

Research Article

Heterogeneous Differential Evolution for Numerical Optimization

Hui Wang,¹ Wenjun Wang,² Zhihua Cui,^{3,4} Hui Sun,¹ and Shahryar Rahnamayan⁵

¹ School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China

² School of Business Administration, Nanchang Institute of Technology, Nanchang 330099, China

³ Complex System and Computational Intelligent Laboratory, Taiyuan University of Science and Technology, Taiyuan 030024, China

⁴ State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

⁵ Department of Electrical, Computer and Software Engineering, University of Ontario Institute of Technology, 2000 Simcoe Street North Oshawa, ON, Canada L1H 7K4

Correspondence should be addressed to Hui Wang; huiwang@whu.edu.cn

Received 28 September 2013; Accepted 23 December 2013; Published 5 February 2014

Academic Editors: G. C. Gini and J. Zhang

Copyright © 2014 Hui Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Differential evolution (DE) is a population-based stochastic search algorithm which has shown a good performance in solving many benchmarks and real-world optimization problems. Individuals in the standard DE, and most of its modifications, exhibit the same search characteristics because of the use of the same DE scheme. This paper proposes a simple and effective heterogeneous DE (HDE) to balance exploration and exploitation. In HDE, individuals are allowed to follow different search behaviors randomly selected from a DE scheme pool. Experiments are conducted on a comprehensive set of benchmark functions, including classical problems and shifted large-scale problems. The results show that heterogeneous DE achieves promising performance on a majority of the test problems.

1. Introduction

Differential evolution (DE) [1] is a well-known algorithm for global optimization over continuous search spaces. Although DE has shown a good performance over many optimization problems, its performance is greatly influenced by its mutation scheme and control parameters (population size, scale factor, and crossover rate). To enhance the performance of DE, many improved DE variants have been proposed based on modified mutation strategies [2, 3] or adaptive parameter control [4–6].

The standard DE and most of its modifications [7–10] make use of homogeneous populations where all of the individuals follow exactly the same behavior. That is, individuals implement the same DE scheme, such as DE/rand/1/bin and DE/best/1/bin. The effect is that individuals in the population behave with the same exploration and/or exploitation characteristics [11]. An ideal optimization algorithm should balance exploration and exploitation during the search process. Initially, the algorithm should concentrate on exploration. As

the iteration increases, it would be better to use exploitation to find more accurate solutions. However, it is difficult to determine when the algorithm should switch from an explorative behavior to an exploitative behavior. To tackle this problem, a new concept of heterogeneous swarms [11] is proposed and applied to particle swarm optimization (PSO) [12, 13], where particles in the swarm use different velocity and position update rules. Therefore, the swarm may consist of explorative particles as well as exploitative particles. This makes the heterogeneous PSO have the ability to balance exploration and exploitation during the search process.

In this paper, a simple and effective heterogeneous DE (HDE) algorithm is proposed inspired by the idea of heterogeneous swarms [11]. In HDE, individuals will be allocated to different search behaviors randomly selected from a DE scheme pool. However, the concept of heterogeneous swarms used in DE is not new. Qin et al. [6] proposed a self-adaptive DE (SaDE), where individuals are also allowed to implement different mutation schemes according to a complex probability model. Gong et al. [14] combined

a strategy adaptation mechanism with four mutation strategies proposed in JADE [3]. For other self-adaptive DE variants [8], the search characteristics of individuals dynamically change according to the adaptation of the control parameters (scale factor and/or crossover rate). The HDE proposed in this paper differs from the above DE variants. In HDE, the behaviors are randomly assigned from a DE scheme pool. By the suggestions of heterogeneous PSO [11], two heterogeneous models are proposed. This first one is static HDE (sHDE), where the randomly selected behaviors are fixed during the evolution. The second one is dynamic HDE (dHDE), where the behaviors can randomly change during the search process. Experimental studies are conducted on a comprehensive set of benchmark functions, including classical problems and shifted large-scale problems. Simulation results demonstrate the efficiency and effectiveness of the proposed heterogeneous DE.

The rest of the paper is organized as follows. In Section 2, the standard DE algorithm is briefly introduced. The HDE is proposed in Section 3. Experimental simulations, results, and discussions are presented in Section 4. Finally, the work is concluded in Section 5.

2. Differential Evolution

There are several variants of DE [1], which use different mutation strategies and/or crossover schemes. To distinguish these different DE schemes, the notation “DE/*x/y/z*” is used, where “DE” indicates the DE algorithm, “*x*” denotes the vector to be mutated, “*y*” is the number of difference vectors used in the mutation, and “*z*” stands for the type of crossover scheme, exponential (exp) or binomial (bin). The following section discusses the mutation and crossover operations.

2.1. Mutation. For each vector $X_{i,G}$ at generation G , this operation creates mutant vectors $V_{i,G}$ based on the current parent population. The following are five well-known variant mutation strategies:

(1) DE/rand/1

$$V_{i,G} = X_{i_1,G} + F \cdot (X_{i_2,G} - X_{i_3,G}), \quad (1)$$

(2) DE/best/1

$$V_{i,G} = X_{\text{best},G} + F \cdot (X_{i_1,G} - X_{i_2,G}), \quad (2)$$

(3) DE/current-to-best/1

$$V_{i,G} = X_{i,G} + F \cdot (X_{\text{best},G} - X_{i,G}) + F \cdot (X_{i_1,G} - X_{i_2,G}), \quad (3)$$

(4) DE/rand/2

$$V_{i,G} = X_{i_1,G} + F \cdot (X_{i_2,G} - X_{i_3,G}) + F \cdot (X_{i_4,G} - X_{i_5,G}), \quad (4)$$

(5) DE/best/2

$$V_{i,G} = X_{\text{best},G} + F \cdot (X_{i_1,G} - X_{i_2,G}) + F \cdot (X_{i_3,G} - X_{i_4,G}), \quad (5)$$

where the indices $i_1, i_2, i_3, i_4,$ and i_5 are mutually different random indices chosen from the set $\{1, 2, \dots, N_p\}$, N_p is the population size, and all are different from the base index i . The scale factor F is a real number that controls the difference vectors. $X_{\text{best},G}$ is the best vector in terms of fitness value at the current generation G .

2.2. Crossover. Similar to EAs, DE also employs a crossover operator to build trial vectors $U_{i,G}$ by recombining the current vector $X_{i,G}$ and the mutant one $V_{i,G}$. The trial vector is defined as follows:

$$u_{i,j,G} = \begin{cases} v_{i,j,G}, & \text{if } \text{rand}_j(0, 1) \leq \text{CR} \vee j = j_{\text{rand}} \\ x_{i,j,G}, & \text{otherwise,} \end{cases} \quad (6)$$

where $\text{CR} \in (0, 1)$ is the predefined crossover probability, $\text{rand}_j(0, 1)$ is a uniform random number within $[0, 1]$ for the j th dimension, and $j_{\text{rand}} \in \{1, 2, \dots, D\}$ is a random index.

2.3. Selection. After the crossover, a greedy selection mechanism is used to select the better one from the parent vector $X_{i,G}$ and the trial vector $U_{i,G}$ according to their fitness values $f(\cdot)$. Without losing of generality, this paper only considers minimization problems. If, and only if, the trial vector $U_{i,G}$ is better than the parent vector $X_{i,G}$, then $X_{i,G+1}$ is set to $U_{i,G}$; otherwise, we keep $X_{i,G+1}$ the same with $X_{i,G}$:

$$X_{i,G+1} = \begin{cases} U_{i,G}, & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G}, & \text{otherwise.} \end{cases} \quad (7)$$

3. Heterogeneous DE

In the standard DE and most of its modifications, individuals in the population behave with the same search characteristics, exploration, and/or exploitation, because of the use of the same DE scheme. The effect is that the algorithms could hardly balance exploration and exploitation during the search process. Inspired by the heterogeneous swarms [11], a simple and effective heterogeneous DE (HDE) algorithm is proposed in this paper. Compared to DE and most of its variants, individuals in HDE are allowed to implement different behaviors.

To implement different DE schemes in HDE, we need to address two questions. First, which DE scheme should be chosen to construct the DE scheme pool? Second, how do we assign DE schemes to individuals?

As mentioned before, there are several DE schemes, and different DE schemes have different search characteristics. In this paper, three different DE schemes are chosen to construct the DE scheme pool: (1) DE/rand/1/bin; (2) DE/best/1/bin; and (3) DE/BoR/1/bin [16]. The first two schemes are two basic DE strategies proposed in [1], and the last one was recently proposed in [16], where a new mutation strategy called “best-of-random” (BoR) is defined as follows:

$$V_{i,G} = X_{i_b,G} + F \cdot (X_{i_1,G} - X_{i_2,G}), \quad (8)$$

where the individuals $X_{i_1}, X_{i_2},$ and X_{i_b} are randomly chosen from the current population, $i \neq i_1 \neq i_2 \neq i_b$, and X_{i_b} is the best one of them; that is, $f(X_{i_b}) \leq \min(f(X_{i_1}), f(X_{i_2}))$.

Variables	DE scheme
$X_{1,G}$	ID ₁
$X_{2,G}$	ID ₂
$X_{3,G}$	ID ₃
...	...
$X_{N_p,G}$	ID _{N_p}

FIGURE 1: The encoding of individuals in heterogeneous DE.

There are two reasons for choosing these DE schemes. First, these three DE schemes are very simple and easy to implement. Second, each of them has different search characteristics. For the DE/rand/1/bin, it obtains higher population diversity. DE/best/1/bin shows faster convergence speed. The last one provides a middle phase between the first two DE schemes. Note that this paper only chooses three DE schemes to construct the DE scheme pool and other DE schemes can also be possibly used.

Figure 1 presents the encoding method in HDE, where ID_{*i*} denotes the employed DE scheme for the *i*th individual. In this paper, ID_{*i*} ∈ {1, 2, 3}; that is, ID_{*i*} = 1 indicates the DE/rand/1/bin scheme, ID_{*i*} = 2 denotes the DE/best/1/bin scheme, and ID_{*i*} = 3 stands for the DE/BoR/1/bin scheme.

In order to address the second question, two different heterogeneous models, namely, static HDE (sHDE) and dynamic HDE (dHDE), are used to assign DE schemes to individuals [11].

- (i) In the static HDE (sHDE), DE schemes are randomly assigned to individuals in population initialization. The assigned DE schemes do not change during the search process.
- (ii) In the dynamic HDE (dHDE), DE schemes are randomly assigned to individuals during initialization. As the iteration increases, the assigned DE schemes of individuals are not fixed. They can randomly change during the search process. An individual randomly selects a new DE scheme from the DE scheme pool when the individual fails to improve its objective fitness value. In the standard DE and its variants, the objective fitness value of each individual X_i satisfies $f(X_{i,1}) \leq f(X_{i,2}) \leq \dots \leq f(X_{i,G})$. If the new generated individual (trail vector U_i) could not improve its previous position (X_i), it may indicate early stagnation. This can be addressed by assigning a new DE scheme to the individual.

The main steps of dynamic heterogeneous DE (dHDE) are presented in Algorithm 1, where P is the current population, FEs is the number of fitness evaluations, and MAX_FEs is

the maximum number of FEs. For static HDE (sHDE), please delete line 21 in Algorithm 1.

4. Experimental Verifications

This section provides experimental studies of HDE on 18 well-known benchmark optimization problems. According to the properties of these problems, two series of experiments are conducted: (1) comparison of HDE on classical optimization problems and (2) comparison of HDE on shifted large-scale optimization problems.

4.1. Results on Classical Optimization Problems. In this section, 12 classical benchmark problems are used to verify the performance of HDE. These problems were considered in [2, 5, 17–19]. Table 1 presents a brief description of these benchmark problems. All the problems are to be minimized.

Experiments are conducted to compare HDE with other six DE variants. The involved algorithms are listed below:

- (i) DE/rand/1/bin,
- (ii) DE/best/1/bin,
- (iii) DE/BoR/1/bin [16],
- (iv) self-adapting DE (jDE) [5],
- (v) DE with neighborhood search (NSDE) [20],
- (vi) DE using neighborhood-based mutation (DEGL) [2],
- (vii) the proposed sHDE and dHDE.

In the experiments, we have two series of comparisons: (1) comparison of sHDE/dHDE with basic DE schemes and (2) comparison of sHDE/dHDE with state-of-the-art DE variants. The first comparison aims to check whether the heterogeneous method is helpful to improve the performance of DE. The second comparison investigates whether the proposed approach is better or worse than some recently proposed DE variants.

For these two comparisons, we use the same parameter settings as follows. For the three basic DE schemes (DE/rand/1/bin, DE/best/1/bin, and DE/BoR/1/bin) and sHDE/dHDE,

TABLE 1: The 12 classical benchmark optimization problems

Problem	Name	D	Properties	Search range
f_1	Sphere	25	Unimodal	$[-100, 100]$
f_2	Schewefel 2.22	25	Unimodal	$[-10, 10]$
f_3	Schewefel 1.2	25	Unimodal	$[-100, 100]$
f_4	Schewefel 2.21	25	Unimodal	$[-100, 100]$
f_5	Rosenbrock	25	Multimodal	$[-30, 30]$
f_6	Step	25	Unimodal	$[-100, 100]$
f_7	Quartic with noise	25	Unimodal	$[-1.28, 1.28]$
f_8	Schewefel 2.26	25	Multimodal	$[-500, 500]$
f_9	Rastrigin	25	Multimodal	$[-5.12, 5.12]$
f_{10}	Ackley	25	Multimodal	$[-32, 32]$
f_{11}	Griewank	25	Multimodal	$[-600, 600]$
f_{12}	Penalized	25	Multimodal	$[-50, 50]$

```

(1) Randomly initialize the population  $P$  including the variables and ID;
(2) Evaluate the objective fitness value of each individuals in  $P$ ;
(3) FEs =  $N_p$ ;
(4) while FEs  $\leq$  MAX_FEs do
(5)   for  $i = 1$  to  $N_p$  do
(6)     if  $ID_i == 1$  then
(7)       Use DE/rand/1/bin to generate a trail vector  $U_i$ ;
(8)     end
(9)     if  $ID_i == 2$  then
(10)      Use DE/best/1/bin to generate a trail vector  $U_i$ ;
(11)    end
(12)    if  $ID_i == 3$  then
(13)      Use DE/BoR/1/bin to generate a trail vector  $U_i$ ;
(14)    end
(15)    Evaluate the objective fitness value of  $U_i$ ;
(16)    FEs++;
(17)    if  $f(U_i) \leq f(X_i)$  then
(18)       $X_i = U_i$ 
(19)    end
(20)    else
(21)      /* For static HDE (sHDE), please delete lines 20–22 */
(21)      Randomly assign a new different DE scheme to  $X_i$ , (change the value of  $ID_i$ );
(22)    end
(23)  end
(24) end

```

ALGORITHM 1: The dynamic heterogeneous DE (dHDE).

the control parameters, F and CR, are set to 0.5 and 0.9, respectively [5]. For jDE, NSDE, and DEGL, the parameters F and CR are self-adaptive. For all algorithms, the population size (N_p) and maximum number of FEs are set to $10 \cdot D$ and $5.00E+05$, respectively [2]. All the experiments are conducted 30 times, and the mean error fitness values are reported.

4.1.1. Comparison of HDE with Basic DE Schemes. The comparison results of HDE with the three basic DE schemes are presented in Table 2, where “Mean” denotes the mean error fitness values. From the results, sHDE and dHDE outperform other three basic DE schemes on a majority of test problems. For unimodal problems ($f_1 - f_4$), DE/best/1/bin shows faster

convergence than other algorithms for the attraction of the global best individual. For f_6 and f_7 , all the algorithms obtain similar performance. DE/rand/1/bin provides high population diversity but slows down convergence rate. That is why DE/rand/1/bin performs better than DE/best/1/bin on most multimodal problems, but worse on unimodal problems. DE/BoR/bin/1 is a middle phase of DE/rand/1/bin and DE/best/1/bin. It obtains better performance than the other two basic DE schemes. For the heterogeneity of these basic DE schemes, sHDE and dHDE significantly improve the performance of DE. For unimodal problems, DE/best/1/bin obtains the best performance, while sHDE and dHDE perform better than DE/rand/1/bin and DE/BoR/1/bin. For

TABLE 2: Comparison of HDE with basic DE schemes on classical benchmark problems.

Problem	DE/rand/1/bin Mean	DE/best/1/bin Mean	DE/BoR/1/bin Mean	sHDE Mean	dHDE Mean
f_1	0.00E + 00	4.38E - 27	2.37E - 46	6.58E - 79	0.00E + 00
f_2	0.00E + 00	2.64E - 13	3.02E - 22	3.32E - 39	4.96E - 103
f_3	2.68E - 125	2.41E - 04	1.51E - 11	3.10E - 21	7.21E - 57
f_4	5.34E - 64	1.77E - 07	1.89E - 11	1.42E - 07	2.49E - 40
f_5	5.75E - 29	2.91E - 04	1.45E - 16	1.11E - 29	1.73E - 29
f_6	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
f_7	1.38E - 03	2.50E - 03	1.52E - 03	8.31E - 04	1.84E - 03
f_8	3.36E + 03	1.68E + 03	2.78E + 03	0.00E + 00	8.56E + 02
f_9	4.38E + 01	1.32E + 02	2.62E + 01	5.22E + 01	0.00E + 00
f_{10}	2.81E + 00	2.55E - 14	4.14E - 15	4.14E - 15	4.14E - 15
f_{11}	2.95E - 02	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
f_{12}	9.96E - 01	5.00E - 28	1.88E - 32	1.88E - 32	1.88E - 32

TABLE 3: Comparison of HDE with jDE, NSDE, and DEGL on classical benchmark problems.

Problem	jDE Mean	NSDE Mean	DEGL Mean	sHDE Mean	dHDE Mean
f_1	4.04E - 35	9.55E - 35	8.78E - 37	6.58E - 79	0.00E + 00
f_2	8.34E - 26	8.94E - 30	4.95E - 36	3.32E - 39	4.96E - 103
f_3	4.76E - 14	3.06E - 09	1.21E - 26	3.10E - 21	7.21E - 57
f_4	3.02E - 14	2.09E - 11	4.99E - 15	1.42E - 04	2.49E - 40
f_5	5.64E - 26	2.65E - 25	6.89E - 25	1.11E - 29	1.73E - 29
f_6	1.67E - 36	4.04E - 28	9.56E - 48	0.00E + 00	0.00E + 00
f_7	3.76E - 02	4.35E - 03	1.05E - 07	8.31E - 04	1.84E - 03
f_8	0.00E + 00	2.60E + 00	0.00E + 00	0.00E + 00	8.56E + 02
f_9	6.74E - 24	4.84E - 21	5.85E - 25	5.22E + 01	0.00E + 00
f_{10}	7.83E - 15	5.97E - 10	5.98E - 23	4.14E - 15	4.14E - 15
f_{11}	1.83E - 28	7.93E - 26	2.99E - 36	0.00E + 00	0.00E + 00
f_{12}	9.37E - 24	5.85E - 21	7.21E - 27	1.88E - 32	1.88E - 32

multimodal problems, DE/best/1/bin is the worst algorithm, while sHDE and dHDE obtain better performance than other algorithms.

For the comparison of sHDE and dHDE, sHDE performs better on 3 problems, while dHDE achieves better results on 5 problems. For the remaining 4 problems, both of them could search the global optimum. Although the dynamic heterogeneous model improves the performance of HDE on many problems, it may lead to premature convergence. In the dHDE, if the current individual could not improve its fitness value, a new DE scheme is assigned to the individual. If the new scheme is DE/best/1/bin, the individual will be attracted by the global best individuals. This will potentially run the risk of premature convergence.

4.1.2. *Comparison of HDE with Other State-of-the-Art DE Variants.* The comparison results of HDE with other three state-of-the-art DE variants are presented in Table 3, where “Mean” denotes the mean error fitness values. From the results, sHDE and dHDE perform better than other three DE algorithms on the majority of test problems. dHDE

TABLE 4: Average rankings achieved by Friedman test.

Algorithms	Ranking
dHDE	4.17
sHDE	3.58
DEGL	3.50
jDE	2.25
NSDE	1.50

outperforms jDE and NSDE on all test problems except for f_8 . On this problem, dHDE falls into the local minima, while sHDE could successfully solve it. dHDE achieves better results than DEGL on 9 problems, while DEGL performs better than dHDE on the remaining 3 problems.

In order to compare the performance of multiple algorithms on the test suite, we conducted Friedman test according to the suggestions of [21]. Table 4 shows the average ranking of jDE, NSDE, DEGL, sHDE, and dHDE. As seen, the performance of the five algorithms can be sorted by average ranking into the following order: dHDE, sHDE, DEGL,

TABLE 5: The 6 shifted large-scale benchmark optimization problems proposed in [15].

Problem	Name	D	Properties	Search range
F_1	Shifted Sphere	500	Unimodal, separable, scalable	$[-100, 100]$
F_2	Shifted Schewefel 2.21	500	Unimodal, nonseparable	$[-100, 100]$
F_3	Shifted Rosenbrock	500	Multimodal, nonseparable	$[-100, 100]$
F_4	Shifted Rastrigin	500	Multimodal, separable	$[-5, 5]$
F_5	Shifted Griewank	500	Multimodal, nonseparable	$[-600, 600]$
F_6	Shifted Ackley	500	Multimodal, separable	$[-32, 32]$

TABLE 6: Comparison of HDE with ODE and MDE on shifted large-scale benchmark problems.

Problem	ODE	MDE	sHDE	dHDE
	Mean	Mean	Mean	Mean
F_1	$8.02E + 01$	$1.95E + 01$	$8.45E + 00$	$1.14E - 10$
F_2	$5.78E + 00$	$2.70E + 01$	$8.54E + 01$	$1.01E + 02$
F_3	$1.54E + 05$	$4.67E + 05$	$1.52E + 05$	$1.29E + 03$
F_4	$4.22E + 03$	$4.14E + 03$	$5.27E + 03$	$3.42E + 03$
F_5	$1.77E + 00$	$1.52E + 00$	$6.94E - 01$	$1.83E - 01$
F_6	$4.51E + 00$	$4.02E + 00$	$1.26E + 00$	$1.58E + 01$

jDE, and NSDE. The best average ranking was obtained by the dHDE algorithm, which outperforms the other four algorithms.

4.2. Results of HDE on Shifted Large-Scale Optimization Problems. To verify the performance of HDE on complex optimization problems, this section provides experimental studies of HDE on 6 shifted large-scale problems. These problems were considered in CEC 2008 special session and competition on large-scale global optimization [15]. Table 5 presents a brief descriptions of these benchmark problems. All the problems are to be minimized.

Experiments are conducted to compare four DE algorithms including the proposed sHDE and dHDE on 6 test problems with $D = 500$. The involved algorithms are listed below:

- (i) opposition-based DE (ODE) [22],
- (ii) modified DE using Cauchy mutation (MDE) [23],
- (iii) the proposed sHDE and dHDE.

To have a fair comparison, all the four algorithms use the same parameter settings by the suggestions of [22]. The population size N_p is set to D . The control parameters, F and CR, are set to 0.5 and 0.9, respectively [22]. For ODE, the rate of opposition J_r is 0.3. The maximum number of FEs is $5000 \cdot D$. All the experiments are conducted 30 times, and the mean error fitness values are reported.

Table 6 presents the comparison results of HDE with ODE and MDE on shifted large-scale problems. As seen, sHDE and dHDE outperform ODE and MDE on four problems. The static heterogeneous model slightly improves the final solutions of DE, while the dynamic model significantly improves the results on F_1 and F_3 . However, the two heterogeneous models do not always work. For some problems, sHDE and/or dHDE perform worse than ODE and MDE. The

possible reason is that the heterogeneous models may hinder the evolution. For the static model, the entire population is divided into three groups. Each group uses a different basic DE scheme to generate new individuals. Though these groups share the search information of the entire population, each group with small population may obtain slow convergence rate. As mentioned before, the dynamic heterogeneous model runs the risk of premature convergence.

5. Conclusion

In the standard DE and most of its modifications, individuals follow the same search behavior for using the same DE scheme. The effect is that the algorithms could hardly balance explorative and exploitative abilities during the evolution. Inspired by the heterogeneous swarms, a simple and effective heterogeneous DE (HDE) is proposed in this paper, where individuals could follow different search characteristics randomly selected from a DE scheme pool. To implement the HDE, two heterogeneous models are proposed, one static, where individuals do not change their search behaviors, and a dynamic model where individuals may change their search behaviors. Both versions of HDE initialize individual behaviors by randomly selecting DE schemes from a DE scheme pool. In the dynamic HDE, when an individual does not improve its fitness value, a new different DE scheme is randomly selected from the DE scheme pool.

To verify the performance of HDE, two types of benchmark problems, including classical problems and shifted large-scale problems, are used in the experiments. Simulation results show that the proposed sHDE and dHDE outperform the other eight DE variants on a majority of test problems. It demonstrates that the heterogeneous models are helpful to improve the performance of DE.

For the dynamic heterogeneous model, the frequency of changing new DE schemes may affect the performance of

dHDE. More experiments will be investigated. The concept of heterogeneous swarms can be applied to other evolutionary algorithms to obtain good performance. Future research will investigate different algorithms based on heterogeneous swarms. Moreover, we will try to apply our approach to some real-world problems [24, 25].

Conflict of Interests

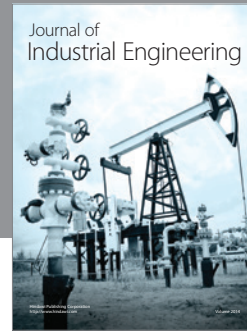
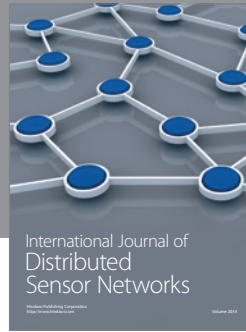
The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the Humanity and Social Science Foundation of Ministry of Education of China (no. 13YJCZH174), the National Natural Science Foundation of China (nos. 61305150, 61261039, and 61175127), the Science and Technology Plan Projects of Jiangxi Provincial Education Department (no. GJJ13744 and GJJ13762), and the Foundation of State Key Laboratory of Software Engineering (no. SKLSE2012-09-19).

References

- [1] 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.
- [2] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.
- [3] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.
- [4] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, vol. 9, no. 6, pp. 448–462, 2005.
- [5] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.
- [6] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [7] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 61–106, 2010.
- [8] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [9] X. S. Yang and S. Deb, "Two-stage eagle strategy with differential evolution," *International Journal of Bio-Inspired Computation*, vol. 4, no. 1, pp. 1–5, 2012.
- [10] Y. W. Zhong, L. J. Wang, C. Y. Wang, and H. Zhang, "Multi-agent simulated annealing algorithm based on differential evolution algorithm," *International Journal of Bioinspired Computation*, vol. 4, no. 4, pp. 217–228, 2012.
- [11] A. P. Engelbrecht, "Heterogeneous particle swarm optimization," *International Conference on Swarm Intelligence*, vol. 6234, pp. 191–202, 2010.
- [12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.
- [13] E. GarGarcíaI-Gonzalo and J. L. Fernández-Martínez, "A brief historical review of particle swarm optimization (PSO)," *Journal of Bioinformatics and Intelligent Control*, vol. 1, no. 1, pp. 3–16, 2012.
- [14] W. Gong, Z. Cai, C. X. Ling, and H. Li, "Enhanced differential evolution with adaptive strategies for numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 41, no. 2, pp. 397–413, 2011.
- [15] K. Tang, X. Yao, P. N. Suganthan et al., Benchmark functions for the CEC' 2008 special session and competition on high-dimensional real-parameter optimization, Nature Inspired Computation and Applications Laboratory, USTC, Hefei, China, 2007.
- [16] C. Lin, A. Qing, and Q. Feng, "A new differential mutation base generator for differential evolution," *Journal of Global Optimization*, vol. 49, no. 1, pp. 69–90, 2011.
- [17] L. Ali and S. L. Sabat, "Particle swarm optimization based universal solver for global optimization," *Journal of Bioinformatics and Intelligent Control*, vol. 1, no. 1, pp. 95–105., 2012.
- [18] X. J. Cai, S. J. Fan, and Y. Tan, "Light responsive curve selection for photosynthesis operator of APOA," *International Journal of Bio-Inspired Computation*, vol. 4, no. 6, pp. 373–379, 2012.
- [19] L. P. Xie, J. C. Zeng, and R. A. Formato, "Selection strategies for gravitational constant G in artificial physics optimisation based on analysis of convergence properties," *International Journal of Bio-Inspired Computation*, vol. 4, no. 6, pp. 380–391, 2012.
- [20] Z. Yang, J. He, and X. Yao, "Making a difference to differential evolution," *Advance in Metaheuristics for Hard Optimization*, pp. 397–414, 2008.
- [21] 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.
- [22] R. S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.
- [23] M. Ali and M. Pant, "Improving the performance of differential evolution algorithm using Cauchy mutation," *Soft Computing*, vol. 15, no. 5, pp. 991–1007, 2011.
- [24] A. Y. Abdelaziz, R. A. Osama, and S. M. Elkhodary, "Application of ant colony optimization and harmony search algorithms to reconfiguration of radial distribution networks with distributed generations," *Journal of Bioinformatics and Intelligent Control*, vol. 1, no. 1, pp. 86–94, 2012.
- [25] S. Y. Zeng, Z. Liu, C. H. Li, Q. Zhang, and W. Wang, "An evolutionary algorithm and its application in antenna design," *Journal of Bioinformatics and Intelligent Control*, vol. 1, no. 2, pp. 129–137, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

