+00</b>

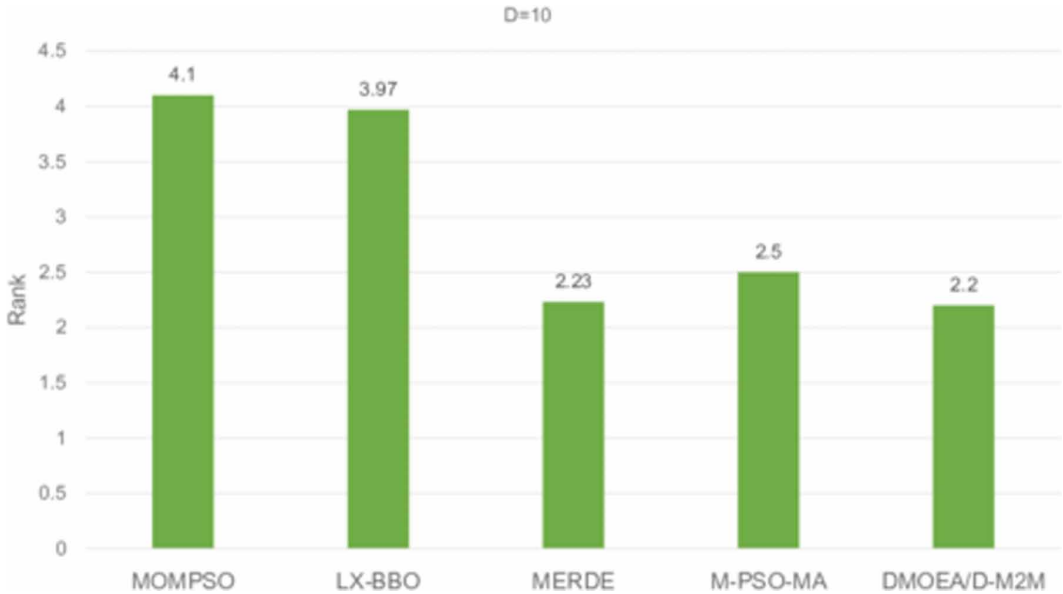
In the case of  $D=10$ , the experimental results are collected in Table 1. From Table 1, for the number of test problems with obtained best results, DMOEA/D-M2M performed best on 12 test problems, while the compared algorithms MOMPSO, LX-BBO, MERDE, and M-PSO-MA obtains 1, 4, 8, and 7 best results, respectively. As far as the multiple-problem Wilcoxon's signed rank test in Table 3, DMOEA/D-M2M provides higher R+ values than R- values in all cases. According to the Friedman's test in Figure 2, DMOEA/D-M2M has the best ranking among five compared algorithms.

**Table 2.** The average and standard deviation error value obtained by five algorithms for 30-dimensional IEEE CEC2014 benchmark problems. The best results in each test function are highlighted.

Pro	MOMPSO	LX-BBO	MERDE	M-PSO-MA	DMOEA/D-M2M
F1	1.98e+06±1.25e+06	1.01e+07±3.81e+06	<b>5.41e+02±6.40e+02</b>	3.09e+03±3.19e+03	5.62e+05±4.34e+05
F2	3.79e+02±1.88e+02	5.34e+04±2.14e+04	2.39e-03±3.24e-03	<b>0.00e+00±0.00e+00</b>	1.13e+07±1.03e+07
F3	1.20e+04±1.18e+04	1.64e+04±1.71e+04	1.13e-03±7.36e-04	<b>0.00e+00±0.00e+00</b>	1.03e+02±1.02e+02
F4	2.35e+02±1.11e+03	9.99e+01±2.85e+01	<b>6.25e-01±1.46e+00</b>	5.29e+00±1.77e+01	2.95e+01±4.78e+01
F5	2.06e+01±2.64e-01	<b>3.06e+00±7.87e-01</b>	2.00e+01±7.17e-05	2.00e+01±4.45e-02	2.01e+01±6.26e-02
F6	2.76e+01±5.99e+00	1.69e+01±3.12e+00	1.82e+01±1.72e+00	1.05e+01±2.36e+00	<b>7.39e+00±1.44e+00</b>
F7	1.28e-02±1.41e-02	1.76e-01±8.56e-02	<b>0.00e+00±0.00e+00</b>	1.22e-02±1.25e-02	5.94e-01±2.83e-01
F8	1.35e+02±3.15e+01	5.53e+01±3.78e+02	<b>0.00e+00±0.00e+00</b>	1.03e+00±1.23e+00	3.30e-04±1.49e-03
F9	1.64e+02±2.35e+01	7.66e+01±1.61e+01	5.50e+01±1.04e+01	7.63e+01±1.36e+01	<b>3.14e+01±8.83e+00</b>
F10	3.86e+03±5.74e+02	1.26e+04±1.16e+02	1.29e+00±1.61e+00	4.54e+02±2.58e+02	<b>3.19e-01±2.47e-01</b>
F11	3.81e+03±4.80e+02	1.23e+04±3.42e+02	2.72e+03±4.69e+02	2.59e+03±4.78e+02	<b>1.55e+03±5.06e+02</b>
F12	1.35e+00±6.69e-01	<b>1.11e-02±1.75e-18</b>	5.64e-01±1.74e-01	8.06e-02±2.86e-02	1.12e-01±4.40e-02
F13	4.34e-01±1.01e-01	6.55e-01±1.56e-01	<b>2.84e-01±4.41e-02</b>	3.29e-01±7.47e-02	3.23e-01±6.47e-02
F14	6.97e-01±2.19e-01	6.20e-01±2.96e-01	2.14e-01±2.69e-02	2.16e-01±7.88e-02	<b>1.77e-01±3.83e-02</b>
F15	1.03e+01±3.24e+00	1.55e+01±5.50e+00	<b>4.14e+00±7.77e-01</b>	9.90e+00±3.27e+00	4.95e+00±1.53e+00
F16	1.18e+01±5.90e-01	1.08e+01±5.84e-01	1.12e+01±4.58e-01	1.03e+01±4.90e-01	<b>7.70e+00±9.70e-01</b>
F17	1.13e+05±3.34e+04	1.46e+06±9.34e+05	<b>1.16e+03±3.73e+02</b>	1.75e+03±5.32e+02	1.55e+04±2.66e+04
F18	2.06e+04±7.01e+03	2.90e+03±4.27e+03	<b>2.24e+01±6.47e+00</b>	2.14e+03±1.83e+03	1.90e+03±3.32e+03
F19	3.15e+01±3.22e+01	5.19e+03±5.67e+03	7.74e+00±7.32e-01	<b>7.35e+00±1.46e+00</b>	9.21e+00±2.83e+00
F20	5.45e+02±4.38e+02	2.61e+04±1.56e+04	<b>2.80e+01±1.06e+01</b>	5.56e+01±2.79e+01	1.82e+02±1.27e+02
F21	5.16e+04±2.89e+04	1.11e+06±7.95e+05	<b>5.99e+02±2.15e+02</b>	1.89e+03±1.57e+03	6.70e+03±8.80e+03
F22	5.39e+02±1.75e+02	1.88e+03±2.04e+02	1.15e+02±7.29e+01	2.80e+02±1.23e+02	<b>5.45e+01±6.35e+01</b>
F23	3.39e+02±4.31e+01	4.11e+02±6.43e+01	3.14e+02±3.74e-09	3.15e+02±1.39e-12	<b>2.89e+02±1.08e+01</b>
F24	2.26e+02±1.41e+01	1.48e+04±8.37e+03	2.25e+02±5.57e-01	<b>2.24e+02±5.34e+00</b>	2.25e+02±1.60e+00
F25	2.06e+02±1.87e+00	5.29e+02±4.37e+01	<b>2.00e+02±2.83e-02</b>	2.04e+02±1.56e+00	2.03e+02±3.82e+00
F26	1.05e+02±4.67e+00	2.13e+00±3.46e+00	1.00e+02±7.61e-02	<b>1.00e+00±6.42e-02</b>	1.01e+02±7.14e-02
F27	1.10e+03±2.41e+02	<b>1.96e+02±1.04e+02</b>	3.84e+02±2.32e+01	6.70e+02±1.56e+02	7.16e+02±3.44e+01
F28	1.50e+03±2.03e+02	1.94e+03±5.49e+02	8.05e+02±2.49e+01	1.13e+03±1.67e+02	<b>3.88e+02±9.82e+00</b>
F29	3.61e+03±1.34e+03	1.98e+07±3.96e+06	1.19e+03±1.06e+02	1.73e+03±4.62e+02	<b>9.25e+01±7.12e+01</b>
F30	2.42e+04±2.46e+04	6.96e+06±1.03e+07	1.07e+03±2.90e+02	2.52e+03±1.24e+03	<b>3.80e+02±1.24e+02</b>

In the case of  $D=30$ , the experimental results are shown in Table 2. We can see that DMOEA/D-M2M has the best performance on 11 test instances, while its competitors MOMPSO, LX-BBO, MERDE, and M-PSO-MA can achieve the best results among five algorithms on 0, 3, 10, and 5 test problems, which all less than the number of test problems with best results obtained by DMOEA/D-M2M. In terms of the multiple-problem Wilcoxon's signed rank test, DMOEA/D-M2M provides higher R+ values than R- values in all cases, and performs significantly better than MOMPSO and

Figure 2. Ranking of 5 algorithms for CEC2014 (D=10) by Friedman's test



LX-BBO as the  $p$ -value are less than 0.05. According to the Friedman's test in Figure 3, DMOEA/D-M2M ranks the second among five compared algorithms.

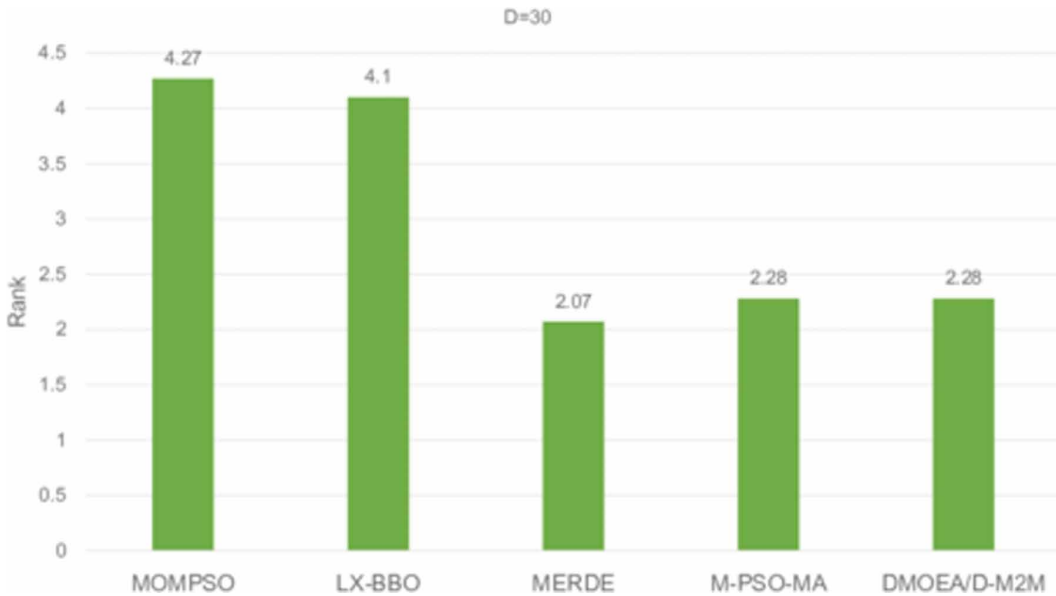
For unimodal test functions, DMOEA/D-M2M has poor performance than its competitors, this can be attributed to the fact that DMOEA/D-M2M puts more emphasis on investigating more regions in the search space to discover more potential solutions in the early stage, which results in the slower convergence speed than its competitors on unimodal test problems. Nevertheless, on complicated hybrid or composition problems, DMOEA/D-M2M outperforms the compared algorithms. These test instances involve multi-modality and strong nonlinearity, so an algorithm is easy to fall into some local optima. For DMOEA/D-M2M, a larger niche radius at the early stage of the evolution could maintain the proper population diversity and makes DMOEA/D-M2M more powerful in global exploration, that is why it has better performance on these problems.

Overall, the above comparison results suggest that DMOEA/D-M2M is better than or highly competitive to the compared four algorithms on solving IEEE CEC 2014 test suite with  $D=10$  and  $D=30$ .

Table 3. Statistical test results of DMOEA/D-M2M and four peer algorithms by the Multiple-Problem Wilcoxon's test for IEEE CEC2014.

DMOEA/D-M2M vs	D=10			D=30		
	R+	R-	?=0.05	R+	R-	?=0.05
MOMPSO	433.0	32.0	+	431.0	34.0	+
LX-BBO	381.0	54.0	+	397.0	68.0	+
MERDE	233.0	202.0	≈	184.0	251.0	≈
M-PSO-MA	302.0	163.0	≈	241.5	223.5	≈

Figure 3. Ranking of 5 algorithms for CEC2014 (D=30) by Friedman's test



## 7. FURTHER DISCUSSION: EXPLORATION AND EXPLOITATION INVESTIGATION

Intuitively, the more objectives a problem has, the more complicated is. To answer why the proposed bi-objective method can be superior to the single objective method, in this section, an another experiment was performed to investigate the exploration and exploitation ability of the proposed DMOEA/D-M2M and DE algorithm, since DMOEA/D-M2M also employs DE to generate offspring subpopulations. The major difference between DE and DMOEA/D-M2M is: the proposed DMOEA/D-M2M takes the niche count function as a helper objective, and uses the decomposition-based MOEA to solve the transformed bi-objective optimization problem. We use two highly multi-modal functions as test problems. The first one is Schwefel function:

$$\min f(\vec{x}) = 418.9829D - \sum_{j=1}^D x_j \sin\left(\sqrt{|x_j|}\right) \quad (8)$$

where  $-500 \leq x_j \leq 500, j = 1, \dots, D$ .

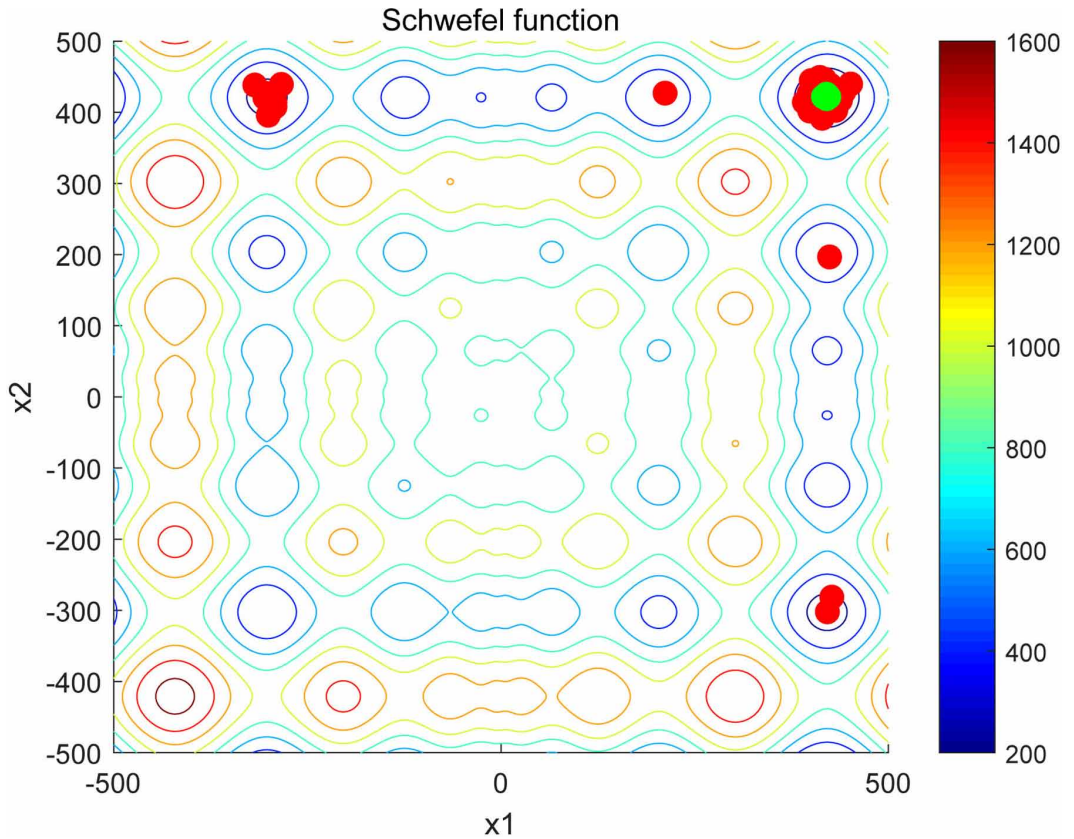
The second test problem is Rastrigin function:

$$\min f(\vec{x}) = 10D + \sum_{i=1}^D (x_i^2 - 10\cos(2\pi x_i)) \quad (9)$$

where  $-5.12 \leq x_j \leq 5.12, j = 1, \dots, D$ .

Schwefel function and Rastrigin function are two typical examples of non-linear multimodal function. Searching the optimum of these two problems are fairly tough due to its large search space and its large number of local optima. When solving Schwefel and Rastrigin problems by DMOEA/D-

Figure 4a. Population distribution on Schwefel function. The green points and red points represent the obtained population by DE and DMOEA/D-M2M, respectively. (Left: generation=30; Center: generation=60; Right: generation=100)

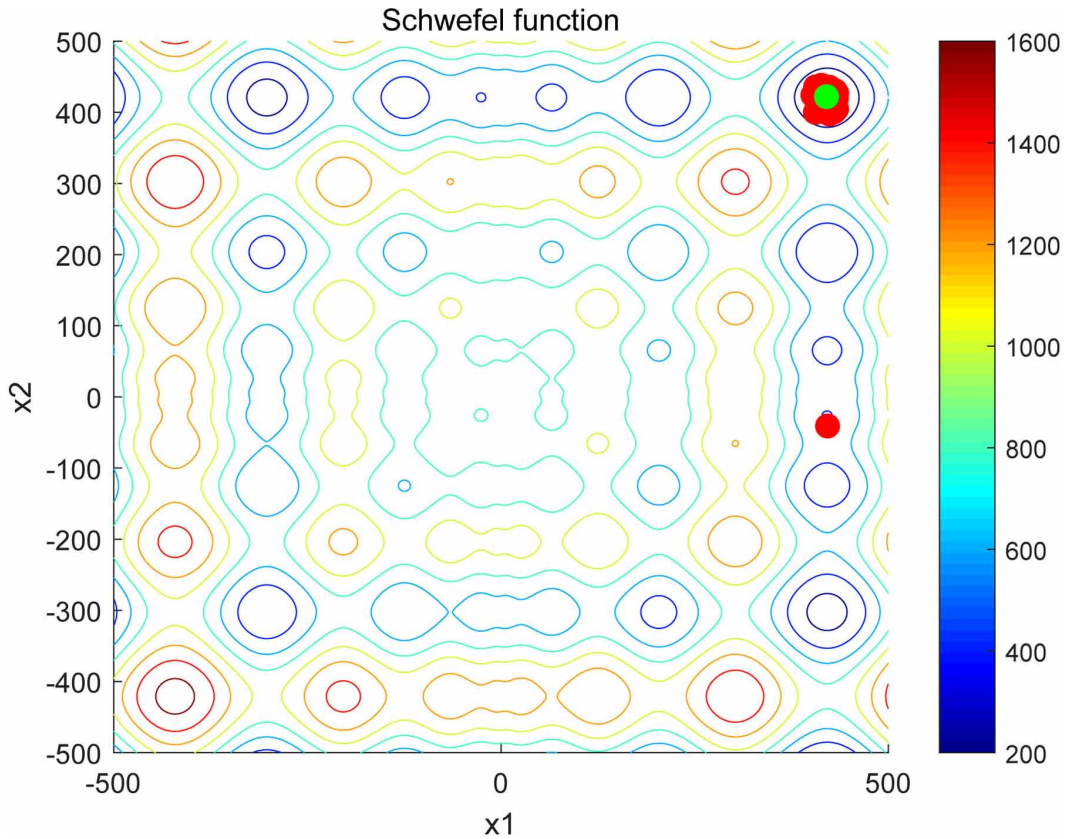


M2M and DE, the maximum number of function evaluations and the population size were set to 20000 and 100, respectively.

First, we plot the population distribution of DE and DMOEA/D-M2M in different generations on these two problems. Figure 4 plots the distribution of individuals in the decision space on Schwefel problem. The green points and red points represent individuals obtained by DE and DMOEA/D-M2M, respectively. From Figure 4, we can see in the 30th generation, the population of DE has converged to a small area, while the proposed DMOEA/D-M2M can cover five peaks, which suggests its good exploration ability. In the 60th generation, due to the niche radius has decreased, the population coverage range of DMOEA/D-M2M has shrunk, which covers two promising peaks. In the 100th generation, as expected, the population of DMOEA/D-M2M converged to one area where the global optimum located.

Figure 5 plots the population distribution of DE and DMOEA/D-M2M on Rastrigin function. The same as Figure 4 on Schwefel function, similar phenomenon can be observed from Figure 5. In the 30th generation, individuals of DMOEA/D-M2M distributed in ten different peaks, while DE can only locate three peaks. In the 60th generation, the population of DE has approached to a small area, which means it has lost its diversity. By contrast, our proposed DMOEA/D-M2M is still capable of covering seven peaks. This can be attributed to the helper objective niche count function can give some solutions with better diversity to survive during the environment selection procedure.

Figure 4b. Population distribution on Schwefel function. The green points and red points represent the obtained population by DE and DMOEA/D-M2M, respectively. (Left: generation=30; Center: generation=60; Right: generation=100)



Then, we employ the changes of the population diversity to investigate the exploration and exploitation ability, which is measured by the standard deviation of the population distribution (Yang, Chen, & Da, 2018):

$$D(XX) = \frac{1}{NP} \sum_{i=1}^{NP} \sqrt{\sum_{j=1}^D (x_i^j - \bar{x}^j)^2}, x_i \in XX \quad (10)$$

where  $\bar{x}$  represents the mean position of the population:

$$\bar{x}^j = \frac{1}{NP} \sum_{i=1}^{NP} x_i^j. \quad (11)$$

It is the fact that a large value of population diversity  $D(XX)$  means individuals of the population distribute relatively sparse, while a small value of population diversity  $D(XX)$  indicates individuals of the population distribute more crowded. Apparently, a good global optimization algorithm should keep a larger value of  $D(XX)$  at the early stage of evolution, with the purpose of investigating



undiscovered regions in the search space to find more potential solutions, while hold a smaller value of  $D(XX)$  at the later stage of evolution to improve the convergence.

Figure 6 illustrates the changes of the population diversity of DE and DMOEA/D-M2M on these two functions over time. We can see that the left and right Figure 6 have a similar variation tendency. For DE, the population diversity keeps unchanged since 60th generation, this can be explained by the middle subfigures of Figure 4 and Figure 5 that all individuals of DE have converged to a small area both on Schwefel and Rastrigin functions. By contrast, the proposed DMOEA/D-M2M has the capability of investigating more areas than DE in the early stage evolution, while it also can provide better exploitation in the later stage of evolution.

To sum up, the above experimental results validate that the proposed DMOEA/D-M2M could provide a sound balance between exploration and exploitation during the evolution.

Figure 4c. Population distribution on Schwefel function. The green points and red points represent the obtained population by DE and DMOEA/D-M2M, respectively. (Left: generation=30; Center: generation=60; Right: generation=100)

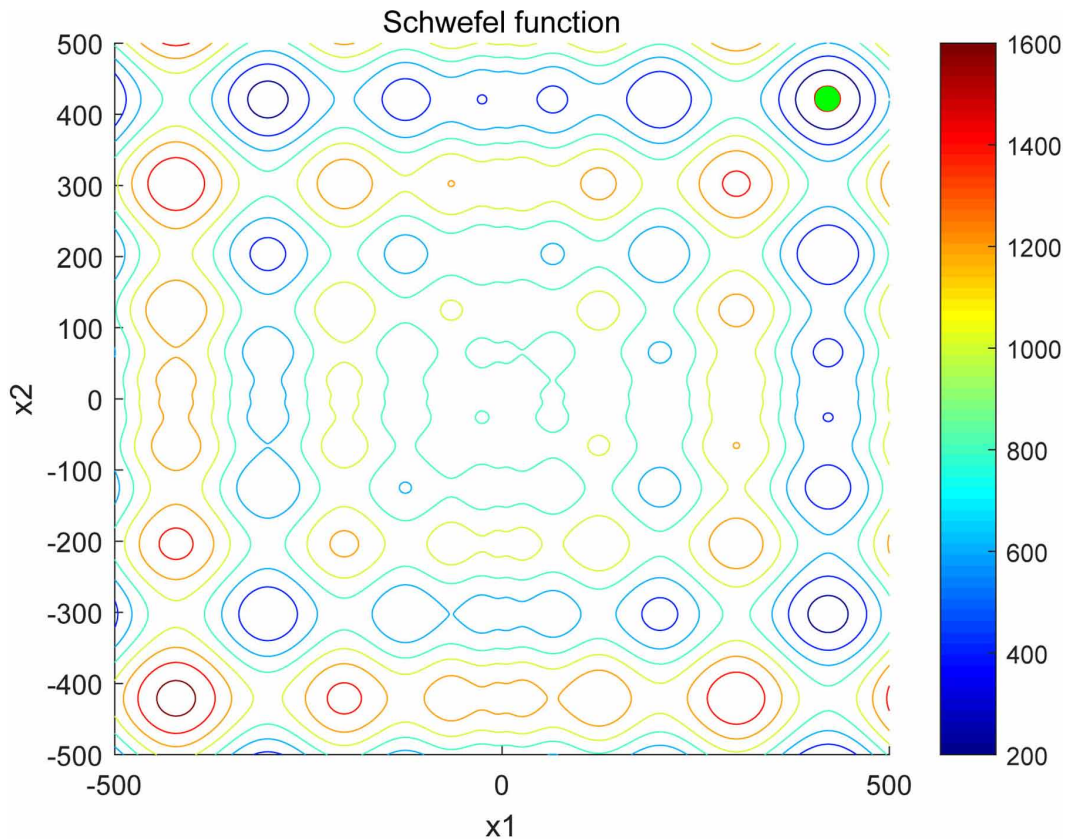


Figure 5a. Population distribution on Rastrigin function. The green points and red points represent the obtained population by DE and DMOEA/D-M2M, respectively. (Left: generation=30; Center: generation=60; Right: generation=100)

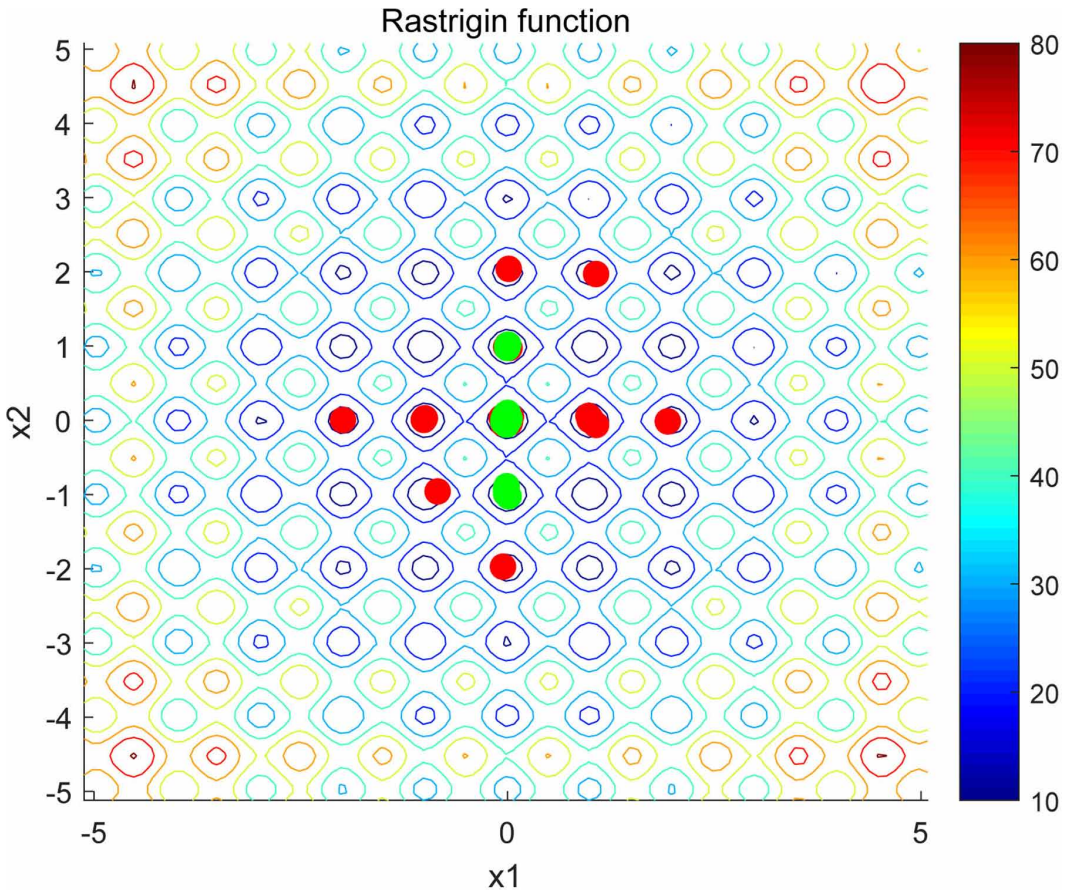




Figure 5b. Population distribution on Rastrigin function. The green points and red points represent the obtained population by DE and DMOEA/D-M2M, respectively. (Left: generation=30; Center: generation=60; Right: generation=100)

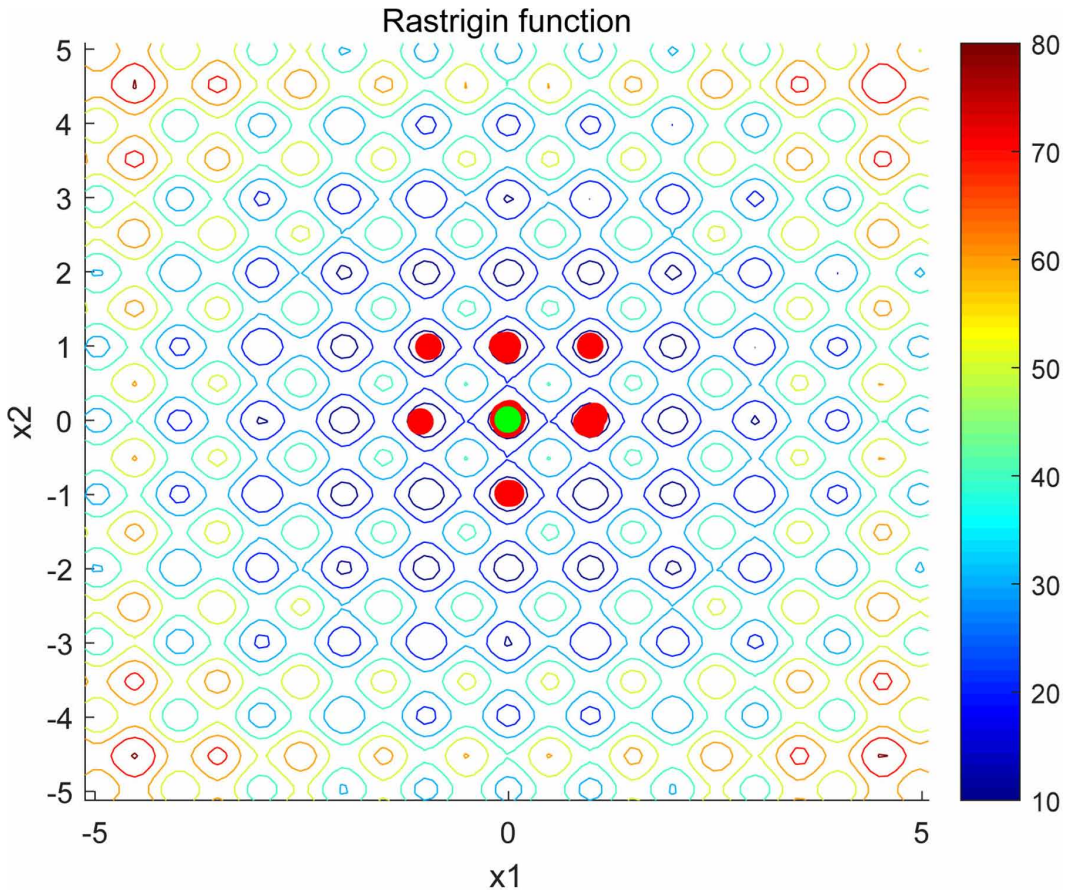


Figure 5c. Population distribution on Rastrigin function. The green points and red points represent the obtained population by DE and DMOEA/D-M2M, respectively. (Left: generation=30; Center: generation=60; Right: generation=100)

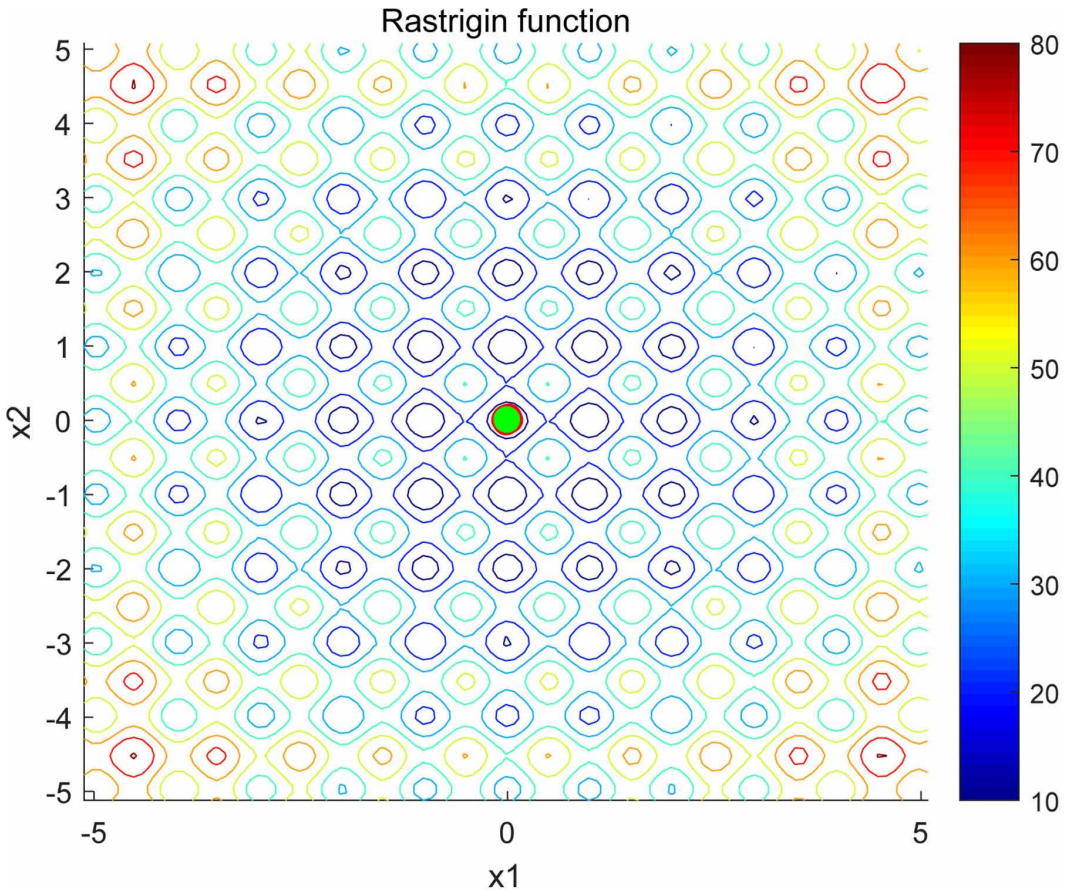


Figure 6a. The changes of the population diversity over time. (Left: Schwefel function; Right: Rastrigin function)

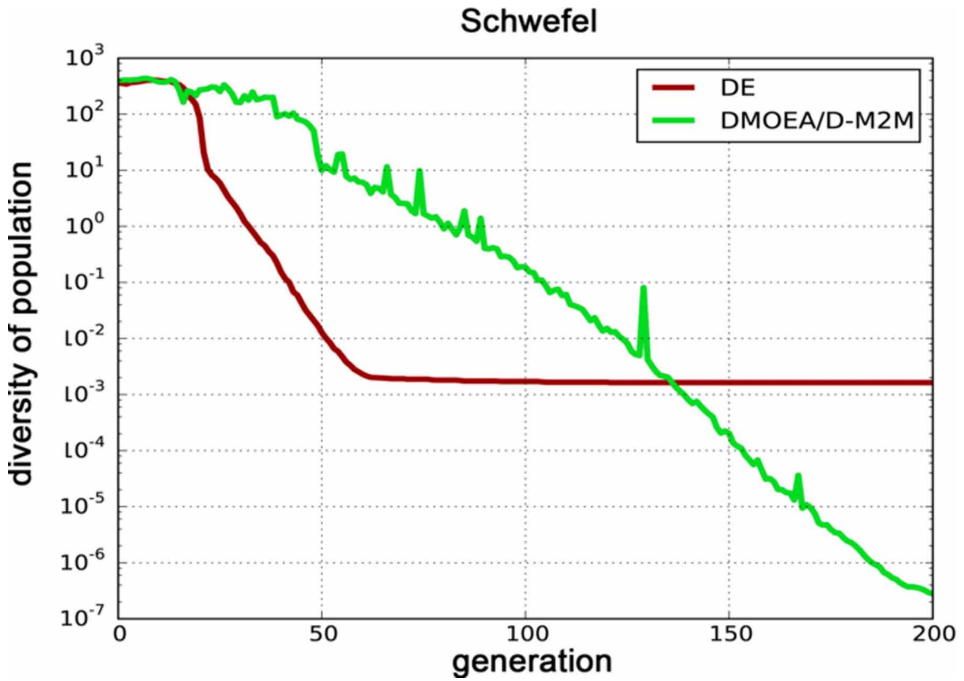
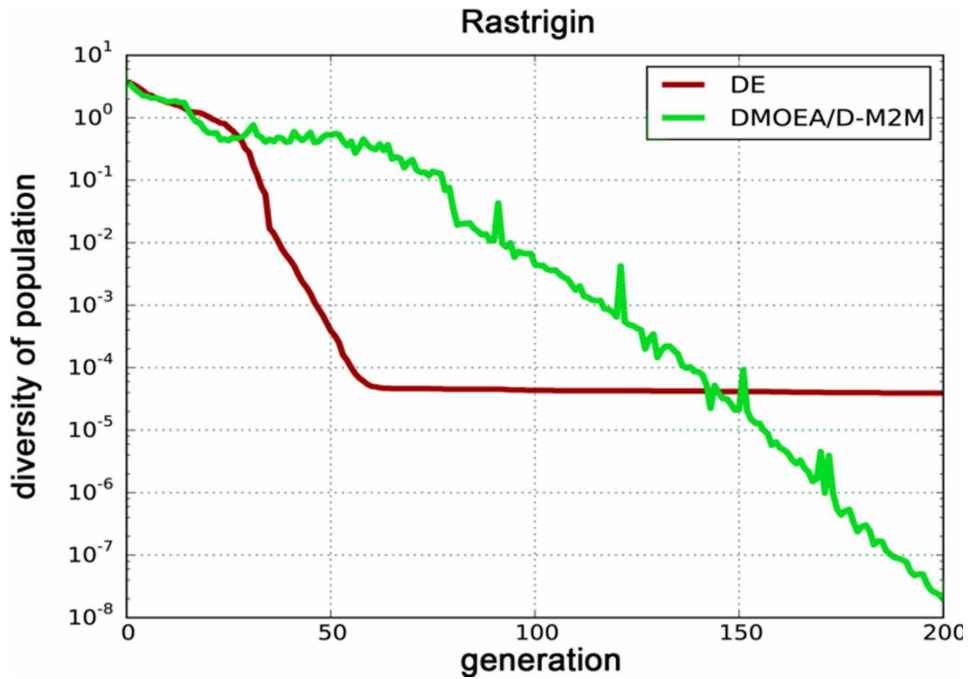


Figure 6b. The changes of the population diversity over time. (Left: Schwefel function; Right: Rastrigin function)



## 8. CONCLUSION AND FUTURE WORK

In this paper, a decomposition-based dynamic bi-objective evolutionary algorithm integrated into the framework of MOEA/D-M2M, called DMOEA/D-M2M, is proposed to deal with complex global optimization problems. DMOEA/D-M2M transforms a global optimization problem into an equivalent dynamic bi-objective optimization problem, one is the original objective and the another is the helper objective: niche count function. At each generation, the proposed DMOEA/D-M2M decomposes the transformed bi-objective optimization problem into a number of bi-objective optimization subproblems which are easy to solve, and these simple bi-objective subproblems are handled in a collaborative way. The added helper objective niche-count is controlled by the niche radius. In this paper, the niche radius is dynamically decreased over time, which can provide a tradeoff between exploration and exploitation.

To demonstrate the strong competitiveness, experimental comparison has conducted on solving IEEE CEC 2014 and compared DMOEA/D-M2M with four state-of-the-art algorithms. The results reveal that the proposed DMOEA/D-M2M is highly competitive for solving global optimization problems, particularly on multimodal problems.

In the future, we will give a deeper insight into the search behavior of DMOEA/D-M2M, so that the strength and weakness of DMOEA/D-M2M can be better understood. In addition, the proposed method can be extended to solve complicated multi-objective optimization problems. Furthermore, some state-of-the-art MOEAs can be regarded as base algorithms to replace MOEA/D-M2M that adopted in this paper. At last, it is interesting to adopt the proposed DMOEA/D-M2M to deal with real-world antenna design optimization problems (Jiao, Sun, & Sun, 2018).

## REFERENCES

- Bader, J., & Zitzler, E. (2011). Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1), 45–76. doi:10.1162/EVCO\_a\_00009 PMID:20649424
- Chen, L., Liu, H.-L., Tan, K. C., Cheung, Y.-M., & Wang, Y. (2019). Evolutionary many-objective algorithm using decomposition-based dominance relationship. *IEEE Transactions on Cybernetics*, 49(12), 4129–4139. doi:10.1109/TCYB.2018.2859171 PMID:30207973
- Chen, Z., Zhou, Y., Zhao, X., Xiang, Y., & Wang, J. (2018). A historical solutions based evolution operator for decomposition-based many-objective optimization. *Swarm and Evolutionary Computation*, 41, 167–189. doi:10.1016/j.swevo.2018.02.008
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. doi:10.1109/4235.996017
- Del Ser, J., Osaba, E., Molina, D., Yang, X., Salcedo-Sanz, S., Camacho, D., Das, S., Suganthan, P., Coello, C., & Herrera, F. (2019). Bio-inspired computation: Where we stand and what's next. *Swarm and Evolutionary Computation*, 48, 220–250. doi:10.1016/j.swevo.2019.04.008
- Garg, V., & Deep, K. (2016). Performance of laplacian biogeography-based optimization algorithm on cec 2014 continuous optimization benchmarks and camera calibration problem. *Swarm and Evolutionary Computation*, 27, 132–144. doi:10.1016/j.swevo.2015.10.006
- Gee, S. B., Arokiasami, W. A., Jiang, J., & Tan, K. C. (2016). Decomposition-based multi-objective evolutionary algorithm for vehicle routing problem with stochastic demands. *Soft Computing*, 20(9), 3443–3453. doi:10.1007/s00500-015-1830-2
- Ishibuchi, H., Tsukamoto, N., & Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. In *IEEE Congress on Evolutionary Computation*, (pp. 2419-2426). IEEE.
- Jiao, R., Sun, Y., Sun, J., Jiang, Y., & Zeng, S. (2018). Antenna design using dynamic multi-objective evolutionary algorithm. *IET Microwaves, Antennas & Propagation*, 12(13), 2065–2072. doi:10.1049/iet-map.2018.5298
- Jiao, R., Zeng, S., Alkassabeh, J., & Li, C. (2017). Dynamic multi-objective evolutionary algorithms for single-objective optimization. *Applied Soft Computing*, 61, 793–805. doi:10.1016/j.asoc.2017.08.030
- Jiao, R., Zeng, S., & Li, C. (2019). A feasible-ratio control technique for constrained optimization. *Information Sciences*, 502, 201–217. doi:10.1016/j.ins.2019.06.030
- Li, H., Deb, K., Zhang, Q., Suganthan, P., & Chen, L. (2019). Comparison between moea/d and nsga-iii on a set of novel many and multi-objective benchmark problems with challenging difficulties. *Swarm and Evolutionary Computation*, 46, 104–117. doi:10.1016/j.swevo.2019.02.003
- Li, K., & Zhang, Q. (2019). Decomposition multi-objective optimisation: Current developments and future opportunities. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, (pp. 1002-1031). ACM. doi:10.1145/3319619.3323369
- Liang, J., Qu, B., & Suganthan, P. (2013). Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. Technical Report, 1-32.
- Liu, H., Gu, F., & Zhang, Q. (2014). Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation*, 18(3), 450–455. doi:10.1109/TEVC.2013.2281533
- Liu, Z.-Z., Wang, Y., & Huang, P.-Q. (2020). And: A manyobjective evolutionary algorithm with angle-based selection and shift-based density estimation. *Information Sciences*, 509, 400–419. doi:10.1016/j.ins.2018.06.063
- Qu, B., Liang, J., Xiao, J., & Shang, Z. (2014). Memetic differential evolution based on fitness euclidean-distance ratio. In *IEEE Congress on Evolutionary Computation*, (pp. 2266-2273). IEEE. doi:10.1109/CEC.2014.6900476
- Segura, C., Coello, C. A. C., Miranda, G., & Le'on, C. (2016). Using multi-objective evolutionary algorithms for single-objective constrained and unconstrained optimization. *Annals of Operations Research*, 240(1), 217–250. doi:10.1007/s10479-015-2017-z

- Singh, A., & Deep, K. (2019). Exploration-exploitation balance in artificial bee colony algorithm: A critical analysis. *Soft Computing*, 23(19), 9525–9536. doi:10.1007/s00500-018-3515-0
- Singh, G., & Deep, K. (2017). Effectiveness of new multiplepso based membrane optimization algorithms on cec 2014 benchmarks and iris classification. *Natural Computing*, 16(3), 473–496. doi:10.1007/s11047-016-9573-2
- Singh, G., Deep, K., & Nagar, A. (2014). Cell-like p-systems based on rules of particle swarm optimization. *Applied Mathematics and Computation*, 246, 546–560. doi:10.1016/j.amc.2014.08.027
- Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341–359. doi:10.1023/A:1008202821328
- Trivedi, A., Srinivasan, D., Sanyal, K., & Ghosh, A. (2017). A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation*, 21(3), 440–462.
- Yang, Q., Chen, W., Da Deng, J., Li, Y., Gu, T., & Zhang, J. (2018). A level-based learning swarm optimizer for large-scale optimization. *IEEE Transactions on Evolutionary Computation*, 22(4), 578–594. doi:10.1109/TEVC.2017.2743016
- Yuan, Y., Xu, H., Wang, B., Zhang, B., & Yao, X. (2015). Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Transactions on Evolutionary Computation*, 20(2), 180–198. doi:10.1109/TEVC.2015.2443001
- Zeng, S., Jiao, R., Li, C., Li, X., & Alkasassbeh, J. (2017). A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization. *IEEE Transactions on Cybernetics*, 47(9), 2678–2688. doi:10.1109/TCYB.2017.2647742 PMID:28092596
- Zeng, S., Jiao, R., Li, C., & Wang, R. (2019). Constrained optimisation by solving equivalent dynamic loosely-constrained multiobjective optimization problem. *International Journal of Bio-inspired Computation*, 13(2), 86–101. doi:10.1504/IJBIC.2019.098406
- Zhang, Q., & Li, H. (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712–731. doi:10.1109/TEVC.2007.892759
- Zhang, S. X., Zheng, L. M., Liu, L., Zheng, S. Y., & Pan, Y. M. (2017). Decomposition-based multi-objective evolutionary algorithm with mating neighborhood sizes and reproduction operators adaptation. *Soft Computing*, 21(21), 6381–6392. doi:10.1007/s00500-016-2196-9
- Zhang, Y.-H., Gong, Y.-J., Gu, T.-L., Yuan, H.-Q., Zhang, W., Kwong, S., & Zhang, J. (2019). Decal: Decomposition-based coevolutionary algorithm for many-objective optimization. *IEEE Transactions on Cybernetics*, 49(1), 27–41. doi:10.1109/TCYB.2017.2762701 PMID:29990116
- Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1), 32–49. doi:10.1016/j.swevo.2011.03.001
- Zhou, A., & Zhang, Q. (2015). Are all the subproblems equally important? resource allocation in decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 20(1), 52–64. doi:10.1109/TEVC.2015.2424251

*Qing Zhang received her Ph.D. degree in geosciences information engineering from China University of Geosciences in 2009. She is now a professor with Huanggang Normal University. Her research interests include evolutionary computing, particle swarm optimization, and ant colony optimization.*

*Ruwang Jiao is a PhD candidate in School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan, China. In 2019-2020, he was a joint PhD student in Data Science and Artificial Intelligence Research Center, Nanyang Technological University, Singapore. His current research interests include evolutionary computation, constrained optimization, multi-objective optimization, and antenna design.*

*Sanyou Zeng received the M.Sc. degree in mathematics from Hunan University, Changsha, China, in 1995, and the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 2002. He has been a Professor with the China University of Geosciences, Wuhan, since 2004. His current research interests include evolutionary computation with machine learning for solving problems with constraints, multi-objective, dynamic environments, and expensive costs, especially the antenna design problems.*

*Zhigao Zeng is a professor at school of computer, Hunan University of Technology, China. In 2012, he received his PhD from State key Laboratory of Software Engineering, Wuhan University, China. His research interests include computer vision, artificial intelligence and pattern recognition.*