Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 960524, 24 pages
http://dx.doi.org/10.1155/2013/960524

Research Article

Modified Biogeography-Based Optimization with Local Search Mechanism

Quanxi Feng,[1,2] Sanyang Liu,[1] Qunying Wu,[2] GuoQiang Tang,[2] Haomin Zhang,[2] and Huazhou Chen[2]

[1] *Department of Applied Mathematics, Xidian University, Xi'an 710071, China*
[2] *School of Science, Guilin University of Technology, Guilin 541004, China*

Correspondence should be addressed to Quanxi Feng; fqx9904@gmail.com and Sanyang Liu; liusanyan@126.com

Received 3 August 2013; Revised 6 September 2013; Accepted 10 September 2013

Academic Editor: Ferenc Hartung

Biogeography-based optimization (BBO) is a new effective population optimization algorithm based on the biogeography theory with inherently insufficient exploration capability. To address this limitation, we proposed a modified BBO with local search mechanism (denoted as MLBBO). In MLBBO, a modified migration operator is integrated into BBO, which can adopt more information from other habitats, to enhance the exploration ability. Then, a local search mechanism is used in BBO to supplement with modified migration operator. Extensive experimental tests are conducted on 27 benchmark functions to show the effectiveness of the proposed algorithm. The simulation results have been compared with original BBO, DE, improved BBO algorithms, and other evolutionary algorithms. Finally, the performance of the modified migration operator and local search mechanism are also discussed.

1. Introduction

In practical application, many problems are regarded as optimization problems. Several effective techniques have been developed for solving optimization problems. Traditional techniques [1, 2] are effective methods for solving these problems, but they need to know the property of problems, such as continuity or differentiability. In the past few decades, various evolutionary algorithms have been sprung up for solving complex optimization problems, for example, genetic algorithm (GA) [3], evolutionary programming (EP) [4], particle swarm optimization (PSO) [5], Ant Colony optimization (ACO) [6], differential evolution (DE) [7], and biogeography-based optimization (BBO) [8]. Compared with traditional techniques, evolutionary algorithms can solve optimization problems without using some information such as differentiability.

Biogeography-based optimization (BBO), proposed by Simon [8], is a new entrant in the domain of global optimization based on the theory of biogeography. BBO is developed through simulating the emigration and immigration of species between habitats in the multidimensional solution space, where each habitat represents a candidate solution. Just as species, in biogeography, migrate back and forth between habitats, features in candidate solutions are shared between solutions through migration operator. Good solutions tend to share their features with poor solutions. However, these features, in the origin good solution, may exist in several solutions, both good and poor solutions, which may weaken exploration ability.

Several scholars have been working for enhancing the exploration ability. To mention just a few examples, Gong et al. [9] presented a real-coded BBO (RCBBO) for continuous optimization problem through the use of real code to represent the candidate solution and integration mutation operator to improve the population diversity. Cai et al. [10] proposed hybrid BBO (BBO-EP) by combing evolutionary programming with BBO. Boussaïd et al. [11] used BBO and DE to update the population alternately and gave a two-stage

```
(1) Begin
(2)     For i = 1 to NP
(3)         Select H_i in the light of immigration rate λ_i
(4)         If H_i is selected
(5)             For j = 1 to NP
(6)                 Select H_j according to emigration rate μ_j
(7)                 If rand (0,1) < μ_j
(8)                     H_i (SIV) = H_j (SIV)
(9)                 EndIf
(10)            EndFor
(11)        EndIf
(12)    EndFor
(13) End
```

Algorithm 1: Pseudocode of migration operator.

algorithm (DBBO). Ergezer et al. [12] employed opposition-based learning alongside BBO and developed oppositional BBO (OBBO). Ma et al. [13] incorporated resampling technique into BBO for solving optimization problems under noisy environments. Lohokare et al. [14] proposed a new variant of BBO, called aBBOmDE. In the proposed algorithm, a modified mutation operator and clear duplicate operators are used to accelerate convergence speed, and the modified DE (mDE) is embeded as a neighborhood search operator to improve the fitness. Gong et al. [15] combined the exploration of DE with the exploitation of BBO effectively and presented a hybrid DE with BBO, namely, DE/BBO, for global numerical optimization problem. Li et al. [16] designed perturbing migration operator based on the sinusoidal migration model and advanced perturbing biogeography-based optimization (PBBO). Inspired by multiparent crossover in evolutionary algorithm [17], Li and Yin [18] generalized migration operation based on multiparent crossover and presented multioperator biogeography-based optimization (MOBBO). These algorithms have acquired excellent performance.

It is well known that both exploration ability and exploitation ability are two bases for population-based optimization algorithm. However, the exploration and exploitation are contradictory to each other [19]. How to enhance the exploration ability is a key problem for improving the performance of BBO. For the sake of this, inspired of DE [7], in this paper, we modify the migration operator by applying a modified mutation strategy to enhance the exploration ability. Meanwhile, a local search mechanism is introduced in biogeography-based algorithm to supplement with the modified migration operator. We name this BBO algorithm as modified biogeography-based optimization with local search mechanism (denoted as MLBBO). Analysis and experimental results show that MLBBO with appropriate parameters can obtain excellent performance.

The rest of this paper is organized as follows. In Section 2, we will review the basic BBO algorithm. The modified BBO algorithm, called MLBBO, is presented and analyzed in Section 3, while experimental results are presented and discussed in Section 4. Finally, conclusions and future work are drawn in Section 5.

2. Review of BBO

Biogeography-based optimization (BBO) [8] is a new emerging population-based algorithm proposed through mimic species migration in natural biogeography. In BBO algorithm, each possible solution is a habitat, and their features that characterize suitability are called suitability index variables (SIVs) [10]. The goodness of each solution is named as habitat suitability index (HSI). In BBO, a habitat H is a vector of N values (SIVs) reaching global optima through migration step and mutation step. In original migration, information is shared among habitats based on the immigration rate λ_i and emigration rate μ_i which are linear functions of the number of species in the habitat. The linear model can be calculated as follows:

$$\lambda_i = I\left(1 - \frac{i}{n}\right), \qquad \mu_i = \frac{Ei}{n}, \tag{1}$$

where E is the maximum possible emigration rate, I is the maximum possible immigration rate, n is the maximum number of species, and i is the number of species of the ith solution.

The pseudocode of migration operator [8] can be loosely described in Algorithm 1. Where NP ($= n$) is the population size.

Because of ravenous predator or some other natural catastrophe [20], ecological equilibrium may be destroyed, which could lead to drastically changing the species count and HSI of habitats. This phenomenon can be modeled as SIV mutation, and species count probabilities are used to

```
(1) Begin
(2)     For i = 1 to NP
(3)       Select H_i according to immigration rate λ_i using sinusoidal migration model
(4)       If H_i is selected
(5)         For j = 1 to NP
(6)           Generate fourmutually different integers r1, r2, r3, and r4 in 1... NP
(7)           Select according to emigration rate μ_j
(8)           If rand(0,1)< μ_j
(9)             H_i(SIV) = H_j(SIV)
(10)          Else
(11)            H_i(SIV) = H_best(SIV) + F · (H_r1(SIV) − H_r2(SIV)) + F · (H_r3(SIV) − H_r4(SIV))
(12)          EndIf
(13)        EndFor
(14)      EndIf
(15)  EndFor
(16) End
```

Algorithm 2: Pseudocode of modified migration operator.

determine mutation rates. The mutation probability m_i is expressed as follows:

$$m_i = m_{\max} \cdot \left(\frac{1 - p_i}{p_{\max}} \right), \tag{2}$$

where $m_{\max}$ is a user-defined parameter and $p_{\max} = \arg\max\ p_i, i = 1, 2, \ldots, n$.

In the original mutation operator, the SIV in each solution is probabilistically replaced with a new feature, randomly generated in the whole solution space, which will tend to increase the diversity of the population. Gong et al. [9] modified this mutation operator and suggested three mutation operators, Gaussian mutation operator, Cauchy mutation operator, and Lévy mutation operator used in real space.

3. Modified Biogeography-Based Optimization with Local Search Mechanism

3.1. Modified Migration Operator. Migration operator is a key operator of original BBO, which can improve the performance of BBO. Meanwhile, a habitat is modified through migration operator by simply replacing one of the SIV of the similar habitat, which means that the new habitat absorbs less information from the others. Therefore, migration operator lacks exploration ability.

In order to absorb more information from other habitats, inspired of DE [7], the migration operator is modified by using a DE mutation strategy. DE/best/two mutation formula is used in this paper:

$$\text{DE/best/2} : V_i = X_{\text{best}} + F \cdot \left(X_{r1} - X_{r2}\right) + F \cdot \left(X_{r3} - X_{r4}\right), \tag{3}$$

where X_{best} is the best solution, F is scaling factor, $V_i\ i \in \{1, 2, \ldots, NP\}$ is the ith trial vector, and X_{r1}, X_{r2}, X_{r3}, and X_{r4} are for mutually different solutions.

BBO has acquired excellent performance since original migration operator can reserve good features in the population which are avoided being loosed or destroyed in the evolution process. In order to keep the exploitation ability of BBO, we modified formula (3) as follows:

$$\begin{aligned} &\text{BBO/best/2} : H_i\,(\text{SIV}) \\ &\quad = H_{\text{best}}\,(\text{SIV}) + F \cdot \left(H_{r1}\,(\text{SIV}) - H_{r2}\,(\text{SIV})\right) \\ &\qquad + F \cdot \left(H_{r3}\,(\text{SIV}) - H_{r4}\,(\text{SIV})\right), \end{aligned} \tag{4}$$

where H_{best} is the best habitat, H_i is the immigration habitat, and H_{r1}, H_{r2}, H_{r3}, and H_{r4} are four mutually different habitats. From this, we can see from formula (4) that two differences are added to the new habitat and more information from other habitats are adopted, which can enhance the exploration ability. Meanwhile, it should be noted that parameter F plays an important role in balancing the exploration ability and exploitation ability. When F takes 0, formula (4) absorbs information from best habitat, its exploitation ability enhanced. When F increases, formula (4) absorbs more information from other habitats, and its exploration ability will increase, but their exploitation ability will decrease a certain degree. Of course, this silver lining has its cloud. Therefore, F takes 0.5 in this paper.

In natural biogeography, the migration model may be nonlinear. Hence, Ma [21] introduced six migration models. Comparison results show that sinusoidal migration model is the best one. The sinusoidal migration model can be calculated as follows:

$$\begin{aligned} \lambda_i &= \frac{I}{2}\left(1 + \cos\left(\frac{i\pi}{n}\right)\right), \\ \mu_i &= \frac{E}{2}\left(1 - \cos\left(\frac{i\pi}{n}\right)\right). \end{aligned} \tag{5}$$

```
(1) Begin
(2)     For i = 1 to NP
(3)       Compute the mutation probability p
(4)       Select variable H_i with probability p
(5)       If H_i is selected
(6)           H_i(SIV) = H_i(SIV) + δ(1)
(7)       Endif
(8)     Endfor
(9) End
```

Algorithm 3: Pseudocode of Cauchy mutation operator.

In order to enhance exploration ability, we modify migration operator, described in Algorithm 1, by combining formula $H_i(\mathrm{SIV}) = H_j(\mathrm{SIV})$ and formula (4). The modified migration operator is described in Algorithm 2 .

We can see from Algorithm 2 that immigration habitats not only accept information from best habitat, but also from BBO/best/2. On the one hand, information from HSI habitats make the operator maintain power exploitation ability. On the other hand, information from BBO/best/2 can enhance the exploration ability. From this, we know that this modified migration operator can preserve the exploitation ability and enhance the exploration ability simultaneously.

3.2. Mutation with Mutation Operator. Cataclysmic events can drastically change the HSI of a natural habitat and cause species count to differ from its equilibrium value. This phenomenon can be modeled in BBO as SIV mutation, and species count probabilities are used to determine mutation rates.

In order to enhance the exploration ability of BBO, a modified mutation operator, Cauchy mutation operator, is integrated into BBO. The probability density function of Cauchy distribution [9] is

$$f_t(x) = \frac{1}{\pi}\frac{t}{t^2 + x^2}, \tag{6}$$

where $x \in R$ and $t > 0$ is a scale parameter.

The Cauchy mutation ($t = 1$) in this paper can be described as

$$H_i(\mathrm{SIV}) = H_i(\mathrm{SIV}) + \delta(1), \tag{7}$$

where $\delta(1)$ is a randomness Cauchy value when parameter t equals 1. In this paper, we use the Cauchy distribution to update the solution, which can be described in Algorithm 3.

3.3. Local Search Mechanism. As Simon has pointed out, the convergence speed is an obstacle for real application. So, finding mechanisms to speed up BBO could be an important area for further research. Therefore, a local search mechanism is proposed to accelerate the convergence speed of BBO. In order to remedy the insufficiency of modified migration operator, which mainly updates worse habitats selected by immigration rate, the local search mechanism is designed mainly for better habitats in the first half population.

For each habitat H_i in the first half population, define

$$H_i(\mathrm{SIV}) = \begin{cases} H_i(\mathrm{SIV}) + alpha * \left(H_k(\mathrm{SIV}) - H_i(\mathrm{SIV})\right) & rand(0,1) < p_l, \\ H_i(\mathrm{SIV}) & \text{else}, \end{cases} \tag{8}$$

where H_k denotes a habitat randomly selected from population, p_l denotes the local search probability, and *alpha* is the strength of local search.

3.4. Selection Operator. Selection operator is used to select better habitats preserving in next generation, which can impel the metabolism of population and retain good habitats. In this paper, greedy selection operator is used. In other words, the original habitat H_i will be replaced by its corresponding trial habitat M_i if the HSI of the trial habitat M_i is better than that of its original habitat H_i:

$$H_i = \begin{cases} H_i & \text{if } f(H_i) < f(M_i) \\ M_i & \text{else.} \end{cases} \tag{9}$$

3.5. Main Procedure of MLBBO. Exploration ability is mainly a contributory factor to step down the performance of BBO algorithm. In order to improve the performance of BBO, a modified BBO algorithm, called MLBBO, is proposed for global optimization problems in the continuous domain. In MLBBO, the original migration operator and mutation operator are modified to enhance the exploration ability. Moreover, local search mechanism and selection operator are integrated into MLBBO to supplement with migration operator. The pseudo-code of MLBBO is described in Algorithm 4, where *NP* is the population size.

4. Experimental Study

4.1. Experimental Setting. In this section, several experimental tests are carried out on 27 benchmark functions for

```
(1) Begin
(2)    Initialize the population Pop with NP habitats randomly
(3)    Evaluate the fitness (HSI) for each Habitat H_i in Pop
(4)    While the halting criteria are not satisfied do
(5)       Sort all the habitats from best to worst in line with their fitness values
(6)       Map the fitness to the number of species count S for each habitat
(7)       Calculate the immigration rate and emigration rate according to sinusoidal migration model
(8)       For i = 1 to NP                /*migration stage*/
(9)          Select H_i according to immigration rate λ_i using sinusoidal migration model
(10)         If H_i is selected
(11)            For j = 1 to NP
(12)               Generate fourmutually different integers r1, r2, r3, and r4 in {1 ··· NP}
(13)               Select H_j according to emigration rate μ_j
(14)               If rand(0,1) < μ_j
(15)                  H_i(SIV) = H_j(SIV)
(16)               Else
(17)                  H_i(SIV) = H_best(SIV) + F · (H_r1(SIV) − H_r2(SIV)) + F · (H_r3(SIV) − H_r4(SIV))
(18)               EndIf
(19)            EndFor
(20)         EndIf
(21)      EndFor              /*end ofmigration stage*/
(22)      For i = 1 to NP             /*mutation stage*/
(23)         Compute the mutation probability p_i
(24)         Select variable H_i with probability p_i
(25)         If H_i is selected
(26)            H_i(SIV) = H_i(SIV) + δ(1)
(27)         Endif
(28)      EndFor                 /*end ofmutation stage*/
(29)      For i = 1 to NP/2         /*local search stage*/
(30)            If rand(0,1) < p_l
(31)               Select a habitat H_k randomly
(32)               H_i(SIV) = H_i(SIV) + alpha ∗ (H_k(SIV) − H_i(SIV))
(33)            EndIf
(34)      EndFor               /*end oflocal search stage*/
(35)      Make sure each habitat legal based on boundary constraints
(36)      Use new generated habitat to replace duplicate
(37)      Evaluate the fitness (HSI) for trial habitat M_i in the new population Pop
(38)      For i = 1 to NP              /*selection stage*/
(39)         If f(M_i) < f(H_i)
(40)            H_i = M_i
(41)         EndIf
(42)      EndFor               /*end ofselection stage*/
(43)   EndWhile
(44) End
```

Algorithm 4: Pseudocode of MLBBO.

validating the performance of the proposed algorithm. These functions consist of 13 standard benchmark functions [22] (f1–f13) and 14 complex functions (F1–F14) from the set of test problems listed in IEEE CEC 2005 [23]. All the test functions used in our experiments are high-dimensional functions with changeable dimension, which are only briefly summarized in Table 1. A more detailed description of each function is given in Table 8 (see also [22, 23]).

For fair comparison, we have chosen a reasonable set of parameter values. For all experimental simulations in this paper, we will use the following parameters setting unless a change is mentioned:

(1) population size: $NP = 100$;

(2) maximum immigration rate: $I = 1$;

(3) maximum emigration rate: $E = 1$;

(4) mutation probability: $m_{\max} = 0.001$;

(5) value to reach (VTR) = 10^{-6}, except for f7, F6–F14 of VTR = 10^{-2};

Table 1: Test unctions used in our experimental tests.

	Function name	Dimension	Search Space	Optima
f1	Sphere	n	[−100, 100]	0
f2	Schwefel's Problem 2.22	n	[10, 10]	0
f3	Schwefel's Problem 1.2	n	[−100, 100]	0
f4	Schwefel's Problem 2.21	n	[−100, 100]	0
f5	Rosenbrock function	n	[−30, 30]	0
f6	Step function	n	[−100, 100]	0
f7	Quartic function	n	[−1.28, 1.28]	0
f8	Schwefel function	n	[−512, 512]	−12569.5
f9	Rastrigin function	n	[−5.12, 5.12]	0
f10	Ackley function	n	[−32, 32]	0
f11	Griewank function	n	[−600, 600]	0
f12	Penalized function 1	n	[−50, 50]	0
f13	Penalized function 2	n	[−50, 50]	0
F1	Shifted sphere function	n	[−100, 100]	−450
F2	Shifted Schwefel's Problem 1.2	n	[−100, 100]	−450
F3	Shifted rotated high conditioned elliptic function	n	[−100, 100]	−450
F4	Shifted Schwefel's Problem 1.2 with noise in fitness	n	[−100, 100]	−450
F5	Schwefel's Problem 2.6 with global optimum on bounds	n	[−30, 30]	−310
F6	Shifted Rosenbrock's function	n	[−100, 100]	390
F7	Shifted rotated Griewank's function without bounds	n	Initialize in [0, 600]	−180
F8	Shifted rotated Ackley's function with global optimum on bounds	n	[−32, 32]	−140
F9	Shifted Rastrigin's function	n	[−5.12, 5.12]	−330
F10	Shifted Rotated Rastrigin's function	n	[−5.12, 5.12]	−330
F11	Shifted Rotated Weierstrass function	n	[−600, 600]	90
F12	Schwefel's Problem 2.13	n	[−100, 100]	−460
F13	Expanded extended Griewank's plus Rosenbrock's function (F8F2)	n	[−50, 50]	−130
F14	Shifted rotated expanded scaffer's F6	n	[−100, 100]	−300

(6) maximum number of fitness function evaluations (MaxNFFEs): 150,000 for f1, f6, f10, f12, and f13, 200,000 for f2, and f11, 300,000 for f7, f8, f9, F1–F14, and 500,000 for f3, f4, and f5.

In our experimental tests, all the functions are optimized over 30 independent runs. All experimental tests are carried out on a computer with 1.86 GHz Intel core Processor, 1 GB of RAM, and Window XP operation system in Matlab software 7.6.

4.2. Effect of the Strength Alpha and Rate p_l on the Performance of MLBBO. In this section, we investigate the impact of local search strength *alpha* and search probability p_l on the proposed algorithm. In order to analyze the performance more scientific, eight functions are chosen from Table 1. They are Sphere, Schwefel 1.2, Schwefel, Penalized2, F1, F3, F10, and F13, which belong to standard unimodal function, standard multimodal function, shifted function, shifted rotated function, and expanded function, respectively. Search strength *alpha* are 0.3, 0.5, 0.8, and 1.1. Search rates p_l are 0, 0.2, 0.4, 0.6, 0.8, and 1. It is important to note that p_l equal to 0 means that local search mechanism is not conducted. Hence, in our experiments, *alpha* and p_l are tested according to 24 situations. MLBBO algorithm executes 30 independent runs on each function, and convergence figures of average fitness values (which are function error values) are plotted in Figure 1.

From Figure 1, we can observe that different search rate p_l have different results. When p_l is 0, the results are worse than the others, which means that local search mechanism can improve the performance of proposed algorithm. When p_l is 0.4, we obtain a faster convergence speed and better results on the Sphere function. For the other seven test functions,

Table 2: Comparisons with BBO and DE.

	DE			BBO			MLBBO		
	Mean (std)	MeanFEs	SR	Mean (Std)	MeanFEs	SR	Mean (std)	MeanFEs	SR
f1	$2.72E-20\ (2.09E-20)^{+}$	$4.47E+04$	30	$3.23E-01\ (1.18E-01)^{+}$	NaN	0	$2.78E-31\ (3.25E-31)$	$2.83E+04$	30
f2	$2.56E-12\ (3.09E-12)^{+}$	$1.06E+05$	30	$4.84E-01\ (9.06E-02)^{+}$	NaN	0	$1.43E-21\ (7.11E-22)$	$5.74E+04$	30
f3	$2.31E-19\ (2.37E-19)^{+}$	$1.51E+05$	30	$1.26E+02\ (5.19E+01)^{+}$	NaN	0	$1.90E-20\ (3.03E-20)$	$1.40E+05$	30
f4	$1.92E-10\ (3.44E-10)^{\sim}$	$2.61E+05$	30	$1.62E+00\ (4.71E-01)^{+}$	NaN	0	$4.49E-08\ (1.25E-07)$	$3.49E+05$	30
f5	$1.05E-28\ (8.92E-29)^{-}$	$1.62E+05$	30	$7.85E+01\ (3.51E+01)^{+}$	NaN	0	$3.34E-21\ (9.08E-21)$	$2.25E+05$	30
f6	$8.67E-01\ (1.20E+00)^{+}$	$2.51E+04$	14	$1.77E+00\ (1.17E+00)^{+}$	$1.19E+05$	3	$0.00E+00\ (0.00E+00)$	$1.14E+04$	30
f7	$5.65E-03\ (2.10E-03)^{+}$	$1.44E+05$	29	$3.64E-04\ (2.77E-04)^{-}$	$2.21E+04$	30	$2.23E-03\ (9.91E-04)$	$6.61E+04$	30
f8	$7.29E+03\ (2.32E+02)^{+}$	NaN	0	$2.53E-01\ (8.82E-02)^{+}$	NaN	0	$0.00E+00\ (0.00E+00)$	$5.53E+04$	30
f9	$1.76E+02\ (1.01E+01)^{+}$	NaN	0	$3.56E-02\ (1.52E-02)^{+}$	NaN	0	$0.00E+00\ (0.00E+00)$	$9.16E+04$	30
f10	$5.77E-07\ (3.16E-06)^{\sim}$	$7.87E+04$	29	$2.06E-01\ (4.94E-02)^{+}$	NaN	0	$6.10E-15\ (1.14E-15)$	$5.05E+04$	30
f11	$6.57E-03\ (6.25E-03)^{+}$	$4.54E+04$	12	$2.81E-01\ (9.76E-02)^{+}$	NaN	0	$2.87E-03\ (5.02E-03)$	$3.33E+04$	22
f12	$1.38E-02\ (4.50E-02)^{\sim}$	$5.37E+04$	26	$1.98E-03\ (2.04E-03)^{+}$	NaN	0	$5.88E-28\ (3.22E-27)$	$2.57E+04$	30
f13	$1.90E-20\ (5.93E-20)^{\sim}$	$4.65E+04$	30	$2.09E-02\ (9.94E-03)^{+}$	NaN	0	$5.09E-32\ (3.13E-32)$	$2.71E+04$	30
F1	$5.59E-28\ (2.06E-28)^{+}$	$4.58E+04$	30	$7.23E-02\ (3.27E-02)^{+}$	NaN	0	$3.03E-29\ (6.97E-29)$	$2.90E+04$	30
F2	$7.59E-12\ (8.42E-12)^{+}$	$1.72E+05$	30	$3.55E+03\ (1.35E+03)^{+}$	NaN	0	$1.97E-12\ (2.55E-12)$	$1.58E+05$	30
F3	$1.39E+05\ (1.06E+05)^{+}$	NaN	0	$1.14E+07\ (4.90E+06)^{+}$	NaN	0	$9.26E+04\ (4.90E+04)$	NaN	0
F4	$7.10E-05\ (1.61E-04)^{\sim}$	$2.85E+05$	1	$1.39E+04\ (4.56E+03)^{+}$	NaN	0	$7.00E-05\ (1.87E-04)$	$2.90E+05$	5
F5	$5.31E+01\ (1.37E+02)^{-}$	NaN	0	$5.81E+03\ (9.97E+02)^{+}$	NaN	0	$8.34E+02\ (3.81E+02)$	NaN	0
F6	$1.58E-13\ (7.82E-13)^{\sim}$	$1.46E+05$	30	$2.23E+02\ (8.03E+01)^{+}$	NaN	0	$2.11E-09\ (6.68E-09)$	$1.85E+05$	30
F7	$1.26E-02\ (1.26E-02)^{\sim}$	$4.93E+04$	15	$1.13E+00\ (7.73E-02)^{+}$	NaN	0	$1.59E-02\ (1.16E-02)$	$4.80E+04$	12
F8	$2.09E+01\ (5.20E-02)^{\sim}$	NaN	0	$2.08E+01\ (1.06E-01)^{-}$	NaN	0	$2.09E+01\ (7.07E-02)$	NaN	0
F9	$1.85E+02\ (1.75E+01)^{+}$	NaN	0	$3.97E-02\ (2.04E-02)^{+}$	NaN	0	$5.92E-17\ (3.24E-16)$	$7.71E+04$	30
F10	$2.05E+02\ (1.96E+01)^{+}$	NaN	0	$5.26E+01\ (1.19E+01)^{+}$	NaN	0	$3.47E+01\ (1.24E+01)$	NaN	0
F11	$3.95E+01\ (1.03E+00)^{+}$	NaN	0	$3.09E+01\ (3.05E+00)^{+}$	NaN	0	$1.72E+01\ (6.58E+00)$	NaN	0
F12	$1.19E+03\ (2.32E+03)^{\sim}$	$2.31E+05$	3	$1.41E+04\ (9.84E+03)^{+}$	NaN	0	$2.07E+03\ (2.92E+03)$	NaN	0
F13	$1.59E+01\ (1.12E+00)^{+}$	NaN	0	$1.09E+00\ (2.34E-01)^{-}$	NaN	0	$3.03E+00\ (1.44E-01)$	NaN	0
F14	$1.34E+01\ (1.51E-01)^{+}$	NaN	0	$1.31E+01\ (3.83E-01)^{+}$	NaN	0	$1.27E+01\ (2.40E-01)$	NaN	0
better	16			24					
similar	9			0					
worse	2			3					

Note: "+", "−", and "~" indicate that MLBBO is significantly better, worse, and indifferent, respectively, compared with other algorithms.

better results are obtained when search rate p_l is 0.2 and 0.4. Moreover, the performance of MLBBO is most sensitive to search rate p_l on function F8.

From Figure 1, we can also see that search strength *alpha* has little effect on the results as a whole. By careful looking, search strength *alpha* has an effect on the results when search rate p_l is large, especially, when p_l equals 0.8 or 1.

For considering opinions of both sides, the search strength *alpha* and search rate p_l are set as 0.8 and 0.2, respectively.

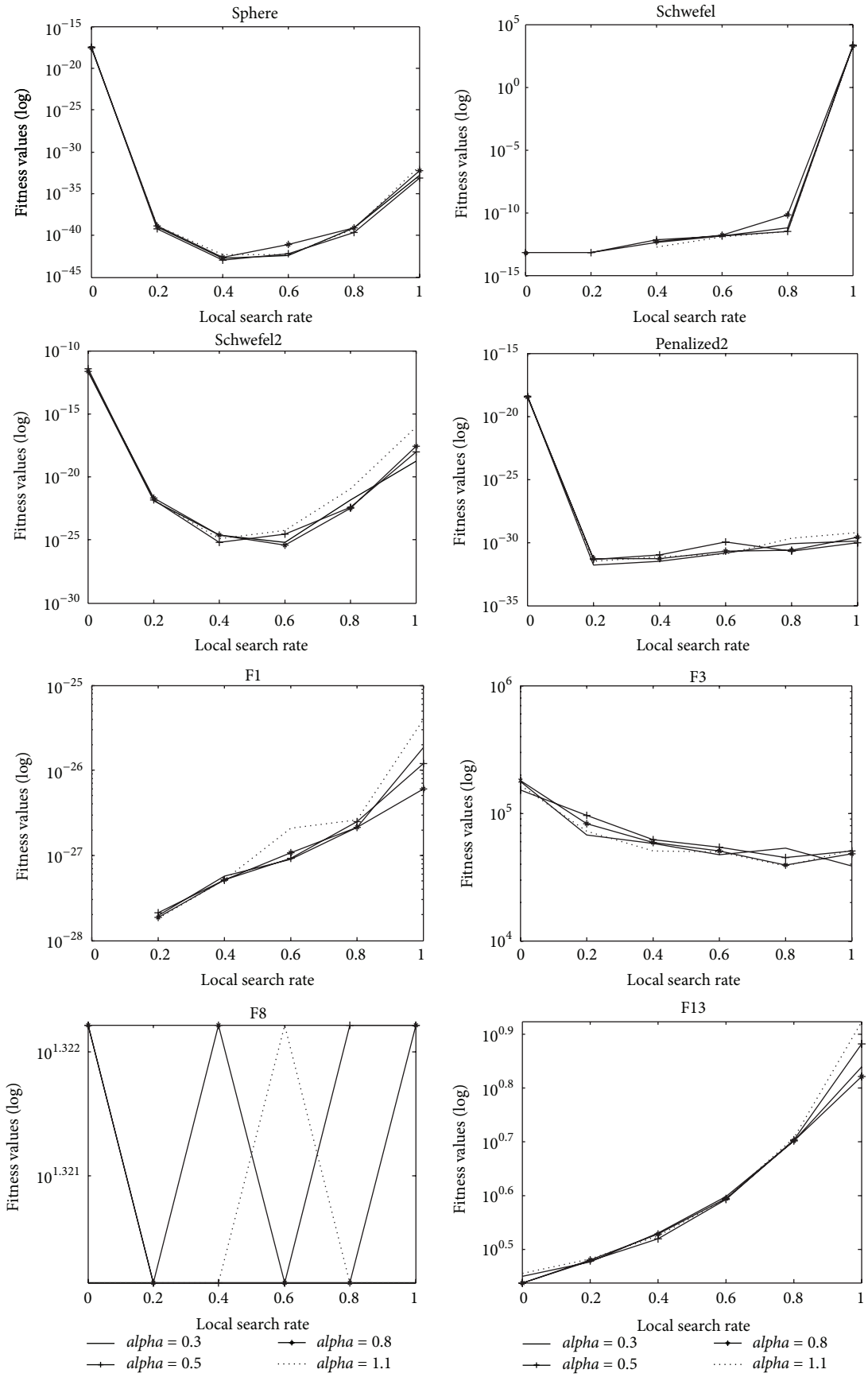


Figure 1: Results of MLBBO on eight test functions with different *alpha* and p_I.

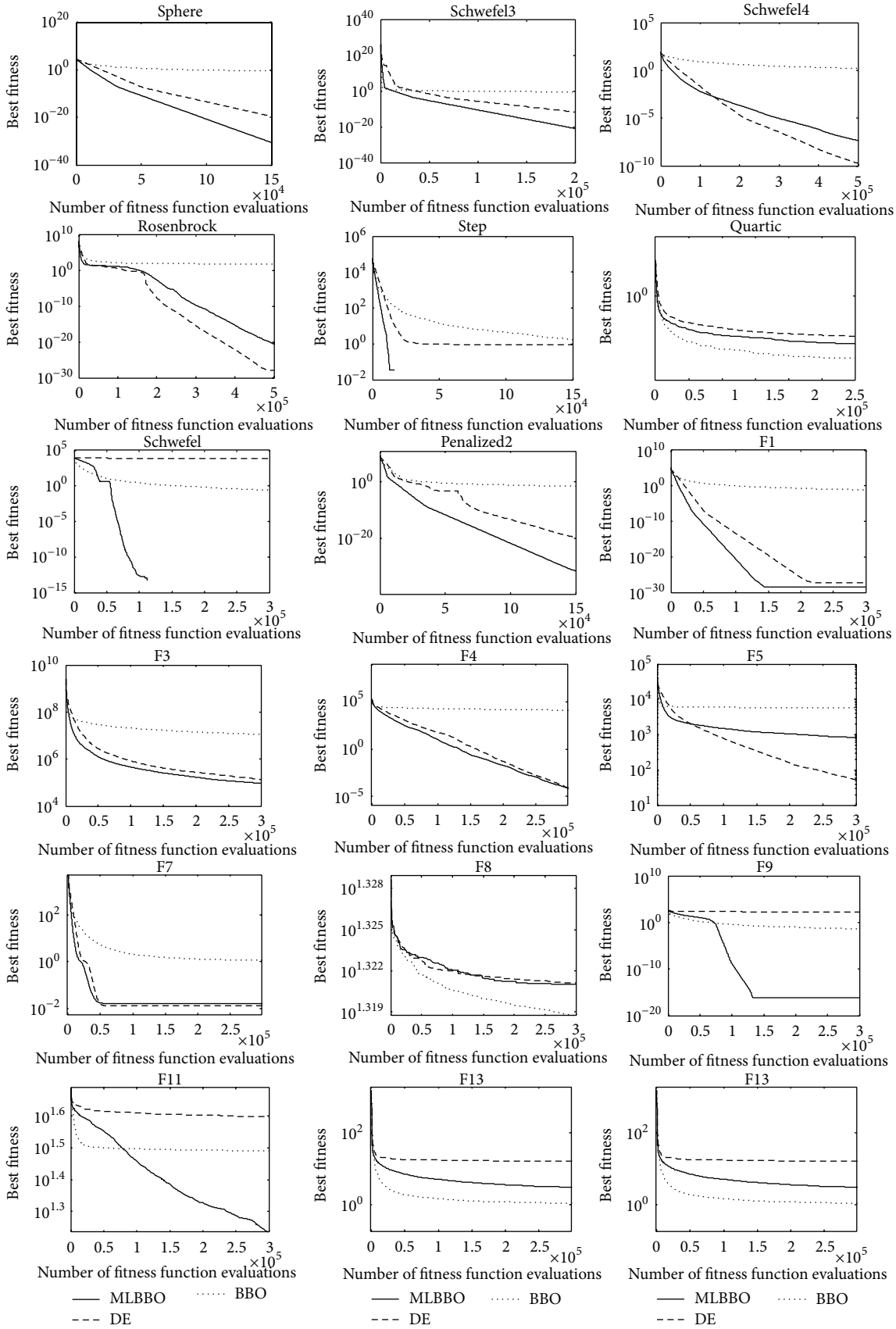


Figure 2: Convergence curves of mean fitness values of MLBBO, BBO, and DE for functions f1, f2, f3, f5, f6, f7, f8, f13, F1, F3, F4, F5, F7, F8, F9, F11, F13, and F14.

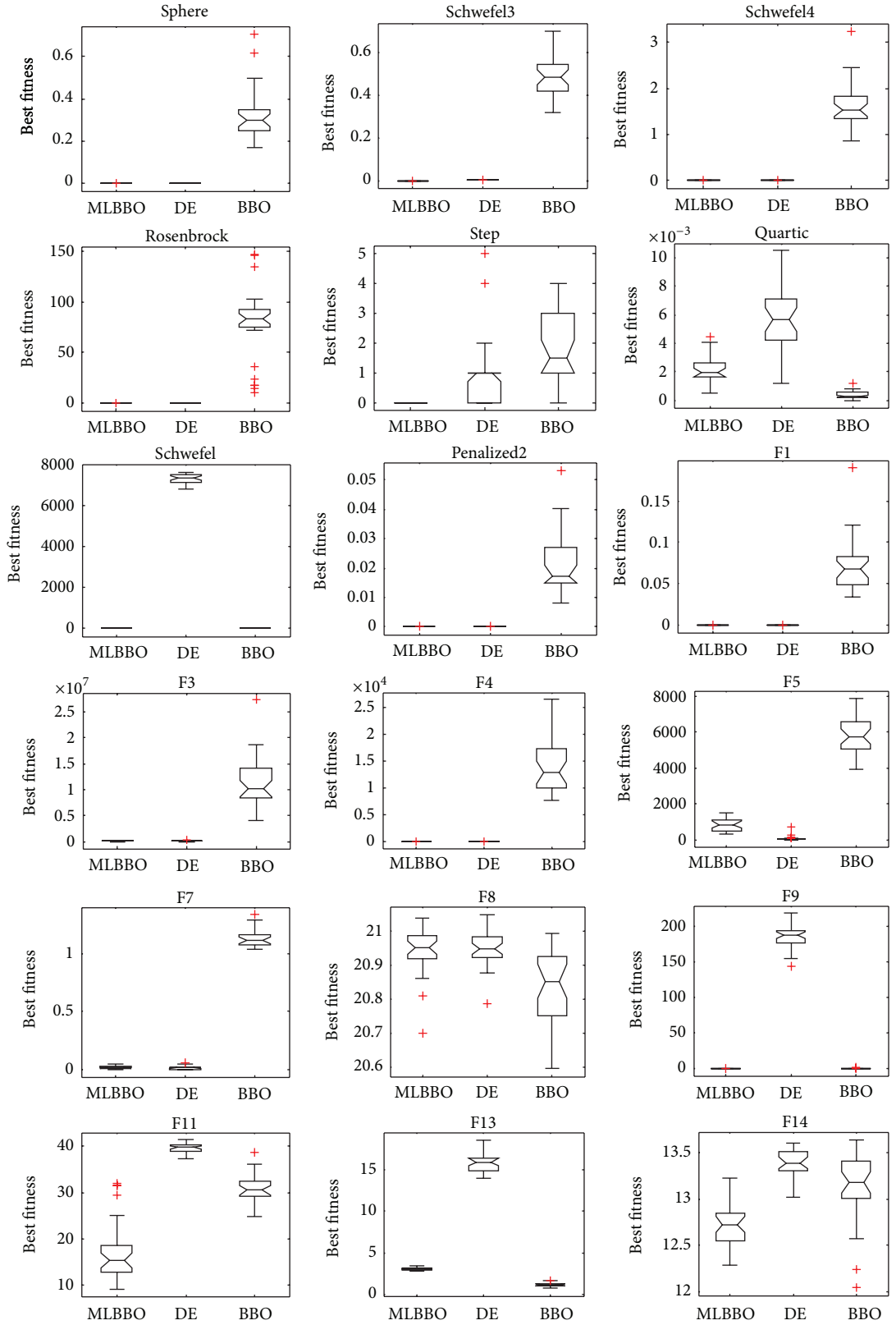


Figure 3: ANOVA tests of fitness values of MLBBO, BBO, and DE for functions f1, f2, f3, f5, f6, f7, f8, f13, F1, F3, F4, F5, F7, F8, F9, F11, F13, and F14.

Table 3: Comparisons with improved BBO algorithms.

	OBBO	PBBO	MOBBO	RCBBO	MLBBO
	Mean (std)	Mean (Std)	Mean (std)	Mean (std)	Mean (std)
f1	$6.88E-05\ (2.58E-05)^{+}$	$2.23E-11\ (2.77E-12)^{+}$	$1.70E-09\ (9.30E-09)^{\sim}$	$2.26E-03\ (1.06E-03)^{+}$	$2.11E-31\ (1.25E-31)$
f2	$2.76E-02\ (4.80E-03)^{+}$	$3.47E-06\ (2.67E-07)^{+}$	$2.74E-05\ (1.48E-04)^{\sim}$	$8.96E-02\ (1.58E-02)^{+}$	$1.58E-21\ (7.25E-22)$
f3	$6.29E-01\ (4.23E-01)^{+}$	$2.06E-02\ (1.70E-02)^{+}$	$3.25E-02\ (4.50E-02)^{+}$	$2.19E+01\ (9.34E+00)^{+}$	$1.42E-20\ (1.70E-20)$
f4	$1.47E-02\ (3.18E-03)^{+}$	$1.20E-02\ (3.30E-03)^{+}$	$2.43E-02\ (6.74E-03)^{+}$	$1.20E+00\ (1.25E+00)^{+}$	$5.28E-08\ (1.02E-07)$
f5	$3.65E+01\ (2.20E+01)^{+}$	$3.60E+01\ (3.40E+01)^{+}$	$6.32E+01\ (8.33E+01)^{+}$	$4.25E+01\ (3.74E+01)^{+}$	$5.34E-21\ (1.10E-20)$
f6	$0.00E+00\ (0.00E+00)^{\sim}$	$0.00E+00\ (0.00E+00)^{\sim}$	$0.00E+00\ (0.00E+00)^{\sim}$	$0.00E+00\ (0.00E+00)^{\sim}$	$0.00E+00\ (0.00E+00)$
f7	$4.02E-04\ (3.48E-04)^{-}$	$2.86E-04\ (2.68E-04)^{-}$	$9.79E-04\ (8.00E-04)^{-}$	$4.21E-04\ (3.81E-04)^{-}$	$2.33E-03\ (1.04E-03)$
f8	$2.40E+02\ (1.82E+02)^{+}$	$1.70E-11\ (2.26E-12)^{+}$	$1.55E-11\ (8.50E-11)^{\sim}$	$8.17E-05\ (3.15E-05)^{+}$	$0.00E+00\ (0.00E+00)$
f9	$5.59E-03\ (2.07E-03)^{+}$	$2.80E-07\ (1.37E-06)^{\sim}$	$5.45E-05\ (2.99E-04)^{\sim}$	$1.38E-02\ (5.11E-03)^{+}$	$0.00E+00\ (0.00E+00)$
f10	$7.38E-03\ (1.88E-03)^{+}$	$2.08E-06\ (5.39E-06)^{+}$	$9.74E-08\ (5.97E-09)^{+}$	$2.94E-02\ (6.63E-03)^{+}$	$6.10E-15\ (6.49E-16)$
f11	$2.68E-02\ (2.49E-02)^{+}$	$4.98E-03\ (1.29E-02)^{\sim}$	$5.08E-03\ (1.05E-02)^{\sim}$	$7.92E-02\ (4.36E-02)^{+}$	$1.73E-03\ (4.00E-03)$
f12	$1.82E-06\ (2.26E-06)^{+}$	$6.71E-11\ (3.18E-10)^{\sim}$	$3.46E-03\ (1.89E-02)^{\sim}$	$3.39E-05\ (3.47E-05)^{+}$	$2.35E-32\ (3.98E-32)$
f13	$2.04E-05\ (1.99E-05)^{+}$	$1.65E-09\ (7.99E-09)^{\sim}$	$6.33E-13\ (3.44E-12)^{\sim}$	$6.43E-04\ (9.66E-04)^{+}$	$5.64E-32\ (3.38E-32)$
F1	$1.40E-05\ (8.26E-06)^{+}$	$2.67E-11\ (9.69E-11)^{\sim}$	$1.16E-07\ (6.34E-07)^{\sim}$	$3.50E-04\ (9.25E-05)^{+}$	$2.02E-29\ (6.16E-29)$
F2	$4.70E+01\ (8.64E+00)^{+}$	$2.92E+00\ (2.02E+00)^{+}$	$2.66E+00\ (2.16E+00)^{+}$	$8.72E+02\ (4.43E+02)^{+}$	$1.37E-12\ (1.53E-12)$
F3	$1.56E+06\ (3.45E+05)^{+}$	$2.10E+06\ (6.86E+05)^{+}$	$2.37E+06\ (9.89E+05)^{+}$	$4.46E+06\ (1.58E+06)^{+}$	$9.02E+04\ (5.54E+04)$
F4	$3.18E+03\ (9.94E+02)^{+}$	$7.74E+02\ (3.86E+02)^{+}$	$1.46E+01\ (1.83E+01)^{+}$	$2.17E+04\ (8.50E+03)^{+}$	$1.75E-05\ (2.92E-05)$
F5	$1.03E+04\ (1.63E+03)^{+}$	$3.68E+03\ (6.27E+02)^{+}$	$5.04E+03\ (1.24E+03)^{+}$	$6.31E+03\ (1.18E+03)^{+}$	$8.00E+02\ (3.79E+02)$
F6	$3.20E+02\ (9.64E+02)^{\sim}$	$8.71E+02\ (2.12E+03)^{+}$	$2.76E+02\ (8.46E+02)^{\sim}$	$8.45E+02\ (2.69E+03)^{\sim}$	$3.55E-09\ (1.40E-08)$
F7	$2.46E-02\ (1.52E-02)^{+}$	$1.62E-02\ (1.45E-02)^{\sim}$	$2.26E-02\ (1.99E-02)^{+}$	$7.64E-01\ (1.39E+00)^{+}$	$1.38E-02\ (1.21E-02)$
F8	$2.08E+01\ (8.04E-02)^{-}$	$2.09E+01\ (1.69E-01)^{-}$	$2.07E+01\ (1.71E-01)^{-}$	$2.07E+01\ (8.51E-02)^{-}$	$2.09E+01\ (5.48E-02)$
F9	$5.80E-03\ (2.39E-03)^{+}$	$9.69E-07\ (3.88E-06)^{\sim}$	$1.22E-07\ (6.61E-07)^{\sim}$	$1.32E-02\ (5.53E-03)^{+}$	$5.92E-17\ (3.24E-16)$
F10	$6.43E+01\ (2.81E+01)^{+}$	$3.29E+01\ (8.80E+00)^{\sim}$	$7.14E+01\ (2.06E+01)^{+}$	$4.80E+01\ (1.62E+01)^{+}$	$3.69E+01\ (1.45E+01)$
F11	$2.36E+01\ (3.19E+00)^{+}$	$2.09E+01\ (4.48E+00)^{\sim}$	$2.70E+01\ (4.11E+00)^{+}$	$3.19E+01\ (3.90E+00)^{+}$	$1.95E+01\ (7.87E+00)$
F12	$1.18E+04\ (8.66E+03)^{+}$	$4.68E+03\ (4.44E+03)^{+}$	$5.04E+03\ (4.10E+03)^{+}$	$1.29E+04\ (8.94E+03)^{+}$	$2.43E+03\ (2.97E+03)$
F13	$1.08E+00\ (1.96E-01)^{-}$	$1.14E+00\ (2.07E-01)^{-}$	$1.09E+00\ (2.29E-01)^{-}$	$9.61E-01\ (1.57E-01)^{-}$	$2.98E+00\ (1.62E-01)$
F14	$1.25E+01\ (3.61E-01)^{\sim}$	$1.19E+01\ (6.17E-01)^{-}$	$1.33E+01\ (3.44E-01)^{+}$	$1.31E+01\ (3.39E-01)^{+}$	$1.27E+01\ (2.39E-01)$
Better	21	13	13	22	
Similar	3	10	11	2	
Worse	3	4	3	3	

Note: "+", "−", and "∼" indicate that MLBBO is significantly better, worse, and indifferent, respectively, compared with other algorithms.

4.3. Comparisons with BBO and DE/Best/2/Bin. For evaluating the performance of algorithms, we first compare MLBBO with BBO [8], which is consistent with original BBO except solution representation (real code) and DE/best/2/bin [7] (abbreviation as DE). For fair comparison, three algorithms are conducted on 27 functions with the same parameters setting above. The average and standard deviation of fitness values (which are function error values) after completion of 30 Monte Carlo simulations are summarized in Table 2. The number of successful runs and average of the number of fitness function evaluations (MeanFEs), needed in each run when the VTR is reached within the MaxNFFEs, are also recorded in Table 2. Furthermore, results of MLBBO and BBO, MLBBO and DE are compared with statistical paired t-test [24, 25], which is used to determine whether two paired solution sets differ from each other in a significant way. Convergence curves of mean fitness values and boxplot figures under 30 independent runs are listed in Figures 2 and 3, respectively.

From Table 2, it is seen that MLBBO is significantly better than BBO on 24 functions out of 27 test functions, and MLBBO is outperformed by BBO on the rest 3 functions. When compared with DE, MLBBO gives significantly better performance on 16 functions. On 9 functions, there are no significant differences between MLBBO and DE based on the two tail t-test. For the rest 2 functions, DE are better than MLBBO. Moreover, MLBBO can obtain the VTR over all 30 successful runs within the MaxNFFEs for 16 out of 27 functions. Especially, for the first 13 standard benchmark functions, MLBBO obtains 30 successful runs on 12 functions except Griewank function, where 22 successful runs are obtained by MLBBO. DE and BBO obtain 30 successful runs on 9 and 1 out of 27 functions, respectively. From Table 2, we can find that MLBBO needs less or similar NFFEs than DE and BBO on most of the successful runs, which means that MLBBO has faster convergence speed than DE and BBO. The convergence speed and the robustness can also show in Figures 2 and 3.

Table 4: Comparisons with hybrid algorithms.

	OXDE			DE/BBO			MLBBO		
	Mean (std)	MeanFEs	SR	Mean (std)	MeanFEs	SR	Mean (std)	MeanFEs	SR
f1	$2.60E-03\ (9.04E-04)^{+}$	NaN	0	$1.53E-30\ (1.48E-30)^{+}$	$4.7E+04$	30	$2.30E-31\ (2.74E-31)$	$2.8E+04$	30
f2	$1.82E-02\ (4.77E-03)^{+}$	NaN	0	$2.52E-24\ (1.74E-24)^{-}$	$6.7E+04$	30	$1.61E-21\ (7.89E-22)$	$5.8E+04$	30
f3	$5.43E-01\ (2.52E-01)^{+}$	NaN	0	$2.60E-03\ (1.72E-03)^{+}$	NaN	0	$2.16E-20\ (3.19E-20)$	$1.4E+05$	30
f4	$4.80E-02\ (7.19E-02)^{+}$	NaN	0	$6.50E-02\ (2.23E-01)^{\sim}$	$2.5E+05$	18	$7.99E-08\ (1.38E-07)$	$3.5E+05$	30
f5	$3.73E-01\ (7.68E-01)^{+}$	NaN	0	$2.33E+01\ (9.10E+00)^{+}$	NaN	0	$2.56E-21\ (6.21E-21)$	$2.3E+05$	30
f6	$0.00E+00\ (0.00E+00)^{\sim}$	$9.3E+04$	30	$0.00E+00\ (0.00E+00)^{\sim}$	$2.2E+04$	30	$0.00E+00\ (0.00E+00)$	$1.1E+04$	30
f7	$6.64E-03\ (1.97E-03)^{+}$	$2.0E+05$	30	$4.69E-03\ (1.54E-03)^{+}$	$1.4E+05$	30	$2.13E-03\ (9.10E-04)$	$6.7E+04$	30
f8	$1.33E-05\ (1.57E-05)^{+}$	NaN	0	$0.00E+00\ (0.00E+00)^{\sim}$	$1.0E+05$	30	$6.06E-14\ (3.32E-13)$	$5.5E+04$	30
f9	$5.32E+01\ (6.72E+00)^{+}$	NaN	0	$0.00E+00\ (0.00E+00)^{\sim}$	$1.8E+05$	30	$0.00E+00\ (0.00E+00)$	$9.2E+04$	30
f10	$1.45E-02\ (2.75E-03)^{+}$	NaN	0	$6.10E-15\ (6.49E-16)^{\sim}$	$6.7E+04$	30	$6.10E-15\ (6.49E-16)$	$5.0E+04$	30
f11	$1.17E-04\ (1.35E-04)^{-}$	NaN	0	$0.00E+00\ (0.00E+00)^{-}$	$5.0E+04$	30	$3.37E-03\ (4.64E-03)$	$2.8E+04$	19
f12	$6.97E-05\ (3.20E-05)^{+}$	NaN	0	$1.68E-31\ (1.43E-31)^{\sim}$	$4.3E+04$	30	$1.63E-25\ (8.95E-25)$	$2.7E+04$	30
f13	$5.10E-04\ (2.19E-04)^{+}$	NaN	0	$2.37E-30\ (2.69E-30)^{+}$	$4.7E+04$	30	$4.98E-32\ (4.19E-32)$	$2.7E+04$	30
F1	$1.57E-09\ (7.48E-10)^{+}$	$2.4E+05$	30	$0.00E+00\ (0.00E+00)^{-}$	$4.8E+04$	30	$2.69E-29\ (6.98E-29)$	$2.9E+04$	30
F2	$6.08E+02\ (1.92E+02)^{+}$	NaN	0	$1.08E+01\ (1.07E+01)^{+}$	NaN	0	$1.60E-12\ (1.57E-12)$	$1.6E+05$	30
F3	$8.26E+06\ (2.15E+06)^{+}$	NaN	0	$2.91E+06\ (1.01E+06)^{+}$	NaN	0	$9.00E+04\ (4.99E+04)$	NaN	0
F4	$2.28E+03\ (7.46E+02)^{+}$	NaN	0	$1.49E+02\ (8.27E+01)^{+}$	NaN	0	$6.70E-05\ (1.87E-04)$	$2.9E+05$	6
F5	$3.56E+02\ (9.42E+01)^{-}$	NaN	0	$1.17E+03\ (2.83E+02)^{+}$	NaN	0	$8.28E+02\ (3.40E+02)$	NaN	0
F6	$2.03E+01\ (1.74E+00)^{+}$	NaN	0	$2.89E+01\ (1.61E+01)^{+}$	NaN	0	$1.76E-09\ (6.58E-09)$	$1.9E+05$	30
F7	$3.52E-03\ (1.07E-02)^{-}$	$2.5E+05$	27	$7.64E-03\ (5.42E-03)^{-}$	$1.3E+05$	29	$1.47E-02\ (1.47E-02)$	$5.0E+04$	19
F8	$2.09E+01(5.72E-02)^{\sim}$	NaN	0	$2.09E+01\ (4.84E-02)^{\sim}$	NaN	0	$2.10E+01\ (4.54E-02)$	NaN	0
F9	$7.46E+01\ (1.03E+01)^{+}$	NaN	0	$0.00E+00\ (0.00E+00)^{\sim}$	$1.5E+05$	30	$0.00E+00\ (0.00E+00)$	$7.8E+04$	30
F10	$1.87E+02\ (1.13E+01)^{+}$	NaN	0	$7.95E+01\ (9.19E+00)^{+}$	NaN	0	$3.37E+01\ (9.16E+00)$	NaN	0
F11	$3.96E+01\ (9.53E-01)^{+}$	NaN	0	$3.09E+01\ (1.35E+00)^{+}$	NaN	0	$1.83E+01\ (8.49E+00)$	NaN	0
F12	$4.78E+03\ (4.49E+03)^{+}$	NaN	0	$4.25E+03\ (3.87E+03)^{+}$	NaN	0	$1.57E+03\ (1.87E+03)$	NaN	0
F13	$1.08E+01\ (7.91E-01)^{+}$	NaN	0	$2.78E+00\ (1.94E-01)^{-}$	NaN	0	$3.05E+00\ (1.88E-01)$	NaN	0
F14	$1.34E+01\ (1.33E-01)^{+}$	NaN	0	$1.29E+01\ (1.19E-01)^{+}$	NaN	0	$1.28E+01\ (3.00E-01)$	NaN	0
Better	22			14					
Similar	2			8					
Worse	3			5					

Note: "+", "−", and "∼" indicate that MLBBO is significantly better, worse, and indifferent, respectively, compared with other algorithms.

From the above results, we know that the total performance of MLBBO is significantly better than DE and BBO, which indicates that MLBBO not only integrates the exploration ability of DE and the exploitation ability of BBO simply, but also can balance both search abilities very well.

4.4. Comparisons with Improved BBO Algorithms. In this section, the performance of MLBBO is compared with improved BBO algorithms: (a) oppositional BBO (OBBO) [12], which is proposed by employing opposition-based learning (OBL) alongside BBO's migration rates; (b) multi-operator BBO (MOBBO) [18], which is another modified BBO with multi-operator migration operator; (c) perturb BBO (PBBO) [16], which is a modified BBO with perturb migration operator; and (d) real-code BBO (RCBBO) [9]. For fair comparison, five improved BBO algorithms are conducted on 27 functions with the same parameters setting above. The mean and standard deviation of fitness values are summarized in Table 3. Further, the results between MLBBO and OBBO, MLBBO and PBBO, MLBBO and MOBBO, and MLBBO and RCBBO are compared using statistical paired t-test. Convergence curves of mean fitness values under 30 independent runs are plotted in Figure 4.

From Table 3, we can see that MLBBO outperforms OBBO on 21 out of 27 test functions whereas OBBO surpasses MLBBO on 3 functions. MLBBO obtains similar fitness values with OBBO on 3 functions based on two tail t-test. Similar results can be obtained when making a comparison between MLBBO and RCBBO. When compared with PBBO and MOBBO, MLBBO is significantly better than both PBBO and MOBBO on 13 functions. MLBBO obtains similar results with PBBO and MOBBO on 10 and 11 functions, respectively. PBBO surpasses MLBBO only on 4 functions, which are

Quartic function, shifted rotated Ackley's function, expanded extended Griewank's plus Rosenbrock's Function, and shifted rotated expanded Scaffer's F6. MOBBO outperforms MLBBO only on 3 functions, which are Quartic function, shifted rotated Ackley's function, and expanded extended Griewank's plus Rosenbrock's function. Furthermore, the total performance of MLBBO is better than PBBO and MOBBO and much better than OBBO and RCBBO.

Furthermore, Figure 4 shows similar results with Table 3. We can see that the convergence speed of PBBO an MOBBO are faster than the others in preliminary stages owing to their powerful exploitation ability, but the convergence speed decreases in the next iteration. However, MLBBO can converge to satisfactory solutions at equilibrium rapidly speed. From the comparison, we know that operators used in MLBBO can enhance exploration ability.

4.5. Comparisons with Hybrid Algorithms. In order to validate the performance of the proposed algorithms (MLBBO) further, we compare the proposed algorithm with hybrid algorithms: OXDE [26] and DE/BBO [15]. The mean standard deviation of fitness values of 27 test functions are listed in Table 4. The average NFFEs and successful runs are also listed in Table 4. Convergence curves of mean fitness values under 30 independent runs are plotted in Figure 5.

From Table 4, we can see that results of MLBBO are significantly better than those of OXDE in 22 out of 27 high-dimensional functions, while the results of OXDE outperform those of MLBBO on only 3 functions. When compared with DE/BBO, MLBBO outperforms DE/BBO on 14 functions out of 27 test functions whereas DE/BBO surpasses MLBBO on 5 functions. For the rest 8 functions, there is no significant difference based on two tails t-test. MLBBO, DE/BBO, and OXDE obtain the VTR over all 30 successful runs within the MaxNFFEs for 16, 12, and 3 test functions, respectively. From this, we can come to a conclusion that MLBBO is more robust than DE/BBO and OXDE, which can be proved in Figure 5.

Further more, the average NFFEs MLBBO needed for the successful run is less than DE/BBO on most functions, which can also be indicate in Figure 5.

4.6. Effect of Dimensionality. In order to investigate the influence of scalability on the performance of MLBBO, we carry out a scalability study with DE, original BBO, DE/BBO [15], and aBBOmDE [14] for scalable functions. In the experimental study, the first 13 standard benchmark functions are studied at $Dim = 10, 50, 100$, and 200, and the other test functions are studied at $Dim = 10, 50$, and 100. For $Dim = 10$, the population size is equal to 50 and the population size is set to 100 for other dimensions in this paper. MaxNFFEs is set to $Dim * 10000$. All the functions are optimized over 30 independent runs in this paper. The average and standard deviation of fitness values of four compared algorithms are summarized in Table 5 for standard test functions f1–f13 and Table 6 for CEC 2005 test functions F1–F14. The convergence curves of mean fitness values under 30 independent runs are listed in Figure 6. Furthermore, the results of statistical paired t-test between MLBBO and others are also list in Tables 5 and 6.

It is to note that the results of DE/BBO [15] for standard functions f1–f13 are taken from corresponding paper under 50 independent runs. The results of aBBOmDE [14] for all the functions are taken from the paper directly, which are carried out with different population size (for $Dim = 10$, the population size equals 30, and for other Dimensions, it sets to Dim) under 50 independently runs.

For standard functions f1–f13, we can see from Table 5 that the overall results are decreasing for five compared algorithms since increasing the problem dimensions leads to algorithms that are sometimes unable to find the global optima solution within limit MaxNFFEs. Similar to $Dim = 30$, MLBBO outperforms DE on the majority of functions and overcomes BBO on almost all the functions at every dimension; MLBBO is better than or similar to DE/BBO on most of the functions.

When compared with aBBOmDE, we can see from Table 5 that MLBBO is better than or similar to aBBOmDE on most of the functions too. By carefully looking at results, we can recognize that results of some functions of MLBBO are worse than those of aBBOmDE in precision, but robustness of MLBBO is clearly better than aBBOmDE. For example, MLBBO and aBBOmDE find better solution for functions f8 and f9 when $Dim = 10$ and 50; aBBOmDE fails to solve functions f8 and f9 when Dim increases to 100 and 200, but MLBBO can solve functions f8 and f9 as before. This may be due to modified DE mutation being used in aBBOmDE as local search to accelerate the convergence speed, but modified DE mutation is used in MLBBO as a formula to supplement with migration operator and enhance the exploration ability of MLBBO.

For CEC 2005 test functions F1–F14, we can obtain similar results, from Table 6, that MLBBO is better than or similar to DE, BBO, and DE/BBO on most of functions when comparing MLBBO with these algorithms. However, results of MLBBO outperform aBBOmDE on most of the functions wherever $Dim = 10$ or $Dim = 50$.

From the comparison, we can arrive to a conclusion that the total performance of MLBBO is better than the others, especially, for CEC 2005 test functions.

Figure 7 shows the NFFEs required to reach VTR with different dimensions for 13 standard benchmark functions (f1–f13) and 6 CEC 2005 test functions (F1, F2, F4, F6, F7, and F9). From Figure 7, we can see that as dimension increases, the required NFFEs increases exponentially, which may be due to an exponentially increase in local minima with dimension. Meanwhile, if we compare Figures 7(a) and 7(b), we can find that CEC 2005 test functions need more NFFEs to reach VTR than standard benchmark functions do, which may be due to that CEC 2005 functions are more complicated than standard benchmark functions. For example, for function f3 (Schwefel 1.2), F2 (Shifted Schwefel 1.2), and F4 (Shifted Schwefel 1.2 with Noise in Fitness), F4, F2, and f3 need 140486.7, 157596.7, and 289960 MeanFEs to reach VTR (10^{-6}), respectively. Obviously, F4 is more complicated than f3 and F2; so, F4 needs 289960 MeanFEs to reach VTR, which is large than 140486.7 and 157596.

Table 5: Scalability study at $Dim = 10, 50, 100$, and 200 a ter $Dim * 10{,}000$ NFFEs for standard functions f1–f13.

	MLBBO Mean (std)	DE Mean (std)	BBO Mean (std)	DE/BBO Mean (std)	aBBOmDE Mean (std)
$dim = 10$					
f1	$1.55E-88$ $(6.53E-88)$	$8.83E-76$ $(1.89E-75)^{+}$	$2.14E-02$ $(2.04E-02)^{+}$	$0.00E+00$ $(0.00E+00)$	$1.49E-270$ $(0.00E+00)$
f2	$2.82E-46$ $(3.66E-46)$	$1.85E-36$ $(1.75E-36)^{+}$	$1.02E-01$ $(3.41E-02)^{+}$	$0.00E+00$ $(0.00E+00)$	—
f3	$6.37E-45$ $(2.80E-44)$	$8.54E-53$ $(1.67E-52)^{-}$	$3.29E+00$ $(2.02E+00)^{+}$	$6.07E-14$ $(9.60E-14)$	—
f4	$3.27E-31$ $(4.18E-31)$	$4.02E-29$ $(6.60E-29)^{+}$	$5.77E-01$ $(2.57E-01)^{-}$	$1.58E-16$ $(9.88E-17)$	—
f5	$1.05E-30$ $(4.30E-30)$	$0.00E+00$ $(0.00E+00)^{-}$	$1.43E+01$ $(1.09E+01)^{+}$	$3.40E+00$ $(8.03E-01)$	$8.26E+00$ $(1.69E+01)$
f6	$0.00E+00$ $(0.00E+00)$	$6.67E-02$ $(2.54E-01)^{-}$	$0.00E+00$ $(0.00E+00)^{\sim}$	$0.00E+00$ $(0.00E+00)$	—
f7	$9.78E-04$ $(7.39E-04)$	$1.20E-03$ $(8.19E-04)^{-}$	$6.52E-04$ $(7.36E-04)^{\sim}$	$1.11E-03$ $(3.96E-04)$	—
f8	$0.00E+00$ $(0.00E+00)$	$5.92E+01$ $(8.66E+01)^{+}$	$5.99E-02$ $(3.51E-02)^{+}$	$0.00E+00$ $(0.00E+00)$	$9.09E-14$ $(2.76E-13)$
f9	$0.00E+00$ $(0.00E+00)$	$7.75E+00$ $(6.62E+00)^{+}$	$1.20E-02$ $(1.26E-02)^{+}$	$0.00E+00$ $(0.00E+00)$	$0.00E+00$ $(0.00E+00)$
f10	$2.31E-15$ $(1.08E-15)$	$2.66E-15$ $(0.00E+00)^{-}$	$6.60E-02$ $(2.58E-02)^{+}$	$5.89E-16$ $(0.00E+00)$	$2.88E-15$ $(8.52E-16)$
f11	$3.94E-03$ $(6.86E-03)$	$7.43E-02$ $(5.45E-02)^{+}$	$9.48E-02$ $(3.40E-02)^{+}$	$0.00E+00$ $(0.00E+00)$	$4.48E-02$ $(2.26E-02)$
f12	$4.71E-32$ $(1.67E-47)$	$4.71E-32$ $(1.67E-47)^{-}$	$1.02E-03$ $(1.34E-03)^{+}$	$4.71E-32$ $(0.00E+00)$	$4.71E-32$ $(0.00E+00)$
f13	$1.35E-32$ $(5.57E-48)$	$2.85E-32$ $(8.23E-32)^{-}$	$4.01E-03$ $(4.29E-03)^{+}$	$1.35E-32$ $(0.00E+00)$	$1.35E-32$ $(0.00E+00)$
$dim = 50$					
f1	$2.47E-63$ $(2.84E-63)$	$4.42E-38$ $(6.97E-38)^{+}$	$1.30E-01$ $(3.46E-02)^{+}$	$0.00E+00$ $(0.00E+00)$	$3.3E-154$ $(1.4E-153)$
f2	$8.54E-35$ $(8.90E-35)$	$6.66E-19$ $(6.04E-19)^{+}$	$5.54E-01$ $(8.06E-02)^{+}$	$0.00E+00$ $(0.00E+00)$	—
f3	$1.23E-07$ $(1.49E-07)$	$3.73E-06$ $(6.23E-06)^{+}$	$7.87E+02$ $(1.76E+02)^{+}$	$5.20E-02$ $(2.48E-02)$	—
f4	$2.08E-02$ $(4.36E-03)$	$1.27E+00$ $(7.82E-01)^{+}$	$4.08E+00$ $(4.88E-01)^{+}$	$3.42E-03$ $(2.28E-02)$	—
f5	$2.25E-02$ $(5.80E-02)$	$1.06E+00$ $(1.79E+00)^{+}$	$1.44E+02$ $(4.15E+01)^{+}$	$4.43E+01$ $(1.39E+01)$	$9.50E+00$ $(2.69E+01)$
f6	$0.00E+00$ $(0.00E+00)$	$5.87E+00$ $(4.56E+00)^{+}$	$1.00E-01$ $(3.05E-01)^{\sim}$	$0.00E+00$ $(0.00E+00)$	—
f7	$5.07E-03$ $(1.31E-03)$	$1.53E-02$ $(5.01E-03)^{+}$	$2.91E-04$ $(2.60E-04)^{-}$	$4.05E-03$ $(9.20E-04)$	—
f8	$1.82E-11$ $(0.00E+00)$	$1.41E+04$ $(3.44E+02)^{+}$	$4.01E-01$ $(1.38E-01)^{+}$	$2.37E+00$ $(1.67E+01)$	$2.39E-11$ $(3.69E-12)$
f9	$0.00E+00$ $(0.00E+00)$	$3.55E+02$ $(1.98E+01)^{+}$	$7.01E-02$ $(2.29E-02)^{+}$	$0.00E+00$ $(0.00E+00)$	$4.43E-14$ $(4.78E-14)$
f10	$9.65E-15$ $(3.55E-15)$	$5.97E-11$ $(2.95E-10)^{-}$	$7.78E-02$ $(1.34E-02)^{+}$	$5.92E-15$ $(1.79E-15)$	$2.56E-14$ $(6.06E-15)$
f11	$1.31E-03$ $(4.51E-03)$	$5.01E-03$ $(6.35E-03)^{+}$	$1.55E-01$ $(4.64E-02)^{+}$	$0.00E+00$ $(0.00E+00)$	$2.78E-02$ $(3.42E-02)$
f12	$9.63E-33$ $(7.86E-34)$	$7.06E-02$ $(1.22E-01)^{+}$	$2.89E-04$ $(2.03E-04)^{+}$	$9.42E-33$ $(0.00E+00)$	$9.42E-33$ $(0.00E+00)$
f13	$1.37E-32$ $(9.00E-34)$	$1.05E-15$ $(5.77E-15)^{-}$	$7.82E-03$ $(5.78E-03)^{+}$	$1.35E-32$ $(0.00E+00)$	$1.35E-32$ $(0.00E+00)$
$dim = 100$					
f1	$6.53E-51$ $(9.15E-51)$	$1.02E-34$ $(1.06E-34)^{+}$	$2.89E-01$ $(6.31E-02)^{+}$	$6.16E-34$ $(1.97E-33)$	$1.18E-96$ $(3.26E-96)$
f2	$1.02E-25$ $(2.25E-25)$	$8.20E-19$ $(1.40E-18)^{+}$	$1.14E+00$ $(1.18E-01)^{+}$	$0.00E+00$ $(0.00E+00)$	—
f3	$1.37E-01$ $(5.93E-02)$	$5.70E+00$ $(1.93E+00)^{+}$	$2.93E+03$ $(4.86E+02)^{+}$	$3.10E-04$ $(1.12E-04)$	—
f4	$4.52E-01$ $(3.11E-01)$	$2.99E+01$ $(3.74E+00)^{+}$	$6.01E+00$ $(5.71E-01)^{+}$	$2.71E+00$ $(2.58E+00)$	—
f5	$3.89E+01$ $(4.22E+01)$	$9.25E+01$ $(4.78E+01)^{+}$	$3.22E+02$ $(6.74E+01)^{+}$	$1.19E+02$ $(3.38E+01)$	$2.57E+01$ $(4.08E+01)$
f6	$0.00E+00$ $(0.00E+00)$	$6.32E+01$ $(2.38E+01)^{+}$	$6.67E-02$ $(2.54E-01)^{\sim}$	$0.00E+00$ $(0.00E+00)$	—
f7	$1.80E-02$ $(3.88E-03)$	$5.49E-02$ $(1.35E-02)^{+}$	$1.92E-04$ $(2.20E-04)^{-}$	$6.87E-03$ $(1.15E-03)$	—
f8	$1.18E-10$ $(1.19E-11)$	$3.20E+04$ $(5.23E+02)^{+}$	$8.23E-01$ $(2.07E-01)^{+}$	$7.11E+00$ $(2.84E+01)$	$1.07E+02$ $(1.13E+02)$
f9	$2.61E-13$ $(2.66E-13)$	$8.25E+02$ $(4.72E+01)^{+}$	$1.37E-01$ $(3.12E-02)^{+}$	$7.36E-01$ $(8.48E-01)$	$1.59E-01$ $(3.68E-01)$
f10	$4.22E-14$ $(1.35E-14)$	$2.25E+00$ $(4.92E-01)^{+}$	$8.26E-02$ $(1.08E-02)^{+}$	$7.84E-15$ $(7.03E-16)$	$4.25E-14$ $(7.99E-15)$
f11	$5.40E-03$ $(1.22E-02)$	$3.69E-03$ $(5.92E-03)^{\sim}$	$1.92E-01$ $(4.99E-02)^{+}$	$0.00E+00$ $(0.00E+00)$	$5.10E-02$ $(8.89E-02)$
f12	$1.89E-32$ $(2.43E-32)$	$5.80E-01$ $(9.46E-01)^{+}$	$2.70E-04$ $(2.70E-04)^{+}$	$4.71E-33$ $(0.00E+00)$	$4.71E-33$ $(0.00E+00)$
f13	$3.37E-32$ $(2.21E-32)$	$1.15E+02$ $(3.29E+01)^{+}$	$1.08E-02$ $(3.98E-03)^{+}$	$1.76E-32$ $(2.68E-32)$	$1.55E-32$ $(5.98E-33)$
$dim = 200$					
f1	$8.01E-25$ $(1.23E-24)$	$8.65E-27$ $(2.72E-26)^{-}$	$7.98E-01$ $(1.33E-01)^{+}$	$3.07E-32$ $(2.48E-32)$	$5.70E-50$ $(7.34E-50)$
f2	$5.12E-05$ $(1.08E-04)$	$3.58E-14$ $(7.43E-14)^{-}$	$2.55E+00$ $(1.71E-01)^{+}$	$5.83E-17$ $(8.11E-17)$	—
f3	$6.39E+01$ $(1.05E+01)$	$8.28E+02$ $(1.97E+02)^{+}$	$9.04E+03$ $(7.12E+02)^{+}$	$2.22E+05$ $(5.90E+04)$	—
f4	$7.13E+00$ $(1.63E+00)$	$4.27E+01$ $(3.11E+00)^{+}$	$8.84E+00$ $(6.49E-01)^{+}$	$1.59E+01$ $(3.43E+00)$	—
f5	$3.09E+02$ $(6.35E+01)$	$3.23E+02$ $(8.13E+01)^{\sim}$	$6.29E+02$ $(6.26E+01)^{+}$	$2.95E+02$ $(4.48E+01)$	$2.16E+02$ $(9.98E+01)$
f6	$0.00E+00$ $(0.00E+00)$	$9.36E+02$ $(4.50E+02)^{+}$	$0.00E+00$ $(0.00E+00)^{\sim}$	$0.00E+00$ $(0.00E+00)$	—

Table 5: Continued.

	MLBBO	DE	BBO	DE/BBO	aBBOmDE
	Mean (std)	Mean (std)	Mean (std)	Mean (std)	Mean (std)
f7	$8.07E-02\ (1.24E-02)$	$2.11E-01\ (5.40E-02)^{+}$	$1.46E-04\ (1.06E-04)^{-}$	$1.56E-02\ (2.82E-03)$	—
f8	$1.14E-09\ (1.55E-09)$	$6.96E+04\ (6.12E+02)^{+}$	$2.13E+00\ (3.30E-01)^{+}$	$2.01E+02\ (1.68E+02)$	$3.08E+02\ (2.22E+02)$
f9	$9.75E-12\ (1.66E-11)$	$3.38E+02\ (2.06E+02)^{+}$	$3.25E-01\ (3.98E-02)^{+}$	$1.76E+01\ (2.89E+00)$	$1.19E+00\ (7.52E-01)$
f10	$5.30E-13\ (1.11E-12)$	$6.00E+00\ (1.09E+00)^{+}$	$9.92E-02\ (9.67E-03)^{+}$	$1.09E-14\ (1.12E-15)$	$1.09E-13\ (1.85E-14)$
f11	$5.29E-03\ (1.73E-02)$	$2.83E-02\ (7.57E-02)^{\sim}$	$2.88E-01\ (5.72E-02)^{+}$	$1.11E-16\ (0.00E+00)$	$5.54E-02\ (1.10E-01)$
f12	$1.90E-15\ (5.83E-15)$	$2.36E+00\ (2.35E+00)^{+}$	$3.09E-04\ (1.61E-04)^{+}$	$2.36E-33\ (0.00E+00)$	$2.36E-33\ (0.00E+00)$
f13	$2.25E-25\ (2.56E-25)$	$4.01E+02\ (5.11E+01)^{+}$	$3.17E-02\ (7.13E-03)^{+}$	$8.36E-32\ (1.44E-31)$	$1.35E-32\ (0.00E+00)$

Note: "+", "−", and "∼" indicate that MLBBO is significantly better, worse, and indifferent, respectively, compared with other algorithms.

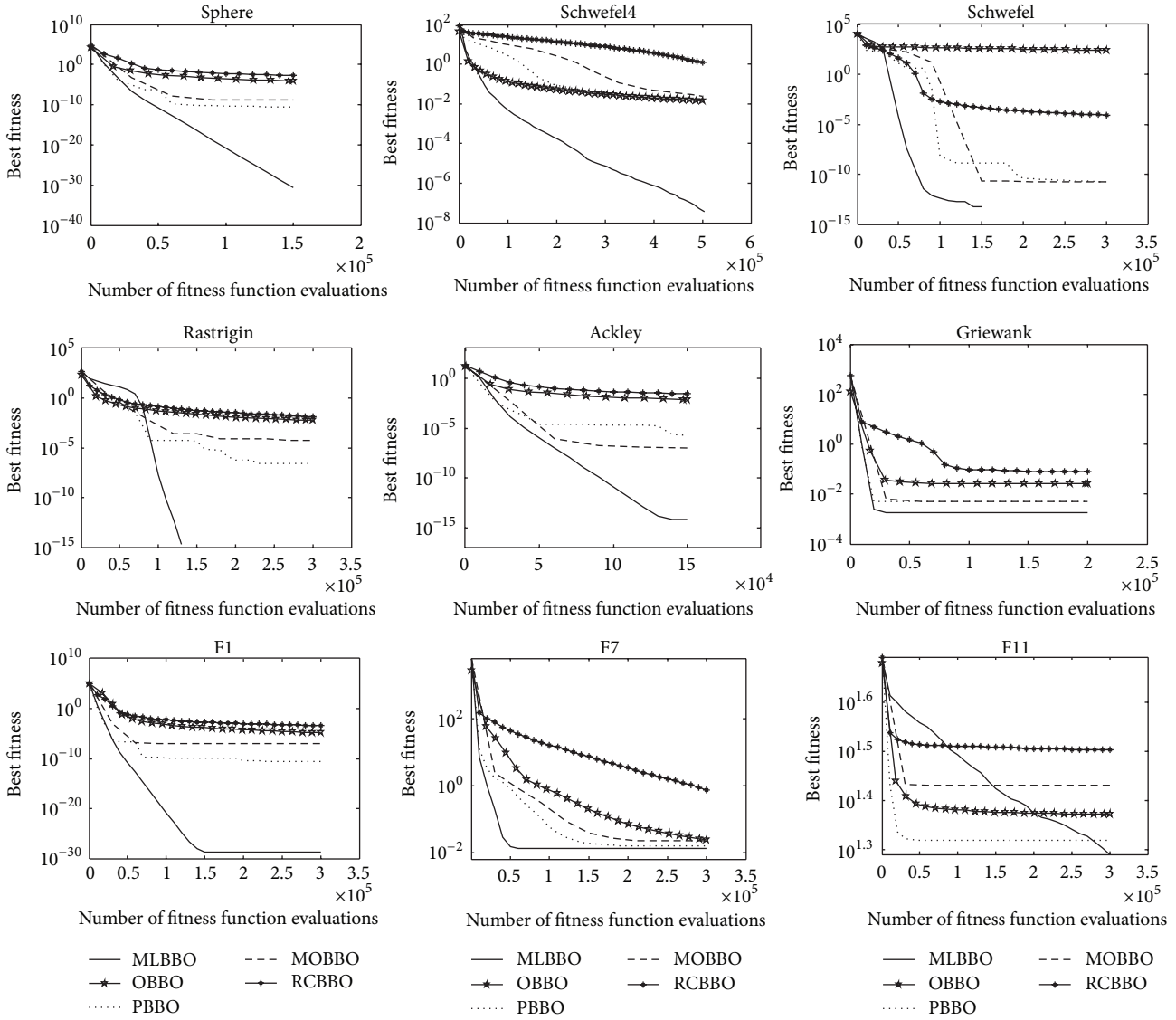


Figure 4: Convergence curves of mean fitness values of MLBBO, OBBO, PBBO, MOBBO, and RCBBO for functions f1, f4, f8, f9, f6, f7, f8, f9, f10, F1, F7, and F11.

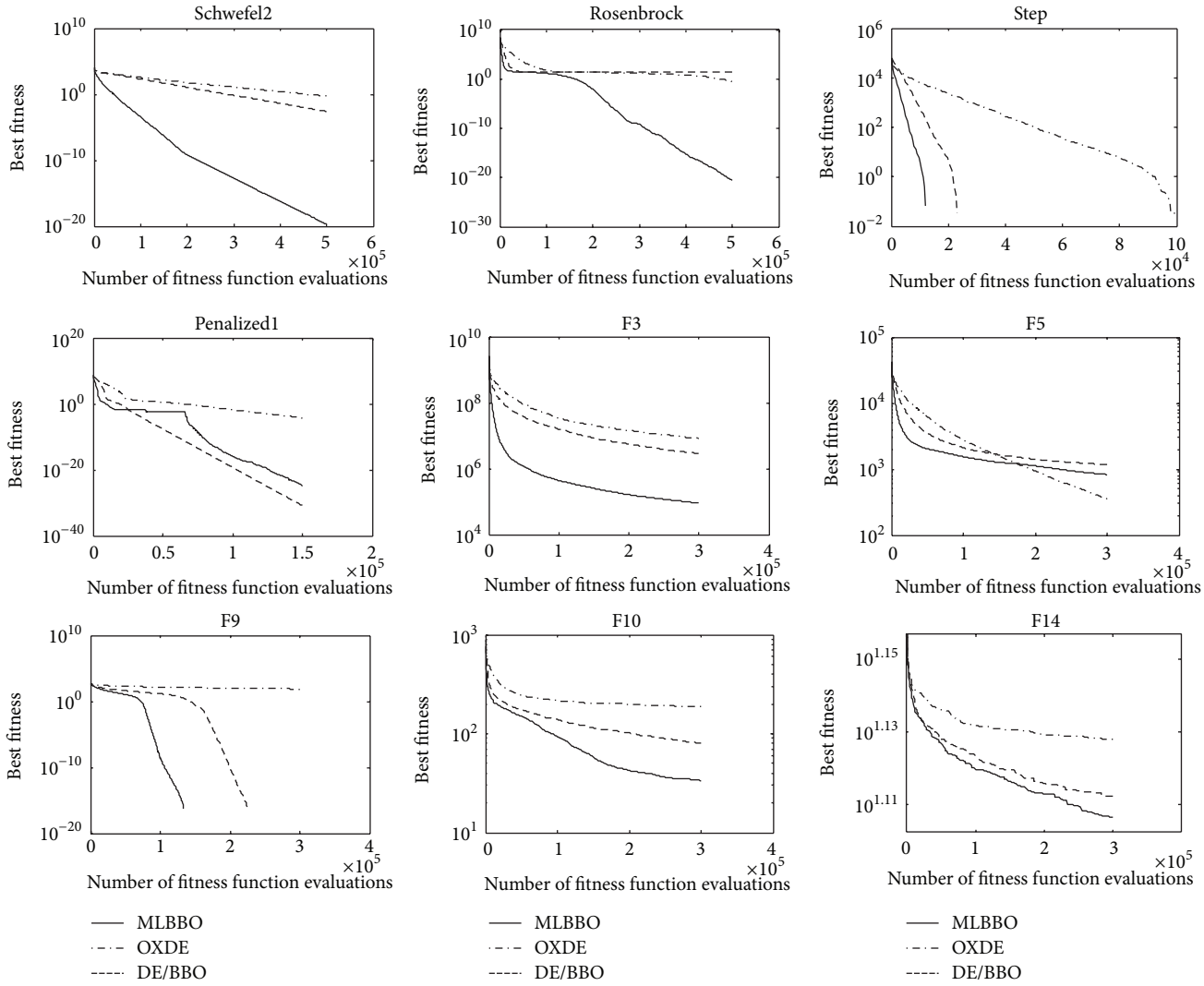


Figure 5: Convergence curves of mean fitness values of MLBBO, OXDE, and DE/BBO for functions f3, f5, f6, f12, F3, F5, F9, F10, and F14.

4.7. Analysis of the Performance of Operators in MLBBO. As the analysis above, the proposed variant of BBO algorithm has achieved excellent performance. In this section, we analyze the performance of modified migration operator and local search mechanism in MLBBO. In order to analyze the performance fairly and systematically, we compare the performance in four cases: MLBBO with both modified migration operator and local search leaning mechanism, MLBBO with the modified migration operator and without local search leaning mechanism, MLBBO without modified migration operator and with local search leaning mechanism, and MLBBO without both modified migration operator and local search leaning mechanism (it is actually BBO). Four algorithms are denoted as MLBBO, MLBBO2, MLBBO3, and MLBBO4. The experimental tests are conducted with 30 independent runs. The parameter settings are set in Section 4.1. The mean, standard deviation of fitness values of experimental test are summarized in Table 7. The numbers of successful runs are also recorded in Table 7. The results between MLBBO and MLBBO2, MLBBO and MLBBO3, and MLBBO and MLBBO4 are compared using two tails t-test. Convergence curves of mean fitness values under 30 independent runs are listed in Figure 8.

From Table 7, we can find that MLBBO is significantly better than MLBBO2 on 15 functions out of 27 tests functions, whereas MLBBO2 outperforms MLBBO on 5 functions. For the rest 7 functions, there are no significant differences based on two tails t-test. From this, we know that local search mechanism has played an important role in improving the search ability, especially, in improving the solution precision, which can be proved through careful looking at fitness values. For example, for function f1–f10, both algorithms MLBBO and MLBBO2 achieve the optima, but the fitness

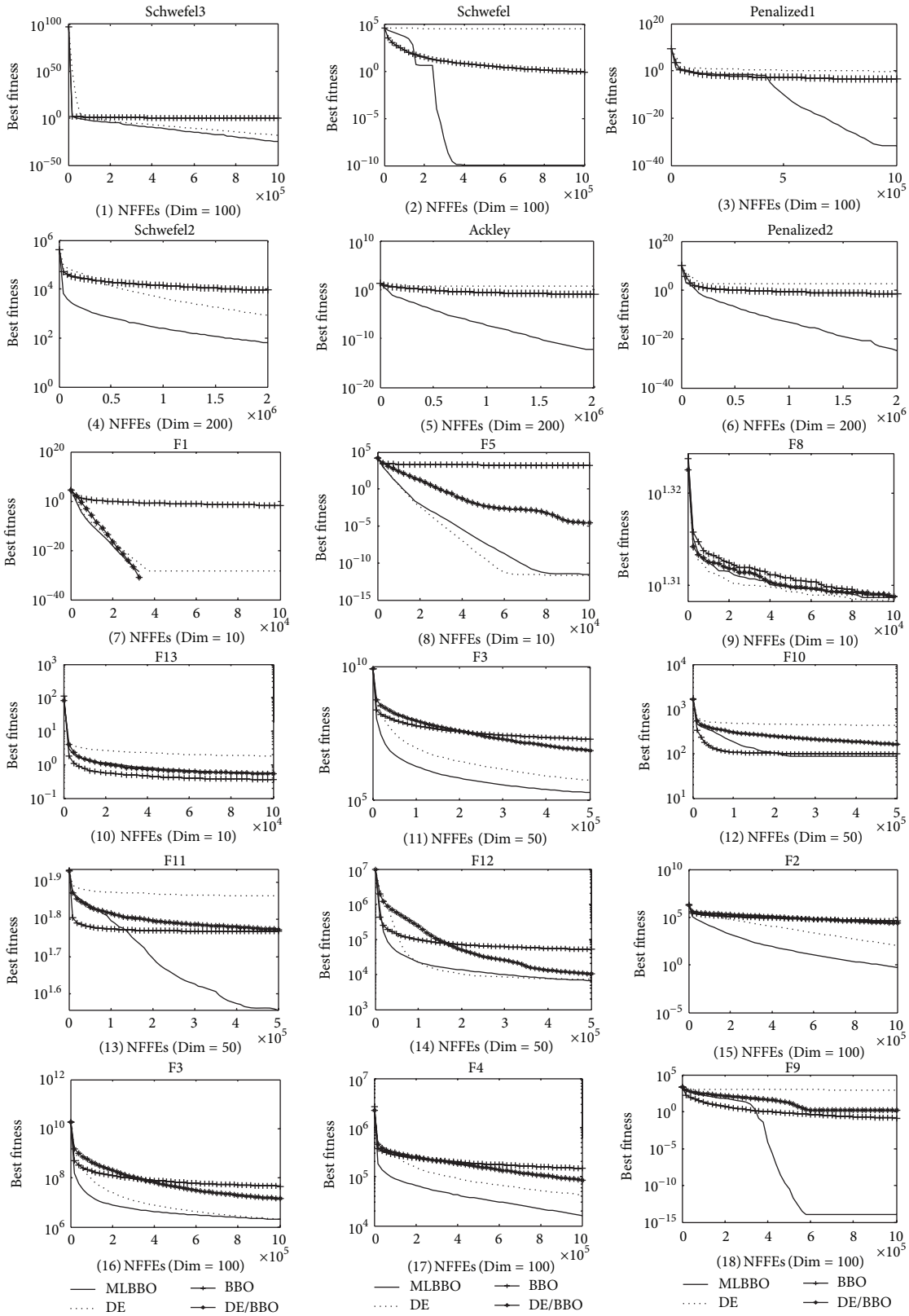
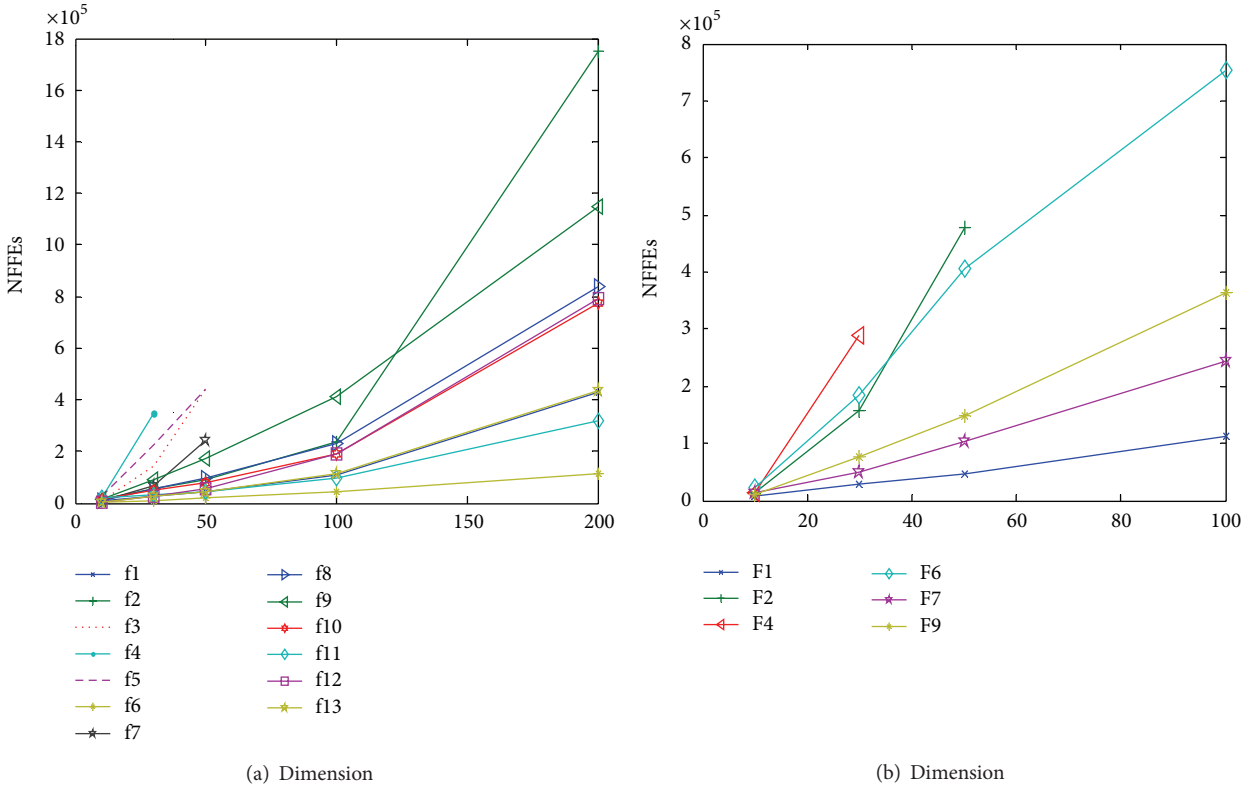


Figure 6: Mean fitness curves to compare the scalability of MLBBO, DE, BBO, and DE/BBO for selected functions: (1) f2 (*Dim* = 100), (2) f8 (*Dim* = 100), (3) f12 (*Dim* = 100), (4) f3 (*Dim* = 200), (5) f10 (*Dim* = 200), (6) f13 (*Dim* = 200), (7) F1 (*Dim* = 10), (8) F5 (*Dim* = 10), (9) F8 (*Dim* = 10), (10) F13 (*Dim* = 10), (11) F3 (*Dim* = 50), (12) F10 (*Dim* = 50), (13) F11 (*Dim* = 50), (14) F12 (*Dim* = 50), (15) F2 (*Dim* = 100), (16) F3 (*Dim* = 100), (17) F4 (*Dim* = 100), and(18) F9 (*Dim* = 100).



(a) Dimension (b) Dimension

Figure 7: NFFEs versus dimensions for (a) standard functions and (b) CEC 2005 functions.

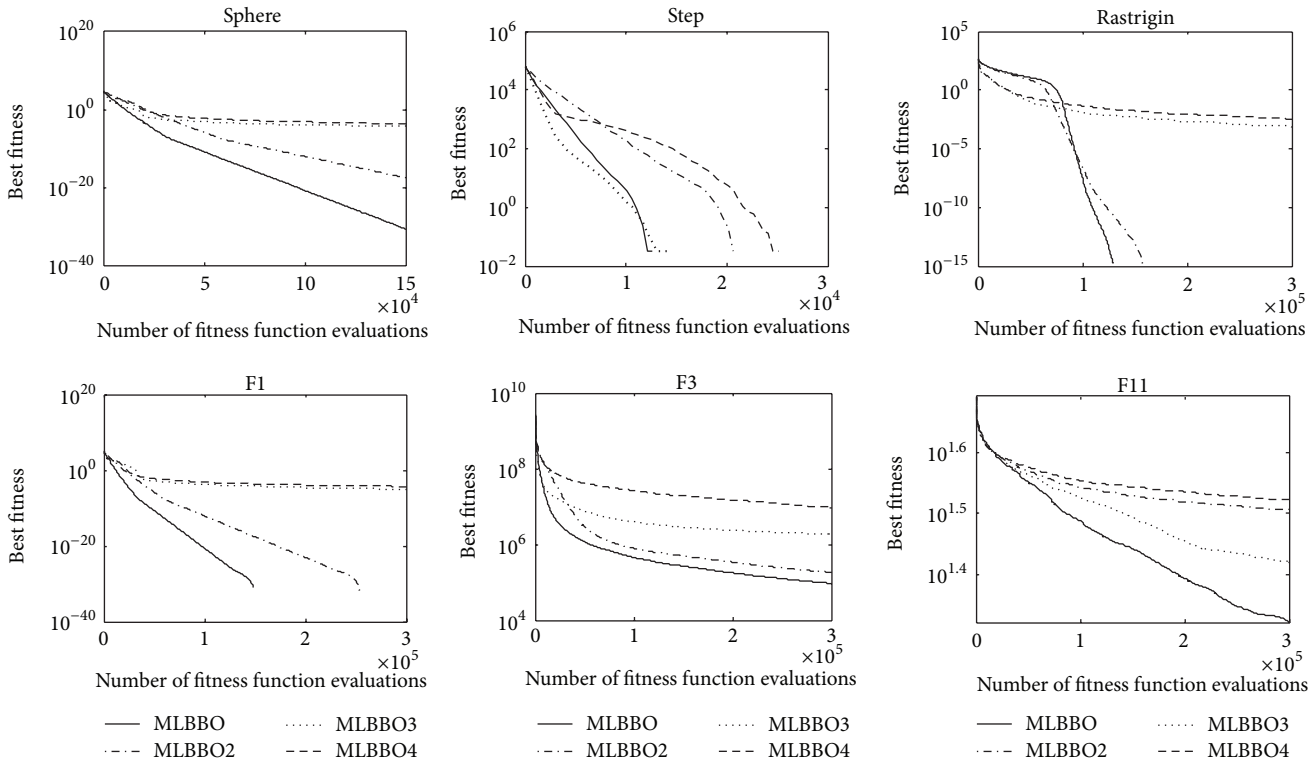


Figure 8: Mean fitness curves to analyze operators of MLBBO for selected functions f1, f6, f9, F1, F3, and F11.

Table 6: Scalability study at *Dim* = 10, 50, and 100 after *Dim* ∗ 10,000 *NFFEs* for CEC 2005 test functions F1–F14.

	MLBBO Mean(std)	DE Mean(std)	BBO Mean(std)	DE/BBO Mean(std)	aBBOmDE Mean(std)
dim = 10					
F1	$0.00E+00\,(0.00E+00)$	$4.71E-29\,(8.06E-29)^{+}$	$1.86E-02\,(9.76E-03)^{+}$	$0.00E+00\,(0.00E+00)^{\sim}$	$0.00E+00\,(0.00E+00)$
F2	$7.11E-29\,(7.42E-29)$	$6.31E-29\,(7.57E-29)^{\sim}$	$3.73E+01\,(3.75E+01)^{+}$	$8.50E-25\,(2.82E-24)^{\sim}$	$3.65E-23(2.58E-22)$
F3	$8.40E+01\,(4.19E+02)$	$8.05E-25(9.16E-25)^{\sim}$	$2.37E+06\,(3.01E+06)^{+}$	$1.86E+04\,(3.53E+04)^{+}$	$1.02E+05\,(9.77E+04)$
F4	$7.28E-29\,(9.57E-29)$	$3.91E-29\,(7.05E-29)^{\sim}$	$3.01E+02\,(3.31E+02)^{+}$	$2.27E-21\,(3.78E-21)^{+}$	$1.96E-02\,(6.19E-02)$
F5	$3.00E-12\,(5.10E-12)$	$1.97E-12\,(2.41E-12)^{\sim}$	$1.68E+03\,(1.09E+03)^{+}$	$2.88E-05\,(1.48E-04)^{+}$	$4.08E+02\,(6.93E+02)$
F6	$4.98E-26\,(1.94E-25)$	$1.32E-26\,(1.78E-26)^{\sim}$	$1.26E+02\,(1.35E+02)^{+}$	$4.10E+00\,(2.20E+00)^{+}$	$1.40E+02\,(8.22E+02)$
F7	$5.87E-02\,(4.35E-02)$	$1.35E-01\,(1.44E-01)^{+}$	$1.33E+00\,(4.04E-01)^{+}$	$3.86E-02\,(2.36E-02)^{-}$	$1.15E+00\,(1.08E+00)$
F8	$2.04E+01\,(7.80E-02)$	$2.03E+01\,(7.32E-02)^{\sim}$	$2.04E+01\,(1.25E-01)^{\sim}$	$2.04E+01\,(5.13E-02)^{\sim}$	$2.03E+01\,(8.35E-02)$
F9	$0.00E+00\,(0.00E+00)$	$8.39E+00\,(6.94E+00)^{+}$	$8.88E-03\,(4.51E-03)^{+}$	$0.00E+00\,(0.00E+00)^{\sim}$	$3.12E-09\,(2.21E-08)$
F10	$6.16E+00\,(2.97E+00)$	$2.29E+01\,(5.74E+00)^{+}$	$1.82E+01\,(8.69E+00)^{+}$	$5.98E+00\,(2.42E+00)^{\sim}$	$1.57E+01\,(6.04E+00)$
F11	$1.72E+00\,(1.59E+00)$	$2.48E+00\,(3.86E+00)^{\sim}$	$6.93E+00\,(1.43E+00)^{+}$	$8.28E-01\,(1.39E+00)^{-}$	—
F12	$3.47E+02\,(6.08E+02)$	$2.55E+02\,(5.30E+02)^{\sim}$	$7.23E+02\,(7.48E+02)^{+}$	$1.04E+02\,(3.40E+02)^{\sim}$	—
F13	$5.56E-01\,(7.44E-02)$	$1.81E+00\,(6.08E-01)^{+}$	$3.62E-01\,(1.35E-01)^{-}$	$5.36E-01\,(4.82E-02)^{\sim}$	—
F14	$2.42E+00\,(4.92E-01)$	$2.54E+00\,(3.02E-01)^{\sim}$	$3.59E+00\,(3.82E-01)^{+}$	$3.09E+00\,(1.76E-01)^{+}$	—
dim = 50					
F1	$8.25E-29\,(9.41E-29)$	$1.77E-27\,(5.83E-28)^{+}$	$1.18E-01\,(3.22E-02)^{+}$	$0.00E+00\,(0.00E+00)^{-}$	$0.00E+00\,(0.00E+00)$
F2	$9.11E-07\,(9.91E-07)$	$1.06E-04\,(1.75E-04)^{+}$	$1.25E+04\,(3.69E+03)^{+}$	$1.04E+03\,(3.04E+02)^{+}$	$2.62E-02\,(7.17E-02)$
F3	$1.86E+05\,(6.84E+04)$	$5.49E+05\,(2.01E+05)^{+}$	$1.93E+07\,(5.90E+06)^{+}$	$7.20E+06\,(2.38E+06)^{+}$	$1.33E+06\,(4.89E+05)$
F4	$1.73E+01\,(1.58E+01)$	$3.19E+02\,(2.52E+02)^{+}$	$4.36E+04\,(1.19E+04)^{+}$	$7.26E+03\,(2.17E+03)^{+}$	$1.29E+04\,(7.15E+03)$
F5	$4.18E+03\,(6.24E+02)$	$2.66E+03\,(5.09E+02)^{-}$	$1.16E+04\,(1.63E+03)^{+}$	$2.88E+03\,(4.22E+02)^{-}$	$1.40E+04\,(2.67E+03)$
F6	$2.88E+00\,(1.48E+01)$	$1.16E+00\,(1.83E+00)^{\sim}$	$3.05E+02\,(8.39E+01)^{+}$	$5.36E+01\,(2.14E+01)^{+}$	$4.06E+01\,(5.66E+01)$
F7	$1.41E-02\,(1.45E-02)$	$8.45E-03\,(1.06E-02)^{\sim}$	$1.23E+00\,(9.07E-02)^{+}$	$2.79E-03\,(6.55E-03)^{-}$	$1.15E-02\,(1.55E-02)$
F8	$2.11E+01\,(3.23E-02)$	$2.11E+01\,(3.37E-02)^{\sim}$	$2.10E+01\,(8.21E-02)^{-}$	$2.11E+01\,(4.21E-02)^{\sim}$	$2.11E+01\,(5.39E-02)$
F9	$4.74E-16\,(1.14E-15)$	$3.83E+02\,(3.57E+01)^{+}$	$6.67E-02\,(2.01E-02)^{+}$	$3.32E-02\,(1.82E-01)^{\sim}$	$2.22E-08\,(1.57E-07)$
F10	$8.66E+01\,(2.22E+01)$	$4.30E+02\,(3.36E+01)^{+}$	$9.98E+01\,(2.82E+01)^{+}$	$1.63E+02\,(1.35E+01)^{+}$	$1.48E+02\,(4.15E+01)$
F11	$3.60E+01\,(8.40E+00)$	$7.28E+01\,(1.86E+00)^{+}$	$5.85E+01\,(4.62E+00)^{+}$	$5.92E+01\,(1.44E+00)^{+}$	—
F12	$6.77E+03\,(5.32E+03)$	$7.37E+03\,(8.05E+03)^{\sim}$	$5.15E+04\,(2.61E+04)^{+}$	$1.05E+04\,(9.38E+03)^{\sim}$	—
F13	$5.56E+00\,(2.76E-01)$	$3.25E+01\,(2.06E+00)^{+}$	$1.86E+00\,(2.58E-01)^{-}$	$5.23E+00\,(2.88E-01)^{-}$	—
F14	$2.23E+01\,(3.53E-01)$	$2.30E+01\,(1.80E-01)^{+}$	$2.27E+01\,(3.76E-01)^{+}$	$2.26E+01\,(1.62E-01)^{+}$	—
dim = 100					
F1	$7.98E-28\,(1.15E-27)$	$6.80E-27\,(2.04E-27)^{+}$	$2.99E-01\,(6.26E-02)^{+}$	$1.68E-29\,(5.19E-29)^{-}$	—
F2	$5.61E-01\,(2.38E-01)$	$1.05E+02\,(4.54E+01)^{+}$	$3.98E+04\,(7.34E+03)^{+}$	$2.42E+04\,(5.08E+03)^{+}$	—
F3	$2.02E+06\,(5.35E+05)$	$2.06E+06\,(7.13E+05)^{\sim}$	$4.52E+07\,(9.83E+06)^{+}$	$1.42E+07\,(4.15E+06)^{+}$	—
F4	$1.63E+04\,(8.27E+03)$	$4.31E+04\,(1.13E+04)^{+}$	$1.48E+05\,(2.52E+04)^{+}$	$8.47E+04\,(1.16E+04)^{+}$	—
F5	$1.32E+04\,(1.40E+03)$	$8.11E+03\,(1.29E+03)^{-}$	$3.01E+04\,(3.32E+03)^{+}$	$6.08E+03\,(6.28E+02)^{-}$	—
F6	$3.98E+01\,(3.59E+01)$	$1.24E+02\,(5.05E+01)^{+}$	$4.74E+02\,(5.32E+01)^{+}$	$1.15E+02\,(2.81E+01)^{+}$	—
F7	$3.69E-03\,(6.49E-03)$	$4.02E-03\,(6.19E-03)^{\sim}$	$1.21E+00\,(6.38E-02)^{+}$	$1.48E-03\,(3.97E-03)^{\sim}$	—
F8	$2.13E+01\,(2.16E-02)$	$2.13E+01\,(2.32E-02)^{\sim}$	$2.11E+01\,(5.63E-02)^{-}$	$2.13E+01\,(2.05E-02)^{\sim}$	—
F9	$1.03E-14\,(4.70E-15)$	$7.80E+02\,(3.06E+02)^{+}$	$1.34E-01\,(3.10E-02)^{+}$	$1.56E+00\,(1.10E+00)^{+}$	—
F10	$2.96E+02\,(5.14E+01)$	$1.10E+03\,(4.50E+01)^{+}$	$2.74E+02\,(3.77E+01)^{\sim}$	$3.97E+02\,(2.50E+01)^{+}$	—

Note: "+", "–", and "~" indicate that MLBBO is significantly better, worse, and indifferent, respectively, compared with other algorithms.

values obtained by MLBBO are better than those obtained by MLBBO2 by several orders of magnitudes on these functions. When compared with MLBBO3, we can see that MLBBO outperforms MLBBO3 on 19 functions out of 27 test functions, while MLBBO is outperformed by MLBBO3 on only 2 functions. From this, we know that the modified migration operator has played a key role in optimizing the process and is better than original migration operator. Similar conclusion can also be obtained if we compare MLBBO2 and MLBBO4, MLBBO3 and MLBBO4.

Table 7: Analysis of the performance of operators in MLBBO.

	MLBBO		MLBBO2		MLBBO3		MLBBO4	
	Mean (std)	SR	Mean (std)	SR	Mean (std)	SR	Mean (std)	SR
f1	$2.03E-31\,(1.77E-31)$	30	$2.95E-18\,(1.05E-18)^{+}$	30	$4.53E-05\,(2.65E-05)^{+}$	0	$2.17E-04\,(7.92E-05)^{+}$	0
f2	$1.60E-21\,(8.48E-22)$	30	$4.40E-13\,(1.13E-13)^{+}$	30	$7.52E-03\,(2.74E-03)^{+}$	0	$3.64E-02\,(7.00E-03)^{+}$	0
f3	$1.47E-20\,(2.10E-20)$	30	$2.94E-12\,(1.94E-12)^{+}$	30	$1.88E+00\,(6.10E-01)^{+}$	0	$1.98E+00\,(5.63E-01)^{+}$	0
f4	$5.04E-08\,(9.88E-08)$	30	$3.04E-09\,(1.22E-09)^{-}$	30	$1.96E-02\,(4.67E-03)^{+}$	0	$2.32E-02\,(6.47E-03)^{+}$	0
f5	$1.02E-20\,(3.71E-20)$	30	$1.54E-14\,(1.06E-14)^{+}$	30	$4.79E+01\,(3.28E+01)^{+}$	0	$3.64E+01\,(3.58E+01)^{+}$	0
f6	$0.00E+00\,(0.00E+00)$	30	$0.00E+00\,(0.00E+00)^{\sim}$	30	$0.00E+00\,(0.00E+00)^{\sim}$	30	$0.00E+00\,(0.00E+00)^{\sim}$	30
f7	$2.63E-03\,(1.24E-03)$	30	$4.00E-03\,(7.68E-04)^{+}$	30	$3.25E-03\,(1.64E-03)^{\sim}$	30	$1.28E-02\,(5.07E-03)^{+}$	11
f8	$0.00E+00\,(0.00E+00)$	30	$0.00E+00\,(0.00E+00)^{\sim}$	30	$1.63E-06\,(1.25E-06)^{+}$	6	$7.92E-06\,(2.84E-06)^{+}$	0
f9	$0.00E+00\,(0.00E+00)$	30	$0.00E+00\,(0.00E+00)^{\sim}$	30	$7.93E-04\,(4.64E-04)^{+}$	0	$3.60E-03\,(1.46E-03)^{+}$	0
f10	$5.98E-15\,(9.01E-16)$	30	$4.48E-10\,(1.16E-10)^{+}$	30	$4.32E-03\,(1.24E-03)^{+}$	0	$9.72E-03\,(1.50E-03)^{+}$	0
f11	$3.70E-03\,(6.02E-03)$	19	$0.00E+00\,(0.00E+00)^{-}$	30	$1.07E-01\,(6.87E-02)^{+}$	1	$8.16E-02\,(5.71E-02)^{+}$	0
f12	$5.69E-27\,(3.07E-26)$	30	$4.41E-20\,(2.49E-20)^{+}$	30	$1.66E-06\,(5.43E-06)^{\sim}$	26	$7.68E-06\,(1.28E-05)^{+}$	1
f13	$5.79E-32\,(5.53E-32)$	30	$3.60E-19\,(1.65E-19)^{+}$	30	$3.33E-05\,(1.16E-04)^{\sim}$	0	$5.70E-05\,(5.07E-05)^{+}$	0
F1	$0.00E+00\,(0.00E+00)$	30	$0.00E+00\,(0.00E+00)^{\sim}$	30	$9.48E-06\,(8.28E-06)^{+}$	2	$4.66E-05\,(1.91E-05)^{+}$	0
F2	$2.37E-12\,(2.46E-12)$	30	$2.79E-06\,(2.44E-06)^{+}$	4	$4.10E+01\,(1.07E+01)^{+}$	0	$2.39E+01\,(1.01E+01)^{+}$	0
F3	$9.48E+04\,(5.14E+04)$	0	$1.86E+05\,(9.84E+04)^{+}$	0	$1.86E+06\,(5.75E+05)^{+}$	0	$9.51E+06\,(7.57E+06)^{+}$	0
F4	$1.21E-04\,(2.76E-04)$	4	$6.57E-03\,(8.98E-03)^{+}$	0	$1.12E+04\,(4.05E+03)^{+}$	0	$4.89E+03\,(1.77E+03)^{+}$	0
F5	$8.18E+02\,(3.25E+02)$	0	$1.27E+02\,(6.59E+01)^{-}$	0	$6.09E+03\,(1.12E+03)^{+}$	0	$4.47E+03\,(7.00E+02)^{+}$	0
F6	$1.93E-09\,(3.11E-09)$	30	$7.82E-04\,(4.05E-03)^{\sim}$	29	$9.26E+02\,(1.86E+03)^{+}$	0	$2.09E+03\,(4.48E+03)^{+}$	0
F7	$1.89E-02\,(1.73E-02)$	13	$1.56E-03\,(4.41E-03)^{-}$	29	$1.67E-01\,(2.06E-01)^{+}$	0	$7.76E-01\,(1.39E+00)^{+}$	0
F8	$2.10E+01\,(5.38E-02)$	0	$2.09E+01\,(6.06E-02)^{\sim}$	0	$2.09E+01\,(4.10E-02)^{\sim}$	0	$2.10E+01\,(3.80E-02)^{\sim}$	0
F9	$5.92E-17\,(3.24E-16)$	30	$0.00E+00\,(0.00E+00)^{\sim}$	30	$1.06E-03\,(6.71E-04)^{+}$	30	$3.44E-03\,(1.45E-03)^{+}$	30
F10	$3.76E+01\,(1.40E+01)$	0	$9.74E+01\,(6.54E+00)^{+}$	0	$3.79E+01\,(1.05E+01)^{\sim}$	0	$1.22E+02\,(8.19E+00)^{+}$	0
F11	$2.10E+01\,(7.37E+00)$	0	$3.21E+01\,(1.05E+00)^{+}$	0	$2.63E+01\,(4.96E+00)^{+}$	0	$3.34E+01\,(1.07E+00)^{+}$	0
F12	$2.07E+03\,(2.71E+03)$	0	$1.36E+04\,(1.97E+04)^{+}$	0	$1.26E+04\,(9.29E+03)^{+}$	0	$8.05E+03\,(7.03E+03)^{+}$	0
F13	$3.08E+00\,(1.60E-01)$	0	$2.77E+00\,(1.38E-01)^{-}$	0	$1.04E+00\,(1.64E-01)^{-}$	0	$9.82E-01\,(1.74E-01)^{-}$	0
F14	$1.28E+01\,(2.38E-01)$	0	$1.30E+01\,(1.62E-01)^{+}$	0	$1.25E+01\,(2.65E-01)^{-}$	0	$1.30E+01\,(1.86E-01)^{+}$	0
Better			15		19		24	
Similar			7		6		2	
Worse			5		2		1	

Note: "+", "−", and "∼" indicate that MLBBO is significantly better, worse, and indifferent, respectively, compared with other algorithms.

Furthermore, if we compare four algorithms together, the results of MLBBO4 are worse than those of MLBBO, especially, on multimodal functions, which means that the MLBBO has more powerful exploration ability than MLBBO4.

5. Conclusions and Future Work

In this paper, we have proposed a variant of modified BBO algorithm, called MLBBO, for solving global optimization problems. For origin BBO algorithm, the exploration ability is a barrier of performance of BBO. We suggest a modified migration operator by combining the formula in migration operator with a new one, which can absorb more information from other habitats and then enhance the exploration ability. Moreover, a local search mechanism is designed and used in modified BBO to supplement with modified migration operator. The proposed algorithm is compared with other improved BBO algorithms and evolutionary algorithms, which show that the proposed algorithm is superior to or at least highly competitive with the others.

During the analysis, we know that MLBBO possesses better performance, and we will investigate the performance of MLBBO in large-scale functions in the future work. We will also investigate the built-in relationship between the population diversity and exploration ability further.

6. Appendix

See Table 8.

Table 8: Benchmark functions used in our experimental tests.

Functions	Mathematical formula	Name	Dimension	Search space	Optima
f1	$f_1(x) = \sum_{i=1}^{n} x_i^2$	Sphere	n	$[-100, 100]$	0
f2	$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	Schwefel's Problem 2.22	n	$[10, 10]$	0
f3	$f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	Schwefel's Problem 1.2	n	$[-100, 100]$	0
f4	$f_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	Schwefel's Problem 2.21	n	$[-100, 100]$	0
f5	$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	Rosenbrock function	n	$[-30, 30]$	0
f6	$f_6(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	Step function	n	$[-100, 100]$	0
f7	$f_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	Quartic function	n	$[-1.28, 1.28]$	0
f8	$f_8(x) = -\sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})$	Schwefel function	n	$[-512, 512]$	-12569.5
f9	$f_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	Rastrigin function	n	$[-5.12, 5.12]$	0
f10	$f_{10}(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos 2\pi x_i\right) + 20 + e$	Ackley function	n	$[-32, 32]$	0
f11	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	Griewank function	n	$[-600, 600]$	0
f12	$f_{12}(x) = \frac{\pi}{n}\{10\sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$, $y_i = 1 + \frac{(x_i - 1)}{4}$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^2, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^2, & x_i < -a \end{cases}$	Penalized function 1	n	$[-50, 50]$	0

Table 8: Continued.

Functions	Mathematical formula	Name	Dimension	Search space	Optima
f13	$f_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4),$	Penalized function 2	n	$[-50, 50]$	0
F1	$F_1 = \sum_{i=1}^{n} z_i^2 + f\_\text{bias}_1, Z = X - O$	Shifted sphere function	n	$[-100, 100]$	−450
F2	$F_2 = \sum_{i=1}^{n}(\sum_{j=1}^{i} z_j)^2 + f\_\text{bias}_2, Z = X - O$	Shifted Schwefel's Problem 1.2	n	$[-100, 100]$	−450
F3	$F_3 = \sum_{i=1}^{n}(10^6)^{(i-1)/(D-1)} z_i^2 + f\_\text{bias}_3, Z = X - O$	Shifted rotated high conditioned elliptic function	n	$[-100, 100]$	−450
F4	$F_4 = \left(\sum_{i=1}^{n}\left(\sum_{j=1}^{i} z_j\right)^2\right) * (1 + 0.4\,\lvert N(0,1)\rvert) + f\_\text{bias}_4, Z = X - O$	Shifted Schwefel's Problem 1.2 with noise in fitness	n	$[-100, 100]$	−450
F5	$F_5 = \max\{\lvert A_i x - B_i\rvert\} + f\_\text{bias}_5$	Schwefel's Problem 2.6 with global optimum on bounds	n	$[-30, 30]$	−310
F6	$F_6 = \sum_{i=1}^{n-1}[100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2] + f\_\text{bias}_6, Z = X - O + 1$	Shifted rosenbrock's function	n	$[-100, 100]$	390

Table 8: Continued.

Functions	Mathematical formula	Name	Dimension	Search space	Optima
F7	$F_7 = \frac{1}{4000}\sum_{i=1}^{n} z_i^2 - \prod_{i=1}^{n} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f\_\text{bias}_7, Z = (X - O) * M$	Shifted rotated Griewank's function without bounds	n	Outside of initialization range [0, 600]	−180
F8	$F_8 = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi z_i\right) + 20 + e + f\_\text{bias}_8, Z = (X - O) * M$	Shifted rotated Ackley's function with global optimum on bounds	n	[−32, 32]	−140
F9	$F_9 = \sum_{i=1}^{n}[z_i^2 - 10\cos(2\pi z_i) + 10] + f\_\text{bias}_8, Z = X - O$	Shifted Rastrigin's function	n	[−5.12, 5.12]	−330
F10	$F_9 = \sum_{i=1}^{n}[z_i^2 - 10\cos(2\pi z_i) + 10] + f\_\text{bias}_8, Z = (X - O) * M$	Shifted rotated Rastrigin's function	n	[−5.12, 5.12]	−330
F11	$F_{11} = \sum_{i=1}^{n}\left(\sum_{k=0}^{k\max}\left[a^k\cos(2\pi b^k(z_i + 0.5))\right]\right) - n\sum_{k=0}^{k\max}\left[a^k\cos(2\pi b^k \cdot 0.5)\right] + f\_\text{bias}_{11}$ $a = 0.5, b = 3, k\max = 20, Z = (X - O) * M$	Shifted rotated weierstrass Function	n	[−600, 600]	90
F12	$F_{12} = \sum_{i=1}^{n}(A_i - B_i(x))^2 + f\_\text{bias}_{12}$ $A_i = \sum_{i=1}^{n}\left(a_{ij}\sin\alpha_j + b_{ij}\cos\alpha_j\right), B_i = \sum_{i=1}^{n}\left(aij\sin x_j + bij\cos x_j\right)$	Schwefel's Problem 2.13	n	[−100, 100]	−460
F13	$F_{13} = f_{11}(f_5(z_1, z_2)) + f_{11}(f_5(z_2, z_3)) + \cdots + f_{11}(f_5(z_{n-1}, z_n)) + f_{11}(f_5(z_n, z_1)) + f\_\text{bias}_{13}$ $Z = X - O + 1$	Expanded extended Griewank's plus Rosenbrock's function (F8F2)	n	[−50, 50]	−130
F14	$F(x, y) = 0.5 + \frac{\sin^2\left(\sqrt{x^2 + y^2}\right) - 0.5}{\left(1 + 0.001\left(x^2 + y^2\right)\right)^2}$ $F_{14} = F(z_1, z_2) + F(z_2, z_3) + \cdots + F(z_{n-1}, z_n) + F(z_n, z_1) + f\_\text{bias}_{14}, Z = (X - O) * M$	Shifted rotated expanded Scaffer's F6	n	[−100, 100]	−300

Conflict of Interests

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

Acknowledgments

The author would like to express thanks to the area editor and anonymous reviewers for their valuable comments and suggestions for this paper. This work is proudly supported in part by the National Natural Science Foundation of China (No. 11101101, No. 11226219, No. 61373174, and No. 11361019), Natural Science Foundation of Guangxi (No. 2013GXNSFBA019008 and No. 2013GXNSFAA019003), Science and Technology Plan Project of Science and Technology Department of Hunan (No. 2013FJ3083), and Project supported by Program to Sponsor Teams for Innovation in the Construction of Talent Highlands in Guangxi Institutions of Higher Learning ([2011] 47).

References

[1] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: a survey," *Operations Research*, vol. 14, pp. 699–719, 1966.

[2] N. Andrei, "Accelerated scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization," *European Journal of Operational Research*, vol. 204, no. 3, pp. 410–420, 2010.

[3] I. Ciornei and E. Kyriakides, "Hybrid ant colony-genetic algorithm (GAAPI) for global continuous optimization," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 42, no. 1, pp. 234–245, 2012.

[4] G. Chen, C. P. Low, and Z. Yang, "Preserving and exploiting genetic diversity in evolutionary programming algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 661–673, 2009.

[5] C. Li, S. Yang, and T. T. Nguyen, "A self-learning particle swarm optimizer for global optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 42, no. 3, pp. 627–646, 2012.

[6] S. L. Ho, S. Yang, Y. Bai, and J. Huang, "An ant colony algorithm for both robust and global optimizations of inverse problems," *IEEE Transactions on Magnetics*, vol. 49, no. 5, pp. 2077–2088, 2013.

[7] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[8] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[9] W. Gong, Z. Cai, C. X. Ling, and H. Li, "A real-coded biogeography-based optimization with mutation," *Applied Mathematics and Computation*, vol. 216, no. 9, pp. 2749–2758, 2010.

[10] Z. Cai, W. Gong, and C. X. Ling, "Research on a novel biogeography-based optimization algorithm based on evolutionary programming," *System Engineering Theory and Practice*, vol. 30, no. 6, pp. 1106–1112, 2010.

[11] I. Boussaïd, A. Chatterjee, P. Siarry, and M. Ahmed-Nacer, "Two-stage update biogeography-based optimization using differential evolution algorithm (DBBO)," *Computers and Operations Research*, vol. 38, no. 8, pp. 1188–1198, 2011.

[12] M. Ergezer, D. Simon, and D. Du, "Oppositional biogeography-based optimization," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '09)*, pp. 1009–1014, San Antonio, Tex, USA, October 2009.

[13] H. Ma, M. Fei, D. Simon, and M. Yu, "Biogeography-based optimization in noisy environments," Technique Report, 2012, http://academic.csuohio.edu/simond/bbo/noisy/.

[14] M. R. Lohokare, S. S. Pattnaik, B. K. Panigrahi, and S. Das, "Accelerated biogeography-based optimization with neighborhood search for optimization," *Applied Soft Computing*, vol. 13, no. 5, pp. 2318–2342, 2013.

[15] W. Gong, Z. Cai, and C. X. Ling, "DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization," *Soft Computing*, vol. 15, no. 4, pp. 645–665, 2011.

[16] X. Li, J. Wang, J. Zhou, and M. Yin, "A perturb biogeography based optimization with mutation for global numerical optimization," *Applied Mathematics and Computation*, vol. 218, no. 2, pp. 598–609, 2011.

[17] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Multi-operator based evolutionary algorithms for solving constrained optimization problems," *Computers and Operations Research*, vol. 38, no. 12, pp. 1877–1896, 2011.

[18] X. Li and M. Yin, "Multi-operator based biogeography based optimization with mutation for global numerical optimization," *Computers & Mathematics with Applications*, vol. 64, no. 9, pp. 2833–2844, 2012.

[19] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166–3173, 2010.

[20] A. Hastings and K. Higgins, "Persistence of transients in spatially structured ecological models," *Science*, vol. 263, no. 5150, pp. 1133–1136, 1994.

[21] H. Ma, "An analysis of the equilibrium of migration models for biogeography-based optimization," *Information Sciences*, vol. 180, no. 18, pp. 3444–3464, 2010.

[22] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.

[23] P. N. Suganthan, N. Hansen, J. J. Liang et al., "Problem definition and evaluation criteria for the CEC, 2005 special session on real parameter optimization," Technical Report, Nanyang Technological University and KanGAL Report 2005005, IIT Kanpur, 2005, http://www.ntu.edu.sg/home/epnsugan.

[24] C. H. Goulden, *Methods of Statistical Analysis*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1956.

[25] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.

[26] Y. Wang, Z. Cai, and Q. Zhang, "Enhancing the search ability of differential evolution through orthogonal crossover," *Information Sciences*, vol. 185, no. 1, pp. 153–177, 2012.