Hindawi
Computational Intelligence and Neuroscience
Volume 2017, Article ID 8469103, 23 pages
https://doi.org/10.1155/2017/8469103

Research Article

Differential Cloud Particles Evolution Algorithm Based on Data-Driven Mechanism for Applications of ANN

Wei Li

School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China

Correspondence should be addressed to Wei Li; liwei@xaut.edu.cn

Received 30 January 2017; Revised 12 May 2017; Accepted 16 May 2017; Published 6 July 2017

Academic Editor: José Alfredo Hernández-Pérez

Computational scientists have designed many useful algorithms by exploring a biological process or imitating natural evolution. These algorithms can be used to solve engineering optimization problems. Inspired by the change of matter state, we proposed a novel optimization algorithm called differential cloud particles evolution algorithm based on data-driven mechanism (CPDD). In the proposed algorithm, the optimization process is divided into two stages, namely, fluid stage and solid stage. The algorithm carries out the strategy of integrating global exploration with local exploitation in fluid stage. Furthermore, local exploitation is carried out mainly in solid stage. The quality of the solution and the efficiency of the search are influenced greatly by the control parameters. Therefore, the data-driven mechanism is designed for obtaining better control parameters to ensure good performance on numerical benchmark problems. In order to verify the effectiveness of CPDD, numerical experiments are carried out on all the CEC2014 contest benchmark functions. Finally, two application problems of artificial neural network are examined. The experimental results show that CPDD is competitive with respect to other eight state-of-the-art intelligent optimization algorithms.

1. Introduction

Optimization problems in engineering are often very complex and difficult to solve. At first, lots of deterministic algorithms based on mathematical programming theory are used for engineering optimization problems. They obtain better results for relative simple and ideal models. Unfortunately, these deterministic algorithms show poor performance for real-world complex problems. Inspired by natural evolution, more and more researchers are interested in the development of nature-inspired algorithm by exploring natural phenomena [1]. These natural phenomena mainly include the biological evolutionary process, animal behavior, and physical phenomena. The nature-inspired algorithms can solve the difficult design and optimization problems by building solutions that are more fit relative to desired properties [2]. The resulting field, nature-inspired algorithms have been successful in solving optimization, design, constrained, large-scale, and multiobjective clustering and forecasting [3–9].

Evolutionary algorithms (EAs), such as Genetic Algorithm (GA) [10], Differential Evolution (DE) [11], and Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES) [12], are inspired by the biological evolutionary process. GA, proposed by Fraser and popularized by Holland, has been widely studied. It solves optimization problems by simulating Darwinian evolution concepts, such as crossover, mutation, and selection. DE, proposed by Storn and Price, is a simple yet powerful population-based evolutionary algorithm. CMA-ES, proposed by Hansen and Ostermeier, adapts the complete covariance matrix of the normal mutation distribution. Swarm intelligence (SI) algorithms, such as Particle Swarm Optimization (PSO) [13, 14], Artificial Bee Colony (ABC) [15], Teaching-Learning-Based Optimization (TLBO) [16, 17], and Jaya (a Sanskrit word meaning victory) algorithm [18], are inspired by all kinds of animal behavior. PSO explores the search space according to *pbest* and *gbest*, which are the past best position and the global best position achieved by particles, respectively. ABC, proposed by Karaboga, simulates the foraging behavior of the honeybee swarm and has been applied to solve many engineering optimization problems [19, 20]. The TLBO method, proposed by Rao, is based on the effect of the influence of a teacher on the output of learners in a class [16]. In order to reduce the complexity of the algorithm, Rao [18] proposed Jaya algorithm which uses one phase instead of two phases of the TLBO algorithm. Jaya algorithm tries to get closer to reaching the best solution and tries to move away from the worst solution. Other heuristic methods, such as biogeography-based optimization (BBO) [21], Simulated Annealing (SA) [22, 23], Chemical Reaction Optimization (CRO) [24], and Brain Storm Optimization (BSO) [25], simulate the physical phenomena or rules. BBO, proposed by Simon in 2008, is a newly proposed metaheuristic algorithm. In BBO, mathematical models are used to describe the evolution process of species, such as migration, mutation, and distribution of species. A solution is regarded as an island (habitat) with a habitat suitability index (HSI). Islands with a high HIS are well suited for species, and vice versa. Suitability index variables (SIV), which refer to the features correlated with HSI, and HSI are considered as the search space and objective function, respectively [26]. SA is a heuristic algorithm which is based on an analog of thermodynamics with the way metals cool and anneal [27]. CRO is a chemical-reaction-inspired metaheuristic which mimics the characteristics of chemical reactions in solving optimization problems [24]. BSO mimics the brainstorming process in which a group of people solves a problem together [28].

The issue of exploration-exploitation is the major factor which influences the performance of evolutionary algorithm. Exploration helps to find new potential solutions and improve the convergence rate of algorithm. Exploitation helps to improve the quality of found-so-far solutions. However, overexploration may lead to slow convergence and overexploitation may increase the risk of premature convergence. Therefore, numerous ideas are proposed to balance the exploitation and exploration search process of EAs [29]. For example, a large scaling factor F in DE is required at the early stage of the evolution to ensure strong exploration capability, while a small F is preferred to improve exploitation capability at the later stage [30]. Therefore, many improved DE variants, such as adaptive differential evolution with optional external archive (JADE) [31], self-adaptive control parameters in differential evolution (jDE) [32], a self-adaptive DE (SaDE) algorithm [33], and a composite DE (CoDE) algorithm [34], are proposed to improve the relationship of exploration and exploitation with the proper settings of control parameters [35–39]. For PSO, many different modified PSO variants, including inertia weight and the acceleration coefficient, are proposed by the researchers to enhance PSO's exploration capability and alleviate premature convergence problem [40–46].

Designing suitable evolutionary strategies and control parameters is important to realize a good balance between exploration and exploitation. In this paper, we proposed a novel nature-inspired algorithm called differential cloud particles evolution algorithm based on data-driven (CPDD) mechanism for solving global optimization problems. The CPDD algorithm simulates the change of matter state and cloud transformation process. The optimization process is divided into two stages, namely, fluid stage and solid stage. The CPDD algorithm carries out the strategy of integrating global exploration with local exploitation in fluid stage. Furthermore, local exploitation is carried out mainly in solid stage. Data-driven mechanism is designed for obtaining better control parameters to keep a better balance between exploration and exploitation.

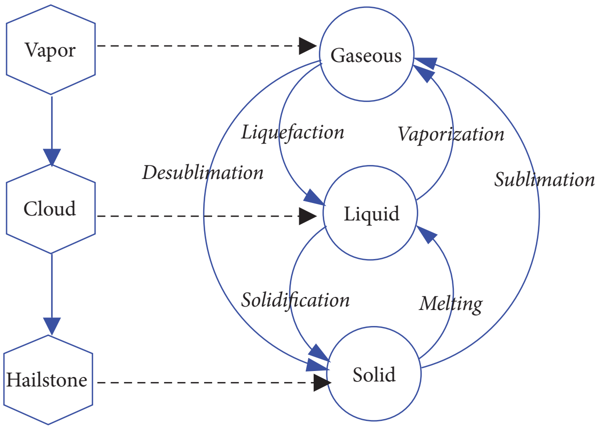


Figure 1: Cloud formation and matter state transition.

The rest of the paper is organized as follows. Section 2 introduces the proposed CPDD and the concepts behind it in detail. In Section 3, the performance of the proposed CPDD is validated on different optimization problems. Section 4 shows the applications for training artificial neural network. Finally, the conclusions and ideas for future research are drawn up in Section 5.

2. Differential Cloud Particles Evolution Algorithm Based on Data-Driven Mechanism

2.1. Algorithm Background. The idea of differential cloud particles evolution algorithm based on data-driven mechanism is inspired from the phenomenon of matter state transition and cloud formation. For further understanding, an explanation on the principles of matter state transition and cloud formation will be stated as follows.

Substances commonly exist in three states: gaseous, liquid, and solid. For example, heat evaporates water into steam, while low temperature turns water to ice. Similarly, three states are involved in the formation and change of cloud. The water evaporated into vapor. Water vapor condenses to form clouds. The clouds transform into snow or hailstone with decreasing temperature. Therefore, water vapor, cloud, and snow can be regarded as gaseous, liquid, and solid, respectively. The cloud transformation process and matter state transition are illustrated in Figure 1. As a result, the matter shows different states with different temperature.

CPDD loosely simulates the cloud transformation process and matter state transition. Because the gas and the liquid have fluidity, there are two types of operation implemented in CPDD, namely, fluid stage operation and solid stage operation. We utilize the term "phase transition" to describe transitions between fluid stage and solid stage. A phase transition is the transformation of a thermodynamic system

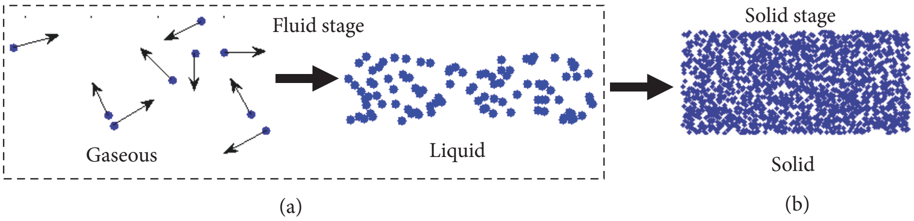


Figure 2: Three states of cloud particles.

from one phase or state of matter to another one by heat transfer.

2.2. The Proposed CPDD. Similar to other metaheuristic algorithms, the proposed algorithm begins with an initial population called the cloud particles. Each cloud particle in CPDD algorithm represents a solution in the population. The population size is similar to the temperature in real world. Liquefaction refers to the process in which gas is transformed into liquid. Solidification is a phase transition in which liquid is turned into solid. Liquefaction and solidification need to give off heat. The temperature will gradually decrease in the exothermic process. In order to improve the performance of the algorithm, EAs had better to start with exploration and then gradually change into exploitation [47]. Therefore, CPDD algorithm has a larger population to ensure strong exploration ability at the beginning of the evolution. The population size gradually decreased and the exploitation ability gradually improved during evolution. In the later evolution, CPDD algorithm has a smaller population to encourage broader exploitation. The change of population number is similar to the change of the temperature; namely, matter has high temperature in gas state and has low temperature in solid state.

2.2.1. Fluid Stage. The fluid stage, which mimics the movement of the fluid, includes gaseous state and liquid state. At the beginning, the initial state is gaseous state, which is composed of many cloud particles. Each cloud particle has good fluidity and can move freely. Therefore, the algorithm shows strong exploration ability in gaseous state. For the evolutionary process, the population size gradually decreases. Then, the local exploitation ability can be improved by reducing the population size. In addition, the state of population transforms from gaseous state into liquid one as the cloud particles move. The movement of cloud particles in liquid state causes macrolocal exploitation. The movement of cloud particles in fluid stage is given in Figure 2(a).

Inspired by JADE, an improved search strategy is introduced in fluid stage. Similar to other evolutionary algorithms for optimization problems, a CPDD population is represented as a set of real parameter vectors which is defined as follows:

$$\mathbf{x}_i = \left(x_1, x_2, \ldots, x_D\right), \quad i = 1, \ldots, N, \tag{1}$$

where D is the dimensionality of the optimization problem and N is the population size. In fluid stage, the search strategy, based on DE/current-to-$p$best with optional archive, is generated in the following manner:

$$\mathbf{v}_i = \mathbf{x}_{r1} + F_i \cdot \left(\mathbf{x}_{best} - \mathbf{x}_{r1} + \mathbf{x}_{r2} - \mathbf{x}_{r3}\right), \tag{2}$$

where $i \in \{1, \ldots, N\}$, r_1, r_2, and r_3 are mutually different random integer indices selected from $i \in \{1, \ldots, N\}$, and $\mathbf{x}_{best}$ is randomly chosen as one of the top $100p\%$ cloud particles in the current population. p is 15% of the population size. $\mathbf{x}_{r3}$ is selected from the union of the population and the archive. The archive is the set of archived inferior solutions and is used for maintaining diversity in JADE [34]. If the archive size exceeds 150% of the population size, some solutions are randomly removed from the archive so that some newly cloud particles can be inserted into the archive. F_i, which is different from F in JADE, is a mutation factor that controls the speed of algorithm process and is generated at each generation by data-driven mechanism introduced later in (4).

2.2.2. Solid Stage. The algorithm has found the potential area of optimal solution after global exploration and macrolocal exploitation in fluid stage. Then the phase transition is carried out; namely, the population transforms from fluid into solid. The cloud particles vibrate only in a small region and carry out microlocal exploitation. The process, in which most of the cloud particles are gathered toward the optimal solution (microlocal exploitation), is analogous to the solidification. Finally, most of the cloud particles will gather to a position, that is, the location of the optimal solution. Figure 2(b) shows the movement of cloud particles in solid stage. The following strategy is search strategy used in solid stage:

$$\mathbf{v}_i = \mathbf{x}_{best} + F_i \cdot \left(\mathbf{x}_{r2} - \mathbf{x}_{r1}\right). \tag{3}$$

Similar to fluid stage, $\mathbf{x}_{best}$ is randomly chosen as one of the top $100p\%$ cloud particles in the current population. $\mathbf{x}_{r2}$ is selected from the union of the population and the archive. r_1 and r_2 are mutually different random integer indices selected from $i \in \{1, \ldots, N\}$. F_i is generated at each generation by data-driven mechanism.

The population size is reducing from N to four in order to balance the exploration and exploitation. In fluid stage, the algorithm carries out global exploration and macrolocal exploitation. In solid stage, the algorithm performs macrolocal exploitation. Consequently, the algorithm can balance the exploration and exploitation based on the transformation from fluid stage to solid stage.

2.3. Control Parameters Assignments. In CPDD algorithm, the parameter F controls the diversity of the population. Higher value of the parameter F will increase the diversity of the cloud particles and enhance the convergence speed. On the contrary, smaller value of the parameter F will result in premature convergence and slow convergence rate. In addition, for different problems, different values of parameters are needed in different evolution stage. Based on our previous research [48], in this paper, F is produced based on data-driven mechanism which can be described as follows:

$$\begin{aligned} f_1 &= \sin\left(\frac{e^{f_0}}{2}\right) \\ f_2 &= \frac{r_1 \times \sin(1 - f_0)}{\sqrt{\max FES}} + \sin(1 - f_0) \\ F_i &= r_2 \times f_2 + f_1. \end{aligned} \tag{4}$$

Initial F_i is generated according to (4). f_0 is a parameter and is set to 0.068. In each generation, F_i value that succeeds in generating a trial $\mathbf{v}_i$ which is better than the parent individual $\mathbf{x}_i$ is preserved and is recorded as F_G. The size of F_G is recorded as $|F_G|$. If $|F_G|$ exceeds the current population size, randomly selected elements are deleted. If $|F_G|$ is less than the current population size N, the $(N\text{-}|F_G|)$ new F' is generated according to (4). Consequently, F_i value that shows better performance is preserved for next generation. It means that F_i value from last generation drives the evolution of new generation.

CR, the crossover rate, is another important control parameter in CPDD algorithm. The initial CR is a Gaussian distribution with mean "0.5" and standard deviation "0.1." Similar to the parameter F, CR values that have performed well in generation G are preserved for next generation. The preserved CR is recorded as CR_G. The size of CR_G is recorded as $|\text{CR}_G|$. If $|\text{CR}_G| = 0$, CR_i is regenerated as follows:

$$\text{CR}_i = 0.9 + \text{rand}. \tag{5}$$

In addition, the cloud particles transfer from fluid stage to solid stage when CR_G is empty.

If $|\text{CR}_G|$ is less than the current population size N, the $(N\text{-}|\text{CR}_G|)$ new CR' is given by

$$\text{CR}' = \sigma(\text{CR}_G) + \text{rand}, \tag{6}$$

where $\sigma(\text{CR}_G)$ refers to the standard deviation of CR_G. rand denotes a uniformly selected random number from [0,1). If $|\text{CR}_G| > N$, extra elements are randomly selected and deleted. Therefore, the next generation of CR is produced by CR_G, which shows better performance in the last generation.

The pseudocode of CPDD is illustrated in Algorithm 1. $\Phi(0.5, 0.1)$ refers to a Gaussian distribution with mean "0.5" and standard deviation "0.1." Popsize is the current population size. FES stands for the number of function evaluations. maxFES stands for the maximum number of function evaluations.

3. Experiments and Discussions

3.1. General Experimental Setting

3.1.1. Test Problems and Dimension Setting. For a comprehensive evaluation of CPDD, all the *CEC2014 Special Session on Real-Parameter Single Objective Optimization* [49] benchmark problems are used to test the performance of CPDD. The CEC2014 benchmark set consists of 30 test problems. According to their shape characteristics, these benchmark problems can be broadly classified into four kinds of optimization problems [49]:

(i) Unimodal problems f_1–f_3

(ii) Simple multimodal problems f_4–f_{16}

(iii) Hybrid problems f_{17}–f_{22}

(iv) Composite problems f_{23}–f_{30}

For all of the problems, the search space is $[-100, 100]^\text{D}$. In this paper, the dimension (D) of all problems is set to 10 and 30.

3.1.2. Experimental Platform and Termination Criterion. For all experiments, 30 independent runs are carried out on the same machine with a Celeron 3.40 GHz CPU, 4 GB memory, and windows 7 operating system with Matlab R2009b and conducted with $D \times 10,000$ (number of function evaluations, FES).

3.1.3. Performance Metrics. In our experimental studies, the mean value (F_mean), standard deviation (SD), maximum value (Max), and minimum value (Min) of the *solution error measure* [50] which is defined as $f(x) - f(x^*)$ are recorded for evaluating the performance of each algorithm, where $f(x)$ is the best fitness value found by an algorithm in a run and $f(x^*)$ is the real global optimization value of tested problem. In order to statistically compare the proposed algorithm with its peers, the statistical tool t-test [16] at a 0.05 significant level is used to evaluate whether the median fitness values of two sets of obtained results are statistically different from each other. Three marks "+," "−," and "≈" are also used to report the results clearly."+," "−," and "≈" denote that the performance of CPDD is better than, worse than, and similar to that of the corresponding algorithm, respectively.

3.2. Comparison with Eight State-of-the-Art Optimization Algorithms on 10 and 30 Dimensions. In this part, CPDD is compared with PSO, PSOcf (PSO with constriction factor) [44], TLBO, DE, JADE, CMA-ES, ABC, and BBO. The appropriate parameters are important for the performance of the optimization algorithms. Therefore, the setting of parameters of different algorithms is given in the following.

For CPDD, F and CR are produced according to the data-driven mechanism. Initial F is produced according to (4). The initial CR is a Gaussian distribution with mean "0.5" and standard deviation "0.1." The population size N is set to $13 \times D$. The maximum size of the archive is set to $1.5 \times N$. For DE, the population size N is set to 100. F and CR are set to 0.5

```
Initialize $D$ (number of dimensions), $N$ (number of population); Archive $A = \varphi$; $A_n = 0.3 \times N$;
$T_0$ (initial temperature) = $N$;
State = 0;
Initialize population randomly
while the termination criteria are not met do
    for $i = 1$ to $N$ do
        $\mathrm{CR}_i = \Phi(0.5, 0.1)$;
        Generate mutation factor $F_i$ according to Eq. (4)
        if State == 0 then
            Generate new individual according to Eq. (2) % fluid stage
        else
            Generate new individual according to Eq. (3) % solid stage
        endif
    endfor
    for $i = 1$ to $N$ do
        if $f(\mathbf{u}_i) < f(\mathbf{x}_i)$ then
            $A = A \cup \mathbf{x}_i$
            $\mathbf{x}_i = \mathbf{u}_i$
        endif
    endfor
    $T = \lceil T_0 - \frac{T_0}{\max \mathrm{FES}} \times \mathrm{FES} \rceil$
    if $N > T$ then
        $N = N - 1$ % cool down
        delete the worst individual
        Resize archive size
    endif
    if $|\mathrm{CR}_G| < N$ then
        if $\mathrm{CR}_G = \phi$ then
            Generate CR according to Eq. (5)
            State = 1
        else
            Generate CR$'$ according to Eq. (6)
            $\mathrm{CR} = \mathrm{CR}_G \cup \mathrm{CR}'$
        endif
    elseif $|\mathrm{CR}_G| > N$ then
        delete randomly selected elements from the $\mathrm{CR}_G$ so that the cross rate size is $N$
    else
        $\mathrm{CR} = \mathrm{CR}_G$
    endif
if $|F_G| < N$ then
    generate $(N - |F_G|)$ new $F'$ according to Eq. (4)
    $F = F_G \cup F'$
elseif $|F_G| > N$ then
    delete randomly selected elements from the $F_G$ so that the mutation factor size is $N$
else
    $F = F_G$
endif
endwhile
Output: the cloud particle with the smallest objection function value in the population.
```

Algorithm 1: CPDD algorithm.

and 0.9, respectively. For PSO, the population size N is set to 40, the linearly decreasing inertia ω from 0.9 to 0.4 is adopted over the course of the search, and the acceleration coefficients c_1 and c_2 are both set to 1.49445. For ABC, the number of colony sizes is set to 20, and the number of food sources is set to half of the colony sizes. For JADE, the population size N is set to 100; $p = 0.05$ and $c = 0.1$. The parameters of other algorithms are the same as those used in the corresponding references.

The statistical results, in terms of F_{mean}, SD, Max, and Min obtained in 30 independent runs by each algorithm, are reported in Tables 1~7.

Table 1: Experimental results of PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, BBO, and CPDD on f_1–f_3 test problems of 10 and 30 variables.

$F(D = 10)$		PSO	PSOcf	TLBO	DE	JADE	CMA-ES	ABC	BBO	CPDD
f_1	F_{mean}	7.05e + 03(+)	4.36e + 04(+)	5.42e + 05(+)	1.42e − 14(+)	**0.00e + 00**(≈)	**0.00e + 00**(≈)	1.93e + 05(+)	4.82e + 05(+)	**0.00e + 00**
	SD	1.50e + 04	2.78e + 04	7.53e + 05	2.90e − 14	**0.00e + 00**	**0.00e + 00**	1.52e + 05	7.59e + 05	**0.00e + 00**
	Max	1.38e + 05	1.06e + 05	2.35e + 06	1.27e − 13	**0.00e + 00**	**0.00e + 00**	5.81e + 05	3.25e + 06	**0.00e + 00**
	Min	1.76e + 02	4.67e + 03	2.18e + 03	**0.00e + 00**	**0.00e + 00**	**0.00e + 00**	1.78e + 04	1.99e + 04	**0.00e + 00**
f_2	F_{mean}	2.86e + 03(+)	2.65e + 02(+)	6.05e + 03(+)	**0.00e + 00**(≈)	**0.00e + 00**(≈)	**0.00e + 00**(≈)	8.01e + 01(+)	2.82e + 04(+)	**0.00e + 00**
	SD	2.38e + 03	4.53e + 02	4.21e + 03	**0.00e + 00**	**0.00e + 00**	**0.00e + 00**	8.73e + 01	2.48e + 04	**0.00e + 00**
	Max	7.77e + 03	2.18e + 03	1.18e + 04	**0.00e + 00**	**0.00e + 00**	**0.00e + 00**	3.16e + 02	1.14e + 05	**0.00e + 00**
	Min	1.03e − 01	1.18e − 02	4.43e + 00	**0.00e + 00**	**0.00e + 00**	**0.00e + 00**	6.22e − 02	4.70e + 03	**0.00e + 00**
f_3	F_{mean}	4.03e + 02(+)	1.05e + 03(+)	2.30e + 02(+)	**0.00e + 00**(≈)	5.31e − 03(+)	**0.00e + 00**(≈)	2.95e + 02(+)	6.05e + 03(+)	**0.00e + 00**
	SD	4.33e + 02	6.08e + 02	2.84e + 02	**0.00e + 00**	1.24e − 02	**0.00e + 00**	3.12e + 02	5.66e + 03	**0.00e + 00**
	Max	1.83e + 03	3.25e + 03	8.67e + 02	**0.00e + 00**	5.38e − 02	**0.00e + 00**	1.43e + 03	2.36e + 04	**0.00e + 00**
	Min	2.73e + 01	1.63e + 02	7.03e − 01	**0.00e + 00**	**0.00e + 00**	**0.00e + 00**	1.76e + 01	1.11e + 02	**0.00e + 00**
$F(D = 30)$		PSO	PSOcf	TLBO	DE	JADE	CMA-ES	ABC	BBO	CPDD
f_1	F_{mean}	7.54e + 06(+)	4.71e + 05(+)	5.91e + 07(+)	7.20e + 04(+)	4.94e + 02(+)	**1.42e − 14**(−)	1.05e + 07(+)	1.88e + 07(+)	4.37e − 05
	SD	8.72e + 06	4.59e + 05	5.69e + 07	5.47e + 04	9.95e + 02	**2.02e − 14**	2.71e + 06	1.12e + 07	2.68e − 05
	Max	3.47e + 07	1.64e + 06	2.55e + 08	2.11e + 05	4.23e + 03	**5.68e − 14**	1.77e + 07	4.90e + 07	9.79e − 05
	Min	1.45e + 05	8.90e + 04	4.42e + 06	5.19e + 03	3.01e − 02	**1.42e − 14**	5.98e + 06	3.23e + 06	5.87e − 06
f_2	F_{mean}	5.22e − 01(+)	1.32e + 01(+)	8.95e + 09(+)	0.00e + 00(≈)	0.00e + 00(≈)	2.84e − 14(+)	2.49e + 02(+)	4.23e + 06(+)	**0.00e + 00**
	SD	1.19e + 00	1.35e + 01	7.54e + 09	7.46e − 15	2.17e − 14	3.87e − 14	2.56e + 02	1.87e + 06	**0.00e + 00**
	Max	5.01e + 00	5.85e + 01	4.09e + 10	2.84e − 14	2.84e − 14	1.13e − 13	1.11e + 03	9.53e + 06	**0.00e + 00**
	Min	2.75e − 07	1.51e + 00	1.12e + 09	0.00e + 00	0.00e + 00	2.84e − 14	1.16e + 00	1.68e + 06	**0.00e + 00**
f_3	F_{mean}	3.92e + 00(+)	6.14e + 02(+)	3.11e + 03(+)	0.00e + 00(≈)	7.53e − 03(+)	5.68e − 14(+)	9.44e + 02(+)	6.73e + 03(+)	**0.00e + 00**
	SD	6.44e + 00	4.16e + 02	6.40e + 03	1.05e − 14	3.50e − 02	6.67e − 14	7.89e + 02	5.92e + 03	**0.00e + 00**
	Max	2.82e + 01	1.69e + 03	2.71e + 04	5.68e − 14	1.90e − 01	2.27e − 13	2.98e + 03	2.12e + 04	**0.00e + 00**
	Min	7.95e − 03	3.50e + 01	5.25e − 02	0.00e + 00	5.68e − 14	5.68e − 14	3.59e + 01	3.75e + 01	**0.00e + 00**

3.2.1. Unimodal Problems f_1–f_3. From the statistical results of Table 1, we can see that CMA-ES and CPDD achieve the optimal solution in each run for unimodal problems f_1–f_3 for 10 dimensions. CPDD performs better than other algorithms on f_2-f_3 and achieves the second best performance on f_1 for 30. The reason that CPDD has the outstanding performance may be because of data-driven mechanism, which is helpful for obtaining better control parameters.

3.2.2. Simple Multimodal Problems f_4–f_{16}. From Table 2, we observe from the statistical results that CPDD is significantly better than other algorithms on f_5–f_7, f_9, f_{13}, f_{14}, and f_{16}. ABC performs well on f_4; JADE performs well on f_{10}-f_{11} and f_{15}, respectively. PSOcf performs well on f_{12}. CPDD and JADE achieve the optimal solution on f_8. Table 3 shows that CPDD obtains better solutions than other algorithms on f_4, f_6, f_{12}, and f_{13}. CMA-ES performs well on f_5. JADE performs well on f_8–f_{11}, f_{15}, and f_{16}. ABC performs well on f_{14}. DE, JADE, and CPDD perform well and achieve the similar solutions on f_7.

3.2.3. Hybrid Problems f_{17}–f_{22}. In the case of f_{17}–f_{22}, we find that CPDD achieves very competitive results from Tables 4 and 5. It beats PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, and BBO on these hybrid problems for 10 and 30 dimensions except f_{17}. DE performs well on f_{17} for 10 dimensions. This may be due to the fact that the scheme of matter state change can help CPDD to keep a better balance between exploration and exploitation.

3.2.4. Composite Problems f_{23}–f_{30}. From Tables 6 and 7, we find that these composite problems are very time-consuming for fitness evaluation compared to others because these problems combine multiple test problems into a complex landscape. Therefore, it is extremely difficult for state-of-the-art optimization algorithms to obtain relatively ideal results. Table 6 shows that CPDD obtains the better solutions on f_{24}, f_{25}, and f_{27} for 10 dimensions. ABC performs well on f_{23} and f_{28}. For f_{26}, all the algorithms can find the similar solutions except CMA-ES. ABC performs well on f_{23} and f_{28}. DE performs well on f_{29}. DE and CPDE achieve the similar solutions on f_{30}. It can be concluded from Table 7 that all the algorithms can find the similar solutions on f_{23} except BBO. All the algorithms can find the similar solutions on f_{26} except PSO, PSOcf, and CMA-ES. CPDD obtains the better solutions on f_{25} and f_{27}. DE obtains the better solutions on f_{29} and f_{30}. TLBO performs well on f_{24}.

According to the experimental results on thirty test problems from Table 8, we find that CPDD outperforms PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, and BBO on twenty-six, twenty-six, twenty-seven, twenty, eighteen, twenty-five, twenty-three, and twenty-five test problems, respectively, for 10 dimensions. CPDD outperforms PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, and BBO on twenty-eight, twenty-six, twenty-six, twenty, fourteen, twenty-two, twenty-one, and twenty-two test problems, respectively, for 30 dimensions. Moreover, Figures 3 and 4 have further displayed the convergence graphs of different benchmark problems in terms of the mean errors (in logarithmic scale) achieved by each of nine algorithms for CEC2014 problems versus the number of FES for 10 and 30 dimensions.

In summary, it is suggested that CPDD beats PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, and BBO in 15 out of 30 benchmark problems for 10 dimensions. CPDD achieves better performance than other seven algorithms in 14 out of 30 benchmark problems for 30 dimensions. The experiment results reveal that CPDD works well for most benchmark problems. This is due to the data-driven mechanism and the phase transition of matter mechanism which are used in CPDD. The data-driven mechanism makes use of the better control parameters which are found by the last generation to produce new control parameters for next generation. The experiment results indicate that the control parameters which are achieved by the data-driven mechanism are appropriate for most benchmark problems and are helpful for finding better solutions. The cloud particles carry out phase transition according to the extent of evolution. The exploration ability and exploitation ability of the algorithm are dynamically adjusted by the phase transition mechanism. Therefore, it not only can improve the convergence rate of algorithm but also can decrease the risk of premature convergence as much as possible.

4. The Real-World Optimization Problem

In this section, the proposed CPDD algorithm is applied to estimate parameters of a real-world problem. The artificial neural network trained by our CPDD algorithm is a three-layer feed-forward network which includes input units, hidden units, and output units. The basic structure of the proposed scheme is depicted in Figure 5.

In the three-layer feed-forward network, input $X = (x_1, x_2, \ldots, x_i, \ldots, x_n)^T$, output $O = (o_1, o_2, \ldots, o_k, \ldots, o_t)^T$, and desired output $D = (d_1, d_2, \ldots, d_k, \ldots, d_t)^T$. Comparison needs to be made to check out the difference between the test output and real demand. The aim of neural network training is to find a set of weights with the smallest error measure. The objective function is the mean sum of squared errors (MSE) over all training patterns which is shown as follows:

$$\mathrm{MSE} = \frac{1}{Q * K} \sum_{i=1}^{Q} \sum_{j}^{K} \frac{1}{2} \left(d_{ij} - o_{ij}\right)^2, \tag{7}$$

where Q is the number of training data sets, K is the number of output units, d_{ij} is desired output, and o_{ij} is output inferred from neural network.

4.1. SISO Nonlinear Function Approximation. In this example, there are one input unit, five hidden units, and one output unit in the three-layer feed-forward ANN. The model is constructed to model the curve of a nonlinear function which is described by the following equation [21]:

$$y = \sin(2x) \exp(-2x). \tag{8}$$

The sigma function is used as activation function in the output layer. The number (dimension) of the variables is 16

Table 2: Experimental results of PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, BBO, and CPDD on f_4–f_{16} test problems of 10 variables.

F		PSO	PSOcf	TLBO	DE	JADE	CMA-ES	ABC	BBO	CPDD
f_4	F_{mean}	2.19e + 01(−)	3.33e + 01(≈)	1.82e + 01(−)	1.98e + 01(−)	3.02e + 01(≈)	2.91e + 01(≈)	**1.87e − 01**(−)	2.04e + 01(−)	3.26e + 01
	SD	1.84e + 01	2.00e + 01	1.68e + 01	1.73e + 01	1.16e + 01	1.28e + 01	**3.41e − 01**	1.87e + 01	8.29e + 00
	Max	4.06e + 01	1.33e + 02	3.47e + 01	3.47e + 01	3.47e + 01	3.47e + 01	**1.80e + 00**	6.46e + 01	3.47e + 01
	Min	6.69e − 03	1.79e + 01	6.53e − 06	**0.00e + 00**	**0.00e + 00**	**0.00e + 00**	5.67e − 03	7.41e − 03	**0.00e + 00**
f_5	F_{mean}	2.02e + 01(+)	2.00e + 01(+)	1.91e + 01(+)	2.03e + 01(+)	1.70e + 01(+)	1.99e + 01(+)	1.83e + 01(+)	1.93e + 01(+)	**8.89e − 01**
	SD	8.33e − 02	9.03e − 02	4.53e + 00	7.21e − 02	4.51e + 00	**3.75e − 05**	5.02e + 00	3.59e + 00	1.37e + 00
	Max	2.04e + 01	2.02e + 01	2.04e + 01	2.04e + 01	2.00e + 01	1.99e + 01	2.00e + 01	2.00e + 01	**4.55e + 00**
	Min	2.00e + 01	1.99e + 01	3.45e − 06	2.01e + 01	6.52e + 00	1.99e + 01	3.31e + 00	3.33e − 01	**0.00e + 00**
f_6	F_{mean}	1.35e + 00(+)	2.43e + 00(+)	3.50e − 01(+)	5.96e − 02(+)	1.53e − 01(+)	1.37e + 01(+)	2.28e + 00(+)	2.26e + 00(+)	**0.00e + 00**
	SD	1.18e + 00	1.68e + 00	4.21e − 01	1.26e − 01	1.97e − 01	3.75e + 00	4.86e − 01	1.05e + 00	**0.00e + 00**
	Max	4.24e + 00	6.53e + 00	1.23e + 00	8.94e − 01	8.30e − 01	1.91e + 01	3.17e + 00	5.13e + 00	**0.00e + 00**
	Min	9.88e − 03	6.77e − 05	1.13e − 13	**0.00e + 00**	1.91e − 05	4.10e + 00	1.16e + 00	8.78e − 01	**0.00e + 00**
f_7	F_{mean}	1.24e − 01(+)	4.27e + 00(+)	4.42e − 02(+)	2.51e − 01(+)	9.72e − 03(+)	1.06e − 02(+)	8.49e − 03(+)	2.91e − 01(+)	**0.00e + 00**
	SD	5.91e − 02	4.77e + 00	2.46e − 02	1.17e − 01	6.53e − 03	1.00e − 02	8.80e − 03	1.51e − 01	**0.00e + 00**
	Max	2.53e − 01	1.78e + 01	9.34e − 02	4.23e − 01	2.88e − 02	3.20e − 02	3.21e − 02	8.21e − 01	**0.00e + 00**
	Min	1.72e − 02	4.42e − 02	7.39e − 03	8.87e − 03	4.91e − 04	**0.00e + 00**	2.16e − 12	1.15e − 01	**0.00e + 00**
f_8	F_{mean}	1.59e + 00(+)	7.53e + 00(+)	5.45e + 00(+)	1.64e + 01(+)	**0.00e + 00**(≈)	1.47e + 02(+)	**0.00e + 00**(≈)	6.54e − 03(+)	**0.00e + 00**
	SD	9.63e − 01	3.72e + 00	2.89e + 00	2.72e + 00	**0.00e + 00**	4.59e + 01	2.11e − 14	3.57e − 03	**0.00e + 00**
	Max	3.97e + 00	1.62e + 01	1.73e + 01	2.19e + 01	**0.00e + 00**	2.58e + 02	1.13e − 13	1.39e − 02	**0.00e + 00**
	Min	**0.00e + 00**	1.98e + 00	9.94e − 01	8.44e + 00	**0.00e + 00**	6.96e + 01	**0.00e + 00**	1.01e − 03	**0.00e + 00**
f_9	F_{mean}	7.94e + 00(+)	1.34e + 01(+)	6.65e + 00(+)	2.39e + 01(+)	3.30e + 00(+)	1.51e + 02(+)	8.43e + 00(+)	8.44e + 00(+)	**2.70e + 00**
	SD	3.58e + 00	5.08e + 00	2.72e + 00	4.05e + 00	7.48e − 01	4.79e + 01	1.88e + 00	2.82e + 00	**6.77e − 01**
	Max	1.49e + 01	2.38e + 01	1.29e + 01	3.19e + 01	4.84e + 00	3.24e + 02	1.13e + 01	1.40e + 01	**3.44e + 00**
	Min	1.98e + 00	3.97e + 00	2.56e + 00	1.57e + 01	1.36e + 00	7.16e + 01	4.01e + 00	2.00e + 00	**1.00e + 00**

Table 2: Continued.

F		PSO	PSOcf	TLBO	DE	JADE	CMA-ES	ABC	BBO	CPDD
f_{10}	F_{mean}	1.28e + 02(+)	2.73e + 02(+)	2.20e + 02(+)	6.24e + 02(+)	**4.16e − 03**(−)	1.76e + 03(+)	9.71e − 02(−)	1.48e − 01(−)	3.64e + 00
	SD	9.69e + 01	2.06e + 02	1.87e + 02	1.54e + 02	**1.58e − 02**	4.62e + 02	5.68e − 02	6.08e − 02	2.31e + 00
	Max	3.75e + 02	7.02e + 02	8.83e + 02	8.90e + 02	**6.24e − 02**	3.12e + 03	1.91e − 01	2.79e − 01	6.95e + 00
	Min	1.34e + 01	1.57e + 01	1.03e + 01	2.74e + 02	**0.00e + 00**	1.03e + 03	1.56e − 07	3.56e − 02	1.87e − 01
f_{11}	F_{mean}	2.62e + 02(+)	4.04e + 02(+)	8.35e + 02(+)	1.04e + 03(+)	**1.05e + 02**(≈)	1.47e + 03(+)	2.11e + 02(+)	2.21e + 02(+)	1.08e + 02
	SD	2.24e + 02	2.09e + 02	2.18e + 02	1.46e + 02	**6.78e + 01**	3.69e + 02	8.70e + 01	1.32e + 02	8.18e + 01
	Max	9.37e + 02	7.63e + 02	1.36e + 03	1.33e + 03	2.98e + 02	2.20e + 03	3.42e + 02	5.26e + 02	**2.73e + 02**
	Min	1.84e + 01	2.40e + 01	3.27e + 02	7.30e + 02	3.19e + 01	8.27e + 02	4.76e + 01	1.61e + 01	**9.76e + 00**
f_{12}	F_{mean}	4.26e − 01(+)	**1.06e − 01**(−)	1.09e + 00(+)	9.01e − 01(+)	2.53e − 01(−)	1.02e + 00(+)	2.70e − 01(≈)	1.34e − 01(−)	2.90e − 01
	SD	3.23e − 01	**5.10e − 02**	2.30e − 01	1.28e − 01	6.31e − 02	1.31e + 00	5.87e − 02	5.59e − 02	7.30e − 02
	Max	1.15e + 00	**2.06e − 01**	1.45e + 00	1.16e + 00	4.28e − 01	4.35e + 00	3.50e − 01	2.71e − 01	3.99e − 01
	Min	2.15e − 02	1.70e − 02	5.38e − 01	6.86e − 01	1.60e − 01	**1.09e − 02**	1.52e − 01	4.86e − 02	8.60e − 02
f_{13}	F_{mean}	1.29e − 01(+)	2.81e − 01(+)	1.38e − 01(+)	1.45e − 01(+)	8.64e − 02(+)	1.09e − 01(+)	1.19e − 01(+)	2.82e − 01(+)	**7.38e − 02**
	SD	5.81e − 02	8.69e − 02	4.07e − 02	2.62e − 02	**1.54e − 02**	4.51e − 02	2.34e − 02	5.85e − 02	**2.33e − 02**
	Max	2.39e − 01	4.49e − 01	2.22e − 01	2.01e − 01	1.13e − 01	2.27e − 01	1.79e − 01	4.16e − 01	**1.01e − 01**
	Min	3.33e − 02	9.48e − 02	6.40e − 02	1.02e − 01	5.81e − 02	2.85e − 02	5.86e − 02	1.50e − 01	**2.15e − 02**
f_{14}	F_{mean}	1.33e − 01(+)	4.65e − 01(+)	2.66e − 01(+)	1.65e − 01(+)	9.72e − 02(+)	4.56e − 01(+)	1.39e − 01(+)	2.56e − 01(+)	**6.70e − 02**
	SD	4.71e − 02	3.69e − 01	6.42e − 02	3.14e − 02	2.63e − 02	5.36e − 02	2.74e − 02	1.44e − 01	**1.98e − 02**
	Max	2.37e − 01	1.00e + 00	4.19e − 01	2.26e − 01	1.47e − 01	4.99e − 01	2.05e − 01	7.41e − 01	**9.56e − 02**
	Min	5.09e − 02	7.10e − 02	1.66e − 01	8.90e − 02	5.36e − 02	2.86e − 01	7.90e − 02	9.61e − 02	**2.03e − 02**
f_{15}	F_{mean}	8.47e − 01(≈)	2.31e + 00(≈)	1.40e + 00(+)	1.99e + 00(+)	**5.52e − 01**(−)	1.00e + 00(+)	9.36e − 01(+)	1.81e + 00(+)	7.98e − 01
	SD	3.90e − 01	7.84e + 00	3.27e − 01	3.92e − 01	**8.77e − 02**	3.29e − 01	2.18e − 01	6.48e − 01	**1.29e − 01**
	Max	2.49e + 00	4.37e + 01	1.97e + 00	2.54e + 00	**7.25e − 01**	1.78e + 00	1.33e + 00	3.29e + 00	**9.55e − 01**
	Min	3.94e − 01	4.31e − 01	8.44e − 01	8.23e − 01	**3.37e − 01**	4.36e − 01	**2.62e − 01**	6.42e − 01	5.15e − 01
f_{16}	F_{mean}	2.16e + 00(+)	2.39e + 00(+)	2.39e + 00(+)	2.76e + 00(+)	1.60e + 00(+)	4.72e + 00(+)	2.23e + 00(+)	2.25e + 00(+)	**1.00e + 00**
	SD	4.58e − 01	4.58e − 01	2.77e − 01	1.94e − 01	2.44e − 01	**1.73e − 01**	2.40e − 01	4.37e − 01	1.83e − 01
	Max	3.06e + 00	3.23e + 00	2.95e + 00	3.07e + 00	2.10e + 00	4.97e + 00	2.63e + 00	2.85e + 00	**1.22e + 00**
	Min	1.43e + 00	1.54e + 00	1.44e + 00	2.20e + 00	9.74e − 01	4.22e + 00	1.72e + 00	9.31e − 01	**5.36e − 01**

Table 3: Experimental results of PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, BBO, and CPDD on f_4–f_{16} test problems of 30 variables.

F		PSO	PSOcf	TLBO	DE	JADE	CMA-ES	ABC	BBO	CPDD
f_4	F_{mean}	1.66e + 02(+)	5.41e + 02(+)	5.41e + 01(+)	1.16e − 01(+)	1.32e − 01(≈)	2.11e + 00(≈)	1.81e + 01(+)	1.29e + 02(+)	**3.57e − 05**
	SD	4.11e + 01	6.37e + 02	3.11e + 01	7.41e − 02	7.27e − 01	1.15e + 01	2.50e + 01	3.44e + 01	**8.95e − 05**
	Max	2.99e + 02	3.18e + 03	9.54e + 01	3.57e − 01	3.98e + 00	6.34e + 01	7.23e + 01	1.95e + 02	**3.60e − 04**
	Min	7.73e + 01	7.74e + 01	2.54e − 01	1.54e − 03	**5.68e − 14**	**5.68e − 14**	3.66e − 01	7.67e + 01	6.36e − 12
f_5	F_{mean}	2.07e + 01(+)	2.02e + 01(−)	2.09e + 01(+)	2.09e + 01(+)	2.02e + 01(−)	**1.99e + 01**(−)	2.03e + 01(−)	2.01e + 01(−)	2.04e + 01
	SD	1.29e − 01	2.72e − 01	8.08e − 02	5.87e − 02	3.57e − 02	**1.01e − 05**	2.57e − 02	4.88e − 02	7.76e − 02
	Max	2.09e + 01	2.08e + 01	2.09e + 01	2.10e + 01	2.03e + 01	**1.99e + 01**	2.04e + 01	2.02e + 01	2.05e + 01
	Min	2.04e + 01	1.99e + 01	2.06e + 01	2.07e + 01	2.02e + 01	**1.99e + 01**	2.02e + 01	2.00e + 01	2.02e + 01
f_6	F_{mean}	1.27e + 01(+)	1.93e + 01(+)	1.21e + 01(+)	2.96e − 01(+)	9.09e + 00(+)	4.29e + 01(+)	1.50e + 01(+)	1.28e + 01(+)	**0.00e + 00**
	SD	3.19e + 00	4.07e + 00	2.27e + 00	5.63e − 01	3.32e + 00	8.41e + 00	1.48e + 00	2.01e + 00	**0.00e + 00**
	Max	2.04e + 01	2.67e + 01	1.68e + 01	1.84e + 00	1.28e + 01	5.79e + 01	1.66e + 01	1.67e + 01	**0.00e + 00**
	Min	6.84e + 00	1.02e + 01	8.15e + 00	1.13e − 13	0.00e + 00	1.97e + 01	1.06e + 01	9.22e + 00	**0.00e + 00**
f_7	F_{mean}	1.08e − 02(+)	7.30e + 01(+)	6.11e − 02(+)	**0.00e + 00**(≈)	**0.00e + 00**(≈)	1.56e − 03(+)	9.52e − 07(+)	1.04e + 00(+)	**0.00e + 00**
	SD	1.32e − 02	4.48e + 01	2.29e − 01	**0.00e + 00**	4.72e − 14	4.06e − 03	1.82e − 06	1.78e − 02	**0.00e + 00**
	Max	4.42e − 02	2.01e + 02	1.26e + 00	**0.00e + 00**	1.13e − 13	1.23e − 02	6.48e − 06	1.07e + 00	**0.00e + 00**
	Min	3.41e − 13	1.49e + 01	5.68e − 13	**0.00e + 00**	**0.00e + 00**	1.13e − 13	2.01e − 11	9.95e − 01	**0.00e + 00**
f_8	F_{mean}	2.99e + 01(+)	8.25e + 01(+)	5.74e + 01(+)	1.36e + 02(+)	**0.00e + 00**(−)	4.23e + 02(+)	4.54e − 13(−)	5.55e − 01(−)	7.35e + 00
	SD	6.08e + 00	2.79e + 01	1.15e + 01	2.49e + 01	**0.00e + 00**	7.73e + 01	6.62e − 13	2.35e − 01	2.25e + 00
	Max	4.27e + 01	1.68e + 02	7.95e + 01	1.80e + 02	**0.00e + 00**	6.03e + 02	2.95e − 12	1.11e + 00	1.05e + 01
	Min	2.08e + 01	4.79e + 01	3.78e + 01	8.55e + 01	**0.00e + 00**	3.10e + 02	2.27e − 13	1.82e − 01	2.98e + 00
f_9	F_{mean}	6.64e + 01(+)	1.11e + 02(+)	6.31e + 01(+)	1.75e + 02(+)	**2.56e + 01**(−)	6.34e + 02(+)	8.68e + 01(+)	6.01e + 01(+)	3.14e + 01
	SD	1.49e + 01	3.55e + 01	1.32e + 01	1.08e + 02	**4.50e + 00**	9.84e + 01	1.30e + 01	1.75e + 01	6.41e + 00
	Max	1.00e + 02	1.76e + 02	9.45e + 01	1.92e + 01	**3.59e + 01**	8.16e + 02	1.09e + 02	1.05e + 02	3.99e + 01
	Min	4.17e + 01	3.48e + 01	4.17e + 01	1.52e + 02	**1.72e + 01**	4.09e + 02	6.44e + 01	2.99e + 01	1.79e + 01

Table 3: Continued.

F		PSO	PSOcf	TLBO	DE	JADE	CMA-ES	ABC	BBO	CPDD
f_{10}	F_{mean}	8.20e + 02(+)	2.30e + 03(+)	1.23e + 03(+)	5.48e + 03(+)	**7.63e − 03**(−)	4.86e + 03(+)	1.35e + 00(−)	3.85e + 00(−)	3.75e + 01
	SD	2.93e + 02	5.85e + 02	8.55e + 02	5.86e + 02	1.28e − 02	8.99e + 02	7.19e − 01	1.02e + 00	2.15e + 01
	Max	1.39e + 03	3.73e + 03	4.40e + 03	6.26e + 03	4.16e − 02	7.19e + 03	3.58e + 00	6.25e + 00	7.96e + 01
	Min	2.53e + 02	1.21e + 03	2.41e + 02	6.26e + 03	**0.00e + 00**	2.84e + 03	2.01e − 01	2.26e + 00	6.99e + 00
f_{11}	F_{mean}	2.87e + 03(+)	3.42e + 03(+)	6.55e + 03(+)	6.67e + 03(+)	**1.70e + 03**(−)	4.95e + 03(+)	2.21e + 03(−)	2.06e + 03(−)	2.54e + 03
	SD	5.83e + 02	6.26e + 02	3.06e + 02	2.60e + 02	2.16e + 02	8.66e + 02	**1.87e + 02**	3.63e + 02	3.62e + 02
	Max	4.01e + 03	4.92e + 03	6.99e + 03	7.13e + 03	**2.04e + 03**	6.81e + 03	2.56e + 03	2.79e + 03	3.11e + 03
	Min	**1.62e + 03**	2.18e + 03	5.49e + 03	6.14e + 03	**1.08e + 03**	3.03e + 03	1.70e + 03	1.08e + 03	1.86e + 03
f_{12}	F_{mean}	5.86e − 01(+)	2.72e − 01(+)	2.50e + 00(+)	2.51e + 00(+)	2.73e − 01(+)	3.08e − 01(≈)	4.12e − 01(+)	2.34e − 01(≈)	**2.22e − 01**
	SD	2.32e − 01	9.25e − 02	2.50e − 01	3.19e − 01	**3.51e − 02**	2.62e − 01	5.55e − 02	5.74e − 02	5.43e − 02
	Max	1.12e + 00	4.39e − 01	2.91e + 00	3.00e + 00	3.32e − 01	1.36e + 00	4.95e − 01	3.48e − 01	**3.18e − 01**
	Min	2.46e − 01	1.21e − 01	1.87e + 00	1.69e + 00	1.77e − 01	**4.15e − 02**	3.02e − 01	1.06e − 01	1.44e − 01
f_{13}	F_{mean}	4.02e − 01(+)	1.23e + 00(+)	3.87e − 01(+)	3.51e − 01(+)	2.16e − 01(≈)	2.37e − 01(+)	2.44e − 01(+)	5.91e − 01(+)	**2.01e − 01**
	SD	9.77e − 02	9.35e − 01	7.57e − 02	5.13e − 02	3.86e − 02	7.37e − 02	**2.55e − 02**	1.05e − 01	5.11e − 02
	Max	5.74e − 01	3.81e + 00	5.61e − 01	4.74e − 01	3.34e − 01	4.22e − 01	2.87e − 01	8.46e − 01	**2.73e − 01**
	Min	2.46e − 01	3.44e − 01	2.35e − 01	2.67e − 01	1.57e − 01	1.27e − 01	1.98e − 01	4.00e − 01	**9.08e − 02**
f_{14}	F_{mean}	2.98e − 01(+)	2.41e + 01(+)	2.72e − 01(+)	2.76e − 01(+)	2.39e − 01(+)	3.62e − 01(+)	**1.94e − 01**(−)	3.94e − 01(+)	2.18e − 01
	SD	1.40e − 01	2.12e + 01	5.02e − 02	2.85e − 02	2.34e − 02	1.12e − 01	**1.17e − 02**	1.76e − 01	2.61e − 02
	Max	7.86e − 01	7.11e + 01	3.69e − 01	3.28e − 01	2.88e − 01	8.92e − 01	**2.17e − 01**	1.07e + 00	2.53e − 01
	Min	1.70e − 01	3.23e − 01	1.91e − 01	2.27e − 01	2.06e − 01	2.43e − 01	1.75e − 01	2.69e − 01	**1.56e − 01**
f_{15}	F_{mean}	8.34e + 00(+)	1.05e + 03(≈)	9.80e + 00(+)	1.57e + 01(+)	**3.10e + 00**(−)	3.74e + 00(−)	1.05e + 01(+)	1.44e + 01(+)	4.94e + 00
	SD	3.68e + 00	3.61e + 03	4.11e + 00	8.13e − 01	**3.37e − 01**	9.59e − 01	1.57e + 00	3.54e + 00	9.54e − 01
	Max	2.08e + 01	1.97e + 04	2.09e + 01	1.74e + 01	**3.90e + 00**	**5.95e + 00**	1.32e + 01	2.13e + 01	6.19e + 00
	Min	4.11e + 00	2.58e + 00	3.19e + 00	1.41e + 01	**2.27e + 00**	**2.21e + 00**	6.51e + 00	8.45e + 00	2.49e + 00
f_{16}	F_{mean}	1.09e + 01(+)	1.10e + 01(+)	1.17e + 01(+)	1.24e + 01(+)	**9.35e + 00**(−)	1.43e + 01(+)	1.06e + 01(≈)	9.53e + 00(−)	1.03e + 01
	SD	5.57e − 01	6.72e − 01	4.21e − 01	2.05e − 01	3.37e − 01	3.32e − 01	**2.98e − 01**	7.71e − 01	7.86e − 01
	Max	1.18e + 01	1.25e + 01	1.24e + 01	1.28e + 01	**9.97e + 00**	1.47e + 01	1.10e + 01	1.07e + 01	1.14e + 01
	Min	9.58e + 00	9.54e + 00	1.06e + 01	1.20e + 01	8.67e + 00	1.35e + 01	9.61e + 00	6.94e + 00	**8.35e + 00**

Table 4: Experimental results of PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, BBO, and CPDD on f_{17}–f_{22} test problems of 10 variables.

F		PSO	PSOcf	TLBO	DE	JADE	CMA-ES	ABC	BBO	CPDD
f_{17}	F_{mean}	3.20e + 03(+)	5.38e + 03(+)	1.49e + 03(+)	**5.78e − 01**(≈)	1.85e + 00(≈)	5.26e + 02(+)	1.74e + 05(+)	3.13e + 05(+)	9.85e − 01
	SD	2.53e + 03	4.21e + 03	6.26e + 02	8.15e − 01	2.20e + 00	1.94e + 02	1.78e + 05	4.17e + 05	**7.30e − 01**
	Max	9.98e + 03	1.50e + 04	3.82e + 03	4.69e + 00	7.04e + 00	8.93e + 02	6.68e + 05	2.02e + 06	**2.46e + 00**
	Min	3.33e + 02	2.72e + 02	6.11e + 02	4.37e − 10	4.54e − 13	1.51e + 02	1.03e + 04	3.18e + 04	**2.27e − 13**
f_{18}	F_{mean}	5.63e + 03(+)	8.15e + 03(+)	1.04e + 03(+)	1.92e − 01(+)	1.84e − 01(+)	5.99e + 01(+)	6.90e + 02(+)	1.08e + 04(+)	**5.06e − 02**
	SD	8.27e + 03	6.92e + 03	1.18e + 03	1.81e − 01	3.16e − 01	3.49e + 01	4.79e + 02	1.20e + 04	**4.05e − 02**
	Max	2.77e + 04	2.72e + 04	5.03e + 03	4.99e − 01	1.70e + 00	1.50e + 02	1.77e + 03	3.68e + 04	**1.26e − 01**
	Min	2.95e + 00	2.03e + 02	7.83e + 01	1.72e − 02	6.05e − 03	4.09e + 00	1.17e + 02	1.11e + 02	**1.42e − 03**
f_{19}	F_{mean}	1.14e + 00(+)	4.28e + 00(+)	1.19e + 00(+)	4.65e − 01(+)	2.33e − 01(+)	2.77e + 00(+)	5.35e − 01(+)	9.57e − 01(+)	**2.93e − 02**
	SD	8.69e − 01	1.80e + 00	6.07e − 01	1.14e − 01	6.52e − 02	1.13e + 00	2.38e − 01	4.22e − 01	**1.69e − 02**
	Max	2.64e + 00	7.82e + 00	3.01e + 00	7.26e − 01	3.73e − 01	4.90e + 00	1.14e + 00	1.73e + 00	**5.12e − 02**
	Min	4.40e − 02	1.05e + 00	2.94e − 01	2.90e − 01	1.04e − 01	1.02e + 00	1.49e − 01	2.81e − 01	**2.60e − 09**
f_{20}	F_{mean}	5.23e + 02(+)	4.01e + 02(+)	1.84e + 02(+)	8.51e − 02(≈)	3.18e − 01(+)	6.22e + 01(+)	1.62e + 02(+)	6.12e + 03(+)	**8.39e − 02**
	SD	8.83e + 02	4.97e + 02	1.08e + 02	1.60e − 01	1.00e − 01	4.38e + 01	1.27e + 02	7.26e + 03	**3.64e − 02**
	Max	4.31e + 03	1.75e + 03	4.72e + 02	5.55e − 01	5.87e − 01	1.96e + 02	4.25e + 02	2.64e + 04	**1.30e − 01**
	Min	7.84e + 00	4.27e + 00	4.41e + 01	**3.71e − 05**	8.78e − 02	1.02e + 01	9.76e + 00	4.56e + 01	2.96e − 02
f_{21}	F_{mean}	8.88e + 01(+)	2.44e + 02(+)	2.13e + 02(+)	2.67e − 01(+)	1.10e + 00(+)	2.95e + 02(+)	1.09e + 04(+)	5.42e + 04(+)	**7.11e − 02**
	SD	5.43e + 01	1.98e + 02	8.94e + 01	2.42e − 01	1.20e + 00	1.65e + 02	1.01e + 04	8.48e + 04	**1.14e − 01**
	Max	1.83e + 02	8.57e + 02	4.12e + 02	8.12e − 01	4.97e + 00	6.59e + 02	4.48e + 04	3.40e + 05	**3.12e − 01**
	Min	6.74e − 01	1.76e + 01	6.29e + 01	**4.30e − 05**	1.10e − 03	1.79e + 01	5.93e + 02	4.37e + 02	**6.58e − 05**
f_{22}	F_{mean}	1.58e + 01(+)	3.42e + 01(+)	1.66e + 01(+)	2.75e − 01(+)	1.97e − 01(+)	2.25e + 02(+)	9.50e − 01(+)	7.05e + 00(+)	**5.08e − 02**
	SD	8.69e + 00	2.35e + 01	9.03e + 00	2.08e − 01	5.75e − 02	2.24e + 02	1.12e + 00	2.22e + 01	**5.54e − 02**
	Max	2.20e + 01	1.40e + 02	2.98e + 01	7.47e − 01	3.30e − 01	7.31e + 02	4.91e + 00	1.21e + 02	**2.00e − 01**
	Min	1.97e − 02	2.03e + 01	1.25e + 00	3.50e − 02	1.03e − 01	2.01e + 01	7.10e − 02	3.81e − 01	**1.30e − 03**

Table 5: Experimental results of PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, BBO, and CPDD on f_{17}–f_{22} test problems of 30 variables.

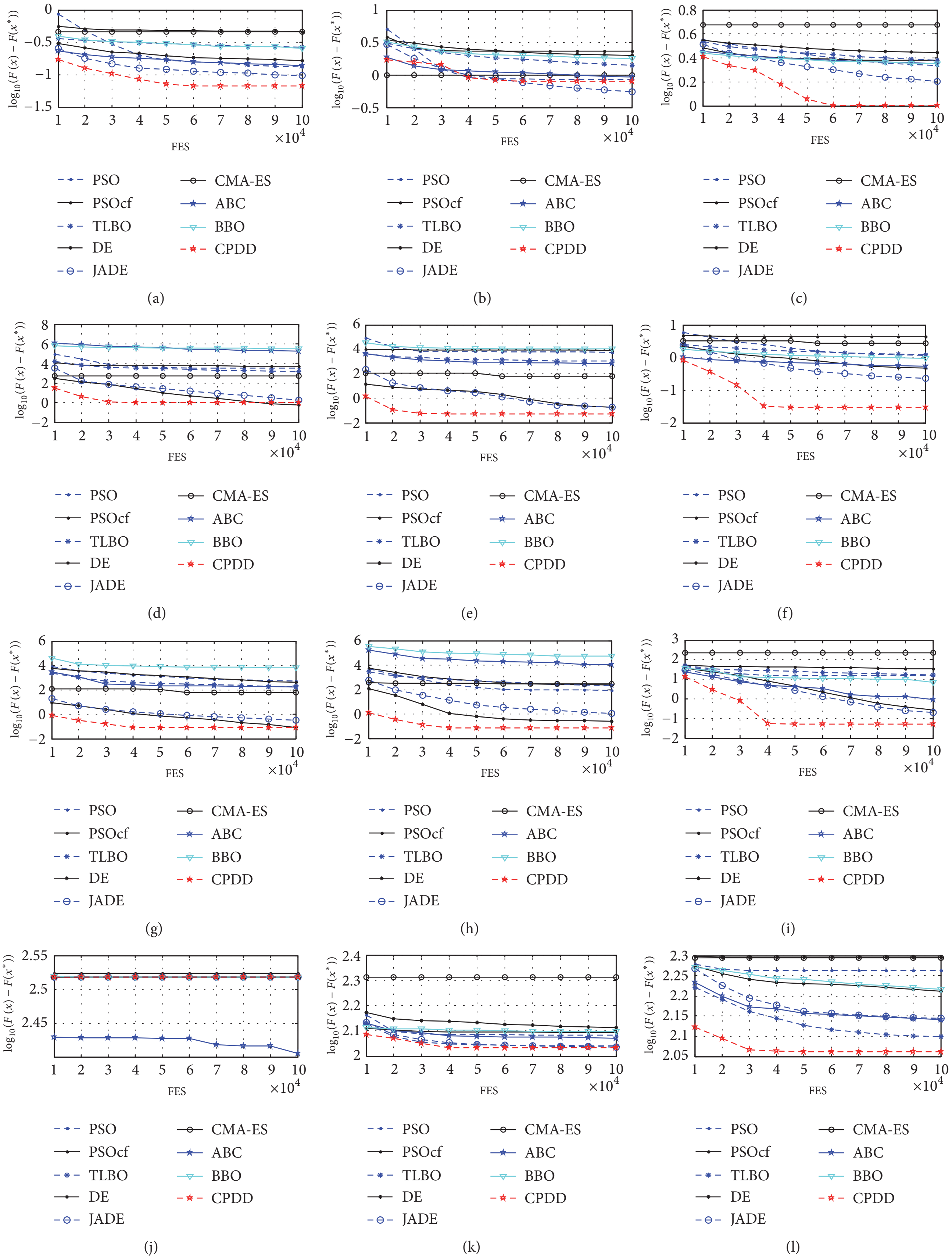
F		PSO	PSOcf	TLBO	DE	JADE	CMA-ES	ABC	BBO	CPDD
f_{17}	F_{mean}	6.01e + 05(+)	1.08e + 06(+)	1.91e + 05(+)	1.35e + 03(+)	1.21e + 03(+)	1.60e + 03(+)	2.82e + 06(+)	1.78e + 06(+)	**1.02e + 02**
	SD	6.73e + 05	1.53e + 06	1.71e + 05	1.86e + 02	3.65e + 02	4.21e + 02	1.29e + 06	1.14e + 06	**4.30e + 01**
	Max	3.04e + 06	6.58e + 06	7.16e + 05	1.66e + 03	1.88e + 03	2.31e + 03	6.11e + 06	4.26e + 06	**1.76e + 02**
	Min	1.96e + 04	4.15e + 04	3.27e + 04	9.02e + 02	6.12e + 02	7.88e + 02	9.93e + 05	2.41e + 05	**3.62e + 01**
f_{18}	F_{mean}	5.59e + 03(+)	3.86e + 07(+)	2.86e + 03(+)	5.47e + 01(+)	8.34e + 01(+)	1.40e + 02(+)	2.58e + 03(+)	2.99e + 03(+)	**6.65e + 00**
	SD	5.03e + 03	1.21e + 07	3.62e + 03	4.64e + 00	2.62e + 01	4.68e + 01	2.04e + 03	2.71e + 03	**1.63e + 00**
	Max	2.53e + 04	5.03e + 08	1.59e + 04	6.54e + 01	1.26e + 02	2.44e + 02	7.25e + 03	1.12e + 04	**8.94e + 00**
	Min	9.78e + 01	4.56e + 02	8.28e + 01	4.58e + 01	3.01e + 01	6.50e + 01	2.20e + 02	3.23e + 02	**3.70e + 00**
f_{19}	F_{mean}	9.13e + 00(+)	3.06e + 01(+)	1.17e + 01(+)	4.43e + 00(+)	4.37e + 00(+)	9.28e + 00(+)	7.06e + 00(+)	2.58e + 01(+)	**2.93e + 00**
	SD	4.97e + 00	3.41e + 01	1.07e + 01	**2.91e − 01**	5.41e − 01	1.60e + 00	6.23e − 01	2.90e + 01	**6.13e − 01**
	Max	3.28e + 01	1.09e + 02	6.73e + 01	5.04e + 00	5.78e + 00	1.21e + 01	8.03e + 00	8.51e + 01	**4.12e + 00**
	Min	4.52e + 00	8.91e + 00	5.49e + 00	3.74e + 00	3.49e + 00	6.13e + 00	5.64e + 00	6.52e + 00	**1.58e + 00**
f_{20}	F_{mean}	3.37e + 02(+)	3.52e + 03(+)	1.05e + 03(+)	3.46e + 01(+)	3.09e + 03(+)	3.07e + 02(+)	5.33e + 03(+)	8.51e + 03(+)	**5.66e + 00**
	SD	1.51e + 02	8.21e + 03	7.76e + 02	8.32e + 00	2.51e + 03	1.11e + 02	2.01e + 03	5.95e + 03	**2.37e + 00**
	Max	7.38e + 02	3.36e + 04	4.68e + 03	4.79e + 01	1.15e + 04	5.43e + 02	9.45e + 03	2.86e + 04	**1.13e + 01**
	Min	6.94e + 01	2.83e + 02	3.23e + 02	6.89e + 00	7.50e + 00	1.31e + 02	1.16e + 03	1.24e + 03	**2.25e + 00**
f_{21}	F_{mean}	7.57e + 04(+)	1.97e + 05(+)	7.22e + 04(+)	6.57e + 02(+)	1.08e + 04(+)	1.06e + 03(+)	4.76e + 05(+)	8.06e + 05(+)	**4.53e + 01**
	SD	1.01e + 05	2.49e + 05	5.74e + 04	1.76e + 02	2.86e + 04	3.76e + 02	2.35e + 05	7.59e + 05	**4.48e + 01**
	Max	5.05e + 05	1.18e + 06	2.89e + 05	1.01e + 03	1.01e + 05	1.75e + 03	1.10e + 06	3.43e + 06	**1.35e + 02**
	Min	3.17e + 03	2.14e + 04	1.50e + 04	2.80e + 02	2.14e + 01	3.91e + 02	1.14e + 05	8.11e + 04	**2.48e + 00**
f_{22}	F_{mean}	3.53e + 02(+)	5.77e + 02(+)	2.38e + 02(+)	3.52e + 01(≈)	1.50e + 02(+)	2.91e + 02(+)	2.85e + 02(+)	3.69e + 02(+)	**2.67e + 01**
	SD	1.33e + 02	2.15e + 02	1.14e + 02	2.83e + 01	6.80e + 01	1.75e + 02	8.25e + 01	1.71e + 02	**6.43e + 00**
	Max	7.09e + 02	1.10e + 03	4.79e + 02	1.77e + 02	3.15e + 02	6.89e + 02	4.31e + 02	8.52e + 02	**4.61e + 01**
	Min	2.87e + 01	2.70e + 02	3.47e + 01	2.44e + 01	3.33e + 01	2.82e + 02	9.26e + 01	1.42e + 02	**2.11e + 01**

Table 6: Experimental results of PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, BBO, and CPDD on f_{23}–f_{30} test problems of 10 variables.

F		PSO	PSOcf	TLBO	DE	JADE	CMA-ES	ABC	BBO	CPDD
f_{23}	F_{mean}	3.29e + 02(≈)	3.34e + 02(+)	3.29e + 02(≈)	3.29e + 02(≈)	3.29e + 02(≈)	3.29e + 02(≈)	**2.54e + 02**(−)	3.29e + 02(≈)	3.29e + 02
	SD	1.20e − 12	1.00e + 01	3.68e − 12	9.25e − 13	9.25e − 13	9.25e − 13	1.15e + 01	1.61e − 02	**2.89e − 13**
	Max	**3.29e + 02**	3.63e + 02	**3.29e + 02**	**3.29e + 02**	**3.29e + 02**	**3.29e + 02**	**3.29e + 02**	3.29e + 02	**3.29e + 02**
	Min	3.29e + 02	3.29e + 02	3.29e + 02	3.29e + 02	3.29e + 02	3.29e + 02	**1.10e + 01**	3.29e + 02	3.29e + 02
f_{24}	F_{mean}	1.21e + 02(+)	1.23e + 02(+)	1.10e + 02(+)	1.29e + 02(+)	1.08e + 02(≈)	2.04e + 02(+)	1.17e + 02(+)	1.25e + 02(+)	**1.07e + 02**
	SD	5.31e + 00	1.07e + 01	4.57e + 00	3.12e + 00	2.15e + 00	1.58e + 02	1.93e + 01	6.82e + 00	**2.43e + 00**
	Max	1.34e + 02	1.58e + 02	1.22e + 02	1.33e + 02	1.12e + 02	1.00e + 03	1.36e + 02	1.40e + 02	**1.09e + 02**
	Min	1.11e + 02	1.10e + 02	**1.00e + 02**	1.20e + 02	**1.00e + 02**	1.15e + 02	1.90e + 01	1.13e + 02	**1.00e + 02**
f_{25}	F_{mean}	1.83e + 02(+)	1.97e + 02(+)	1.25e + 02(+)	1.62e + 02(+)	1.39e + 02(+)	1.97e + 02(+)	1.38e + 02(+)	1.64e + 02(+)	**1.15e + 02**
	SD	3.29e + 01	1.69e + 01	1.58e + 01	3.96e + 01	3.89e + 01	1.44e + 01	8.60e + 00	3.41e + 01	**7.17e + 00**
	Max	2.02e + 02	2.04e + 02	1.77e + 02	2.01e + 02	2.01e + 02	2.04e + 02	1.60e + 02	2.03e + 02	**1.23e + 02**
	Min	1.21e + 02	1.34e + 02	**1.00e + 02**	**1.00e + 02**	**1.00e + 02**	1.35e + 02	1.22e + 02	1.15e + 02	**1.00e + 02**
f_{26}	F_{mean}	**1.00e + 02**(≈)	**1.00e + 02**(≈)	**1.00e + 02**(≈)	**1.00e + 02**(≈)	**1.00e + 02**(≈)	2.14e + 02(+)	**1.00e + 02**(≈)	1.00e + 02(≈)	**1.00e + 02**
	SD	4.73e − 02	1.65e − 01	3.15e − 02	2.65e − 02	1.57e − 02	**1.22e − 02**	1.03e + 00	7.02e − 02	2.28e − 02
	Max	**1.00e + 02**	**1.00e + 02**	**1.00e + 02**	**1.00e + 02**	**1.00e + 02**	5.75e + 02	**1.00e + 02**	1.00e + 02	**1.00e + 02**
	Min	1.00e + 02	1.00e + 02	1.00e + 02	1.00e + 02	1.00e + 02	**8.84e + 01**	9.45e + 01	1.00e + 02	1.00e + 02
f_{27}	F_{mean}	3.03e + 02(+)	2.92e + 02(+)	1.81e + 02(+)	8.82e + 01(+)	4.92e + 01(+)	3.37e + 02(+)	8.06e + 01(+)	3.19e + 02(+)	**1.23e + 00**
	SD	1.06e + 02	1.94e + 02	1.68e + 02	1.60e + 02	1.23e + 02	9.90e + 01	1.42e + 02	1.45e + 02	**3.23e − 01**
	Max	4.04e + 02	4.89e + 02	4.00e + 02	4.00e + 02	4.00e + 02	4.00e + 02	4.00e + 02	4.49e + 02	**2.09e + 00**
	Min	2.46e + 00	1.55e + 00	2.54e + 00	1.53e + 00	1.16e + 00	4.03e + 00	5.01e + 00	3.92e + 00	**7.26e − 01**
f_{28}	F_{mean}	4.70e + 02(+)	4.70e + 02(+)	3.93e + 02(+)	3.80e + 02(≈)	4.01e + 02(+)	1.58e + 03(+)	**3.58e + 02**(≈)	4.15e + 02(+)	3.71e + 02
	SD	7.30e + 01	8.13e + 01	3.18e + 01	3.78e + 01	4.83e + 01	1.26e + 03	5.81e + 01	4.78e + 01	**1.36e + 00**
	Max	6.75e + 02	6.65e + 02	4.80e + 02	4.82e + 02	4.78e + 02	4.58e + 03	3.83e + 02	5.20e + 02	**3.72e + 02**
	Min	3.71e + 02	3.92e + 02	3.69e + 02	3.56e + 02	3.56e + 02	3.72e + 02	**1.04e + 02**	3.58e + 02	3.69e + 02
f_{29}	F_{mean}	4.28e + 05(+)	1.26e + 05	5.93e + 02(+)	**2.18e + 02**(≈)	2.55e + 02(+)	5.77e + 04(+)	3.45e + 02(+)	4.73e + 02(+)	2.22e + 02
	SD	1.00e + 06	9.79e + 04	1.36e + 02	1.54e + 01	4.53e + 01	1.14e + 05	4.71e + 01	2.59e + 02	**7.94e − 01**
	Max	3.62e + 06	2.04e + 06	9.05e + 02	**2.23e + 02**	3.54e + 02	1.72e + 06	4.53e + 02	1.63e + 03	2.25e + 02
	Min	3.36e + 02	3.04e + 02	3.08e + 02	**1.37e + 02**	2.21e + 02	1.44e + 02	2.44e + 02	2.63e + 02	2.21e + 02
f_{30}	F_{mean}	8.09e + 02(+)	8.51e + 02(+)	5.44e + 02(+)	**4.62e + 02**(≈)	4.91e + 02(+)	1.05e + 03(+)	6.32e + 02(+)	7.75e + 02(+)	**4.62e + 02**
	SD	2.62e + 02	3.91e + 02	1.03e + 02	3.64e + 00	3.04e + 01	3.62e + 02	1.16e + 02	1.81e + 02	**1.49e + 00**
	Max	1.52e + 03	2.46e + 03	9.95e + 02	4.80e + 02	5.80e + 02	1.88e + 03	1.01e + 03	1.22e + 03	**4.62e + 02**
	Min	5.09e + 02	4.67e + 02	4.59e + 02	**4.54e + 02**	4.62e + 02	5.50e + 02	5.00e + 02	5.35e + 02	**4.54e + 02**

Table 7: Experimental results of PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, BBO, and CPDD on f_{23}–f_{30} test problems of 30 variables.

F		PSO	PSOcf	TLBO	DE	JADE	CMA-ES	ABC	BBO	CPDD
f_{23}	F_{mean}	**3.15e + 02**(≈)	3.51e + 02(+)	**3.15e + 02**(≈)	**3.15e + 02**(≈)	**3.15e + 02**(≈)	**3.15e + 02**(≈)	**3.15e + 02**(≈)	3.16e + 02(+)	**3.15e + 02**
	SD	2.21e − 01	3.01e + 01	1.78e − 12	1.38e − 12	**1.37e − 12**	5.20e − 12	7.91e − 02	5.95e − 01	1.38e − 12
	Max	3.16e + 02	4.61e + 02	**3.15e + 02**	**3.15e + 02**	**3.15e + 02**	**3.15e + 02**	**3.15e + 02**	3.17e + 02	**3.15e + 02**
	Min	**3.15e + 02**	**3.15e + 02**	**3.15e + 02**	**3.15e + 02**	**3.15e + 02**	**3.15e + 02**	**3.15e + 02**	3.15e + 02	**3.15e + 02**
f_{24}	F_{mean}	2.34e + 02(+)	2.53e + 02(+)	**2.00e + 02**(−)	2.16e + 02(+)	2.24e + 02(+)	2.62e + 02(+)	2.27e + 02(+)	2.32e + 02(+)	2.06e + 02
	SD	8.37e + 00	1.56e + 01	**1.06e − 03**	1.03e + 01	1.09e + 00	1.32e + 02	6.41e − 01	3.73e + 00	1.00e + 01
	Max	2.50e + 02	2.79e + 02	**2.00e + 02**	2.27e + 02	2.28e + 02	9.06e + 02	2.28e + 02	2.48e + 02	2.21e + 02
	Min	2.23e + 02	2.24e + 02	**2.00e + 02**	**2.00e + 02**	2.22e + 02	2.23e + 02	2.25e + 02	2.27e + 02	**2.00e + 02**
f_{25}	F_{mean}	2.09e + 02(−)	2.13e + 02(−)	**2.00e + 02**(−)	2.02e + 02(−)	2.04e + 02(−)	2.04e + 02(−)	2.08e + 02(−)	2.07e + 02(−)	2.18e + 02
	SD	2.81e + 00	8.57e + 00	1.55e + 00	9.80e − 02	1.64e + 00	2.38e + 00	**1.11e + 00**	1.61e + 00	8.25e + 00
	Max	2.17e + 02	2.34e + 02	**2.06e + 02**	2.03e + 02	2.09e + 02	2.12e + 02	2.09e + 02	2.11e + 02	2.23e + 02
	Min	2.05e + 02	2.03e + 02	2.00e + 02	2.02e + 02	2.02e + 02	2.02e + 02	2.05e + 02	2.04e + 02	**2.00e + 02**
f_{26}	F_{mean}	1.27e + 02(+)	1.19e + 02(+)	**1.00e + 02**(≈)	**1.00e + 02**(≈)	**1.00e + 02**(≈)	1.14e + 02(≈)	**1.00e + 02**(≈)	**1.00e + 02**(≈)	**1.00e + 02**
	SD	4.49e + 01	3.70e + 01	1.06e − 01	3.37e − 02	**3.24e − 02**	5.43e + 01	5.63e − 02	1.20e − 01	5.57e − 02
	Max	2.01e + 02	2.01e + 02	**1.00e + 02**	**1.00e + 02**	**1.00e + 02**	3.23e + 02	**1.00e + 02**	**1.00e + 02**	**1.00e + 02**
	Min	**1.00e + 02**	**1.00e + 02**	**1.00e + 02**	**1.00e + 02**	**1.00e + 02**	**1.00e + 02**	**1.00e + 02**	**1.00e + 02**	**1.00e + 02**
f_{27}	F_{mean}	6.49e + 02(+)	8.62e + 02(+)	5.24e + 02(+)	3.45e + 02(+)	3.34e + 02(+)	3.97e + 02(+)	4.09e + 02(+)	6.07e + 02(+)	**3.00e + 02**
	SD	1.27e + 02	2.40e + 02	1.27e + 02	4.84e + 01	4.76e + 01	9.75e + 01	2.74e + 00	1.37e + 02	**1.19e − 13**
	Max	8.55e + 02	1.17e + 03	7.51e + 02	4.00e + 02	4.12e + 02	7.95e + 02	4.14e + 02	8.38e + 02	**3.00e + 02**
	Min	4.05e + 02	4.13e + 02	4.01e + 02	**3.00e + 02**	**3.00e + 02**	3.01e + 02	4.04e + 02	4.02e + 02	**3.00e + 02**
f_{28}	F_{mean}	1.31e + 03(+)	1.70e + 03(+)	1.02e + 03(+)	7.93e + 02(−)	**7.83e + 02**(−)	4.42e + 03(+)	9.63e + 02(+)	9.25e + 02(+)	8.40e + 02
	SD	3.73e + 02	4.47e + 02	1.34e + 02	4.53e + 01	4.69e + 01	2.85e + 03	4.52e + 01	1.14e + 02	**2.06e + 01**
	Max	2.37e + 03	2.99e + 03	1.38e + 03	8.45e + 02	**8.33e + 02**	9.37e + 03	1.09e + 03	1.47e + 03	8.82e + 02
	Min	9.29e + 02	1.04e + 03	8.16e + 02	**6.79e + 02**	6.80e + 02	7.09e + 02	8.60e + 02	7.87e + 02	8.00e + 02
f_{29}	F_{mean}	9.06e + 05(+)	6.34e + 06(+)	1.45e + 06(+)	**7.20e + 02**(≈)	7.88e + 02(≈)	7.89e + 02(+)	1.08e + 03(+)	1.84e + 03(+)	7.30e + 02
	SD	1.95e + 06	6.24e + 06	3.30e + 06	**1.88e + 01**	2.26e + 02	7.02e + 01	6.67e + 01	4.51e + 02	2.56e + 01
	Max	2.71e + 07	2.68e + 07	9.18e + 06	**8.18e + 02**	1.64e + 03	1.00e + 03	1.20e + 03	2.95e + 03	8.53e + 02
	Min	**5.34e + 02**	3.96e + 04	8.84e + 02	7.14e + 02	7.15e + 02	5.81e + 02	8.80e + 02	1.26e + 03	7.14e + 02
f_{30}	F_{mean}	6.59e + 03(+)	1.19e + 05(+)	3.14e + 03(+)	**5.82e + 02**(−)	1.84e + 03(+)	2.16e + 03(+)	4.08e + 03(+)	6.09e + 03(+)	9.74e + 02
	SD	9.18e + 03	8.99e + 04	1.64e + 03	2.23e + 02	7.33e + 02	5.14e + 02	7.43e + 02	2.61e + 03	**2.12e + 02**
	Max	5.23e + 04	3.29e + 05	7.00e + 03	1.37e + 03	3.74e + 03	3.14e + 03	5.77e + 03	1.13e + 04	**1.30e + 03**
	Min	2.47e + 03	4.96e + 03	7.81e + 03	**3.96e + 02**	8.25e + 02	1.22e + 03	2.21e + 03	2.31e + 03	4.94e + 02



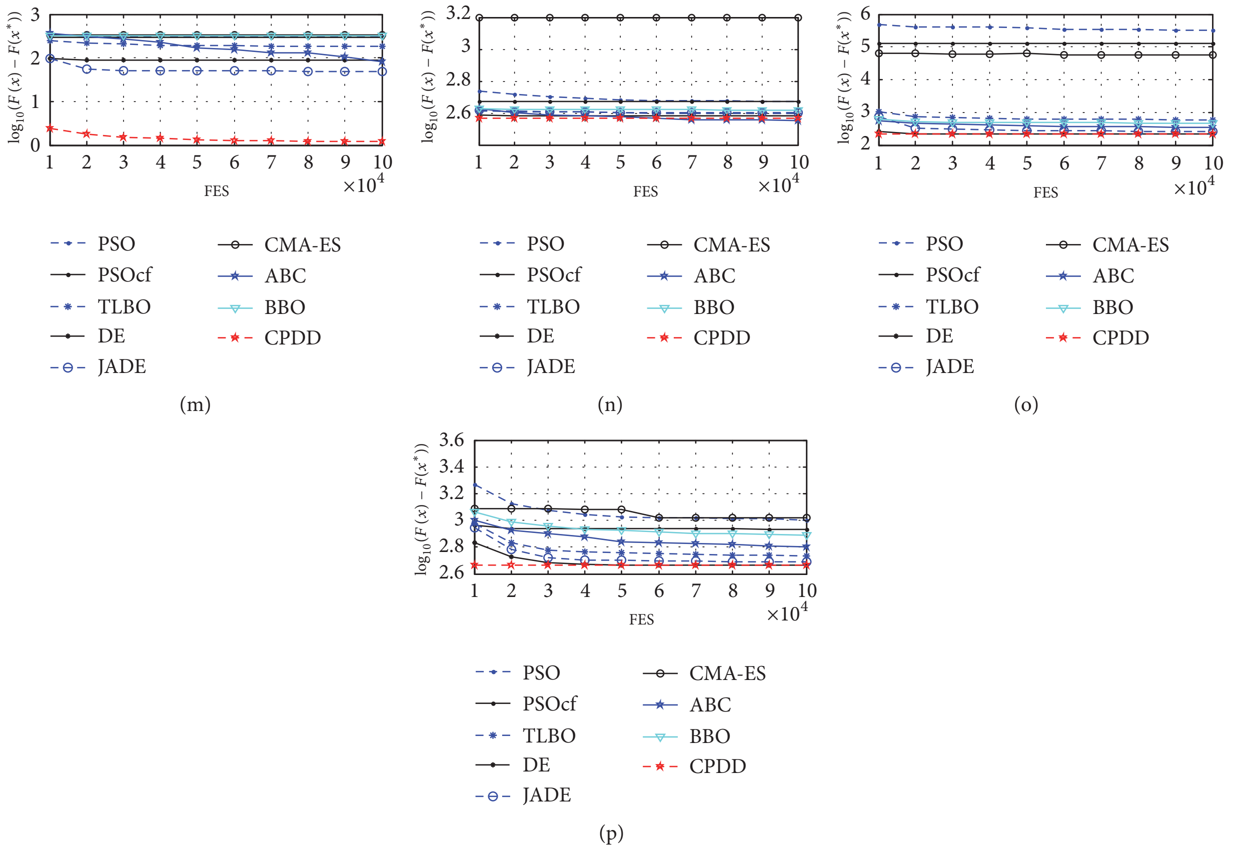
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 3: Continued.



(m)

(n)

(o)

(p)

Figure 3: Evolution of the mean function error values derived from PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, BBO, and CPDD versus the number of FES on sixteen test problems of ten variables. (a) f_{14}. (b) f_{15}. (c) f_{16}. (d) f_{17}. (e) f_{18}. (f) f_{19}. (g) f_{20}. (h) f_{21}. (i) f_{22}. (j) f_{23}. (k) f_{24}. (l) f_{25}. (m) f_{27}. (n) f_{28}. (o) f_{29}. (p) f_{30}.

Table 8: Comparison of CPDD with PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, and BBO on the CEC2014 benchmarks (D = 10 and 30 dimensions).

D		PSO	PSOcf	TLBO	DE	JADE	CMA-ES	ABC	BBO
10	+	26	26	27	20	18	25	23	25
	−	1	1	1	1	3	0	3	3
	≈	3	3	2	9	9	5	4	2
30	+	28	26	26	20	14	22	21	22
	−	1	2	2	3	9	4	6	6
	≈	1	2	2	7	7	4	3	2

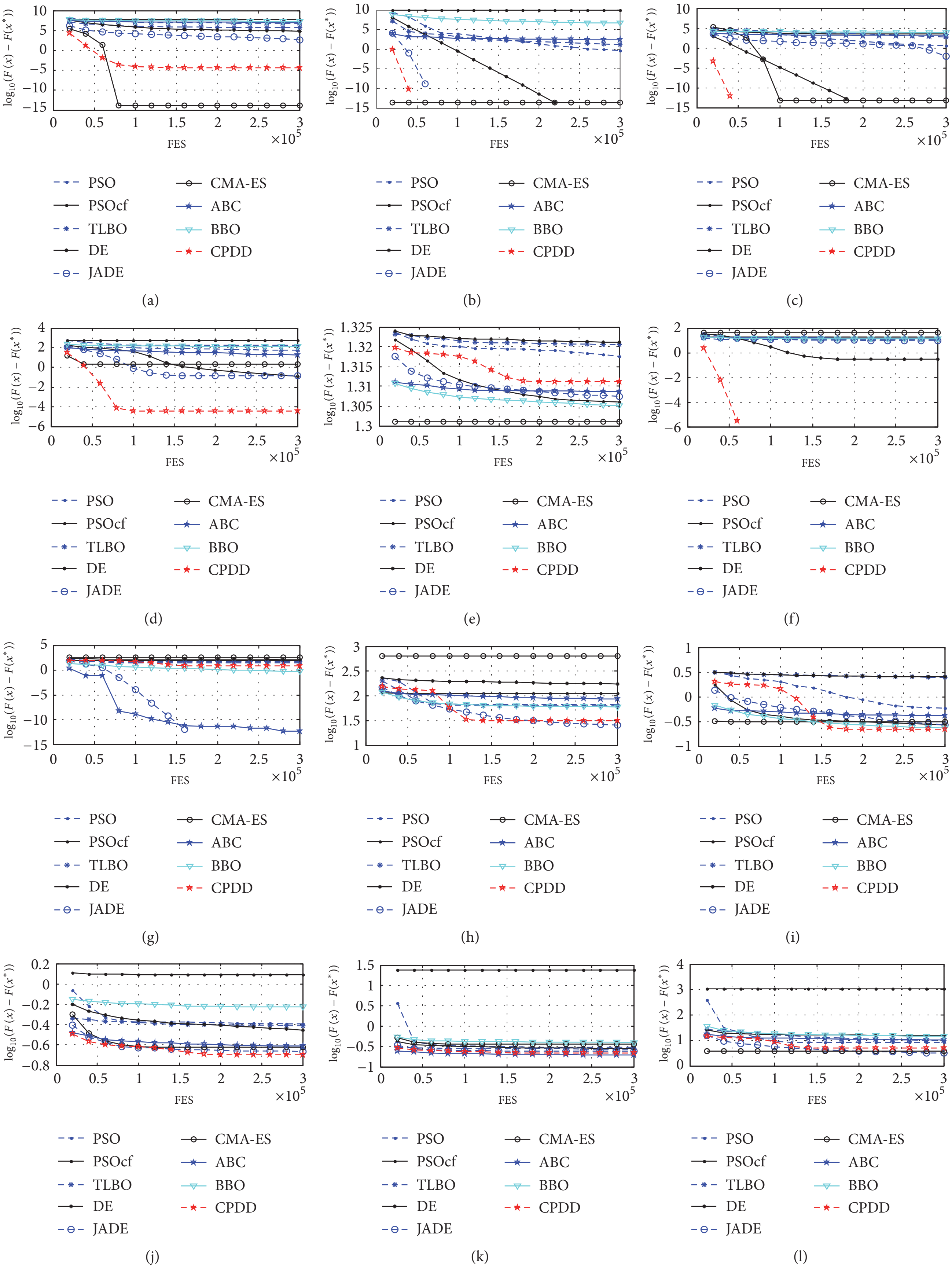
for CPDD-based ANN. In order to train the ANN, 200 pairs of data are chosen from the real model. The population size is set as 50; the maximal number of function evaluations (FES = 10000) is used as ended condition of each algorithm. To assess the performance of each algorithm in noise, 30 db additive with Gaussian noise is added to the experiment. For each algorithm, 50 runs are performed. The other parameters are same as those of the previous investigations.

Table 9 shows that CPDD performs better than other compared algorithms in terms of the mean MSE and the standard deviation of MSE. The approximation curves for training and test using different algorithms are shown in Figure 6. It indicates that CPDD outperforms other algorithms for training the model.

4.2. Lorenz Chaotic Time Series Prediction. In this example, there are three input units, five hidden units, and one output unit in the three-layer feed-forward ANN. It is used to forecast a Lorenz chaotic time series which is described by the following equation [51]:

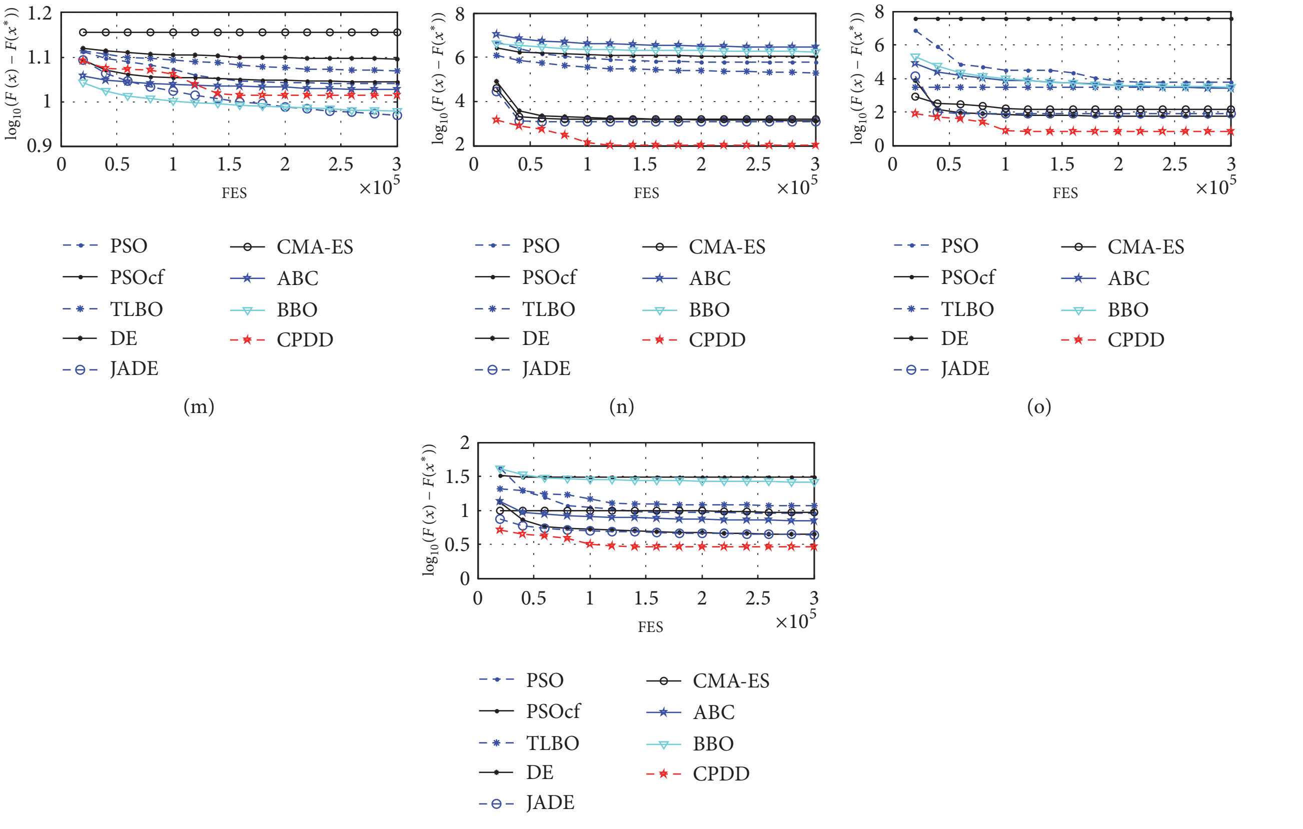
$$\begin{aligned}\frac{dx}{dt} &= a(y-x)\\ \frac{dy}{dt} &= bx - xz + cy\\ \frac{dz}{dt} &= xy - dz,\end{aligned} \tag{9}$$

where $a = 10$, $b = 28$, and $c = 8/3$.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figure 4: Continued.



(m)

(n)

(o)

(p)

Figure 4: Evolution of the mean function error values derived from PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, BBO, and CPDD versus the number of FES on sixteen test problems of thirty variables. (a) f_1. (b) f_2. (c) f_3. (d) f_4. (e) f_5. (f) f_6. (g) f_8. (h) f_9. (i) f_{12}. (j) f_{13}. (k) f_{14}. (l) f_{15}. (m) f_{16}. (n) f_{17}. (o) f_{18}. (p) f_{19}.

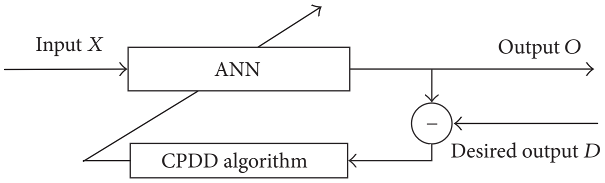


Figure 5: CPDD-based ANN.

In order to train the ANN, 10000 pieces of data are selected according to the real model, among which the first 8000 pieces of data are discarded; the rest normalized 2000 pieces of data are selected as experiment data. The former 1500 points are chosen as the training data points, and the rest 500 points are chosen as the testing data points to test the validity of the model. The simulation goal is to build the single-step-ahead prediction model of chaotic time series as the following form [51]:

$$\hat{x}(t+1) = f(x(t-6), x(t-3), x(t)). \tag{10}$$

In our experiment, population size is set as 50; the maximal number of function evaluations (FES = 10000) is used as ended condition of each algorithm. The results are shown in Table 10 in terms of the mean MSE and the standard deviation obtained in the 50 independent runs for nine optimization algorithms.

Table 10 indicates that CPDD shows better performance than other compared methods in terms of the mean MSE and the standard deviation. Figure 7 shows the prediction of Lorenz chaotic time series for training and test with different optimization algorithm. The curves show that the prediction obtained by CPDD performs better than other algorithms. The experimental results represent that CPDD has better prediction performance for Lorenz chaotic time series compared with other optimization algorithms.

5. Conclusion

A new metaheuristic optimization algorithm CPDD, which is inspired from the phenomenon of cloud transformation and the transition of matter state, is proposed in this paper. Data-driven mechanism is introduced into differential cloud particles evolution algorithm and it is applied to 30 benchmark problems from the CEC2014 Special Session on Real-Parameter Single Objective Optimization benchmark suite. The experimental results showed that phase transition and data-driven mechanism can not only balance the exploration and exploitation capacity of CPDD but also accelerate the convergence rate. It can be also concluded that CPDD shows

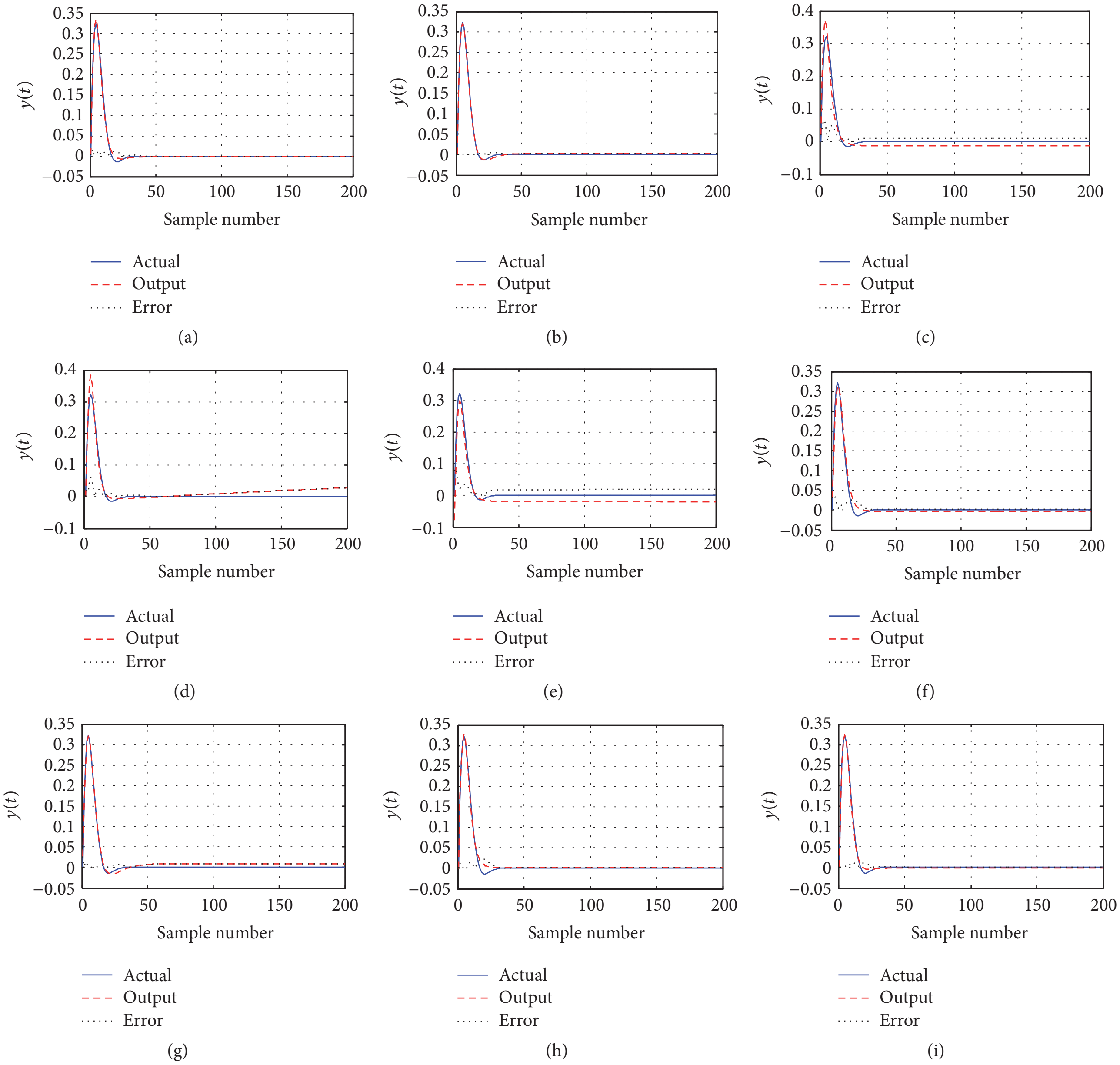
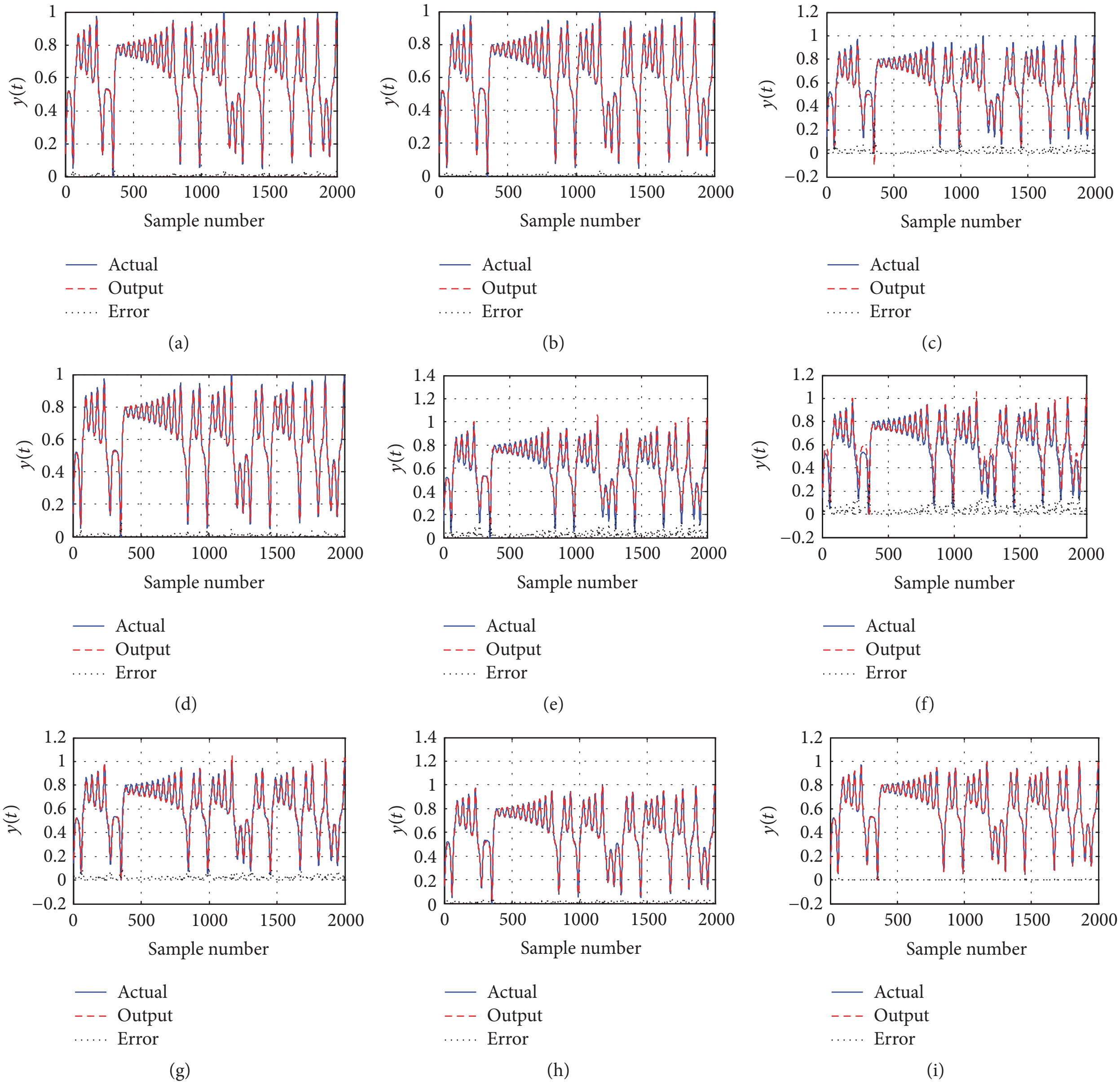


Figure 6: Comparison of the performance curves using different algorithm. (a)~(i) Function approximation with noise using PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, BBO, and CPDD.

Table 9: Comparisons between PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, BBO, and CPDD on MSE.

Algorithm	Training error		Testing error	
	Mean	Std.	Mean	Std.
PSO	4.52*e* − 04	8.25*e* − 04	1.54*e* − 02	5.44*e* − 02
PSOcf	3.38*e* − 04	7.77*e* − 04	4.52*e* − 04	1.19*e* − 03
TLBO	3.41*e* − 03	2.45*e* − 03	2.68*e* − 03	2.22*e* − 03
DE	3.31*e* − 03	1.59*e* − 03	4.31*e* − 03	2.31*e* − 03
JADE	1.15*e* − 03	7.90*e* − 04	1.07*e* + 00	2.78*e* + 00
CMA-ES	8.65*e* − 03	1.49*e* − 02	5.05*e* − 03	9.27*e* − 03
ABC	1.32*e* − 03	1.32*e* − 03	1.53*e* − 02	4.03*e* − 02
BBO	3.13*e* − 03	7.46*e* − 03	8.65*e* − 02	2.47*e* − 01
CPDD	7.31*e* − 05	9.01*e* − 05	3.70*e* − 04	8.54*e* − 04



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 7: Comparison of the performance curves using different algorithm. (a)~(i) Time series prediction using PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, BBO, and CPDD.

Table 10: Comparisons between PSO, PSOcf, TLBO, DE, JADE, CMA-ES, ABC, BBO, and CPDD on MSE.

Algorithm	Training error		Testing error	
	Mean	Std.	Mean	Std.
PSO	$4.83e-04$	$3.36e-04$	$5.02e-04$	$3.50e-04$
PSOcf	$1.11e-03$	$9.93e-04$	$1.16e-03$	$1.05e-03$
TLBO	$4.88e-03$	$2.32e-03$	$5.00e-03$	$2.45e-03$
DE	$4.54e-03$	$3.71e-03$	$4.69e-03$	$3.80e-03$
JADE	$2.00e-03$	$8.51e-04$	$6.78e-01$	$2.87e+00$
CMA-ES	$3.14e-02$	$3.83e-02$	$3.17e-02$	$3.72e-02$
ABC	$3.61e-03$	$1.76e-03$	$3.76e-03$	$1.87e-03$
BBO	$1.50e-03$	$1.19e-03$	$1.57e-03$	$1.26e-03$
CPDD	$2.90e-04$	$2.44e-04$	$3.00e-04$	$2.55e-04$

better performance on artificial neural network training compared with other optimization algorithms.

However, these methods show poor performance on composite problems. This indicates that the control parameters produced by data-driven mechanism are not appropriate for these problems. Therefore, the exploitation of CPDD is poor on these problems. How to improve the exploitation ability of CPDD will need to be further gone into. It is necessary to introduce some new techniques in CPDD for improving exploration and exploitation ability to solve hybrid composition problems. In addition, CPDD will be used to solve real-world engineering problems.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research is partly supported by the Doctoral Foundation of Xi'an University of Technology (112-451116017).

References

[1] J. O. Kephart, "Learning from nature," *Science*, vol. 331, no. 6018, pp. 682-683, 2011.

[2] W. Banzhaf, G. Beslon, S. Christensen et al., "Guidelines: from artificial evolution to computational evolution: a research agenda," *Nature Reviews Genetics*, vol. 7, no. 9, pp. 729–735, 2006.

[3] R. P. Parouha and K. N. Das, "A memory based differential evolution algorithm for unconstrained optimization," *Applied Soft Computing Journal*, vol. 38, pp. 501–517, 2016.

[4] H. A. Taboada, J. F. Espiritu, and D. W. Coit, "MOMS-GA: A multi-objective multi-state genetic algorithm for system reliability optimization design problems," *IEEE Transactions on Reliability*, vol. 57, no. 1, pp. 182–191, 2008.

[5] B. Liu, Q. Zhang, F. V. Fernandez, and G. G. E. Gielen, "An efficient evolutionary algorithm for chance-constrained bi-objective stochastic optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 6, pp. 786–796, 2013.

[6] C. Segura, C. A. CoelloCoello, and A. G. Hernández-Díaz, "Improving the vector generation strategy of differential evolution for large-scale optimization," *Information Sciences*, vol. 323, pp. 106–129, 2015.

[7] E. Carreño Jara, "Multi-objective optimization by using evolutionary algorithms: The p-optimality criteria," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 167–179, 2014.

[8] C.-L. Yang, R. J. Kuo, C.-H. Chien, and N. T. P. Quyen, "Non-dominated sorting genetic algorithm using fuzzy membership chromosome for categorical data clustering," *Applied Soft Computing Journal*, vol. 30, pp. 113–122, 2015.

[9] S.-H. Cheng, S.-M. Chen, and W.-S. Jian, "Fuzzy time series forecasting based on fuzzy logical relationships and similarity measures," *Information Sciences. An International Journal*, vol. 327, pp. 272–287, 2016.

[10] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Universityof Michigan Press, The MIT Press, London, UK, 1975.

[11] 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.

[12] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (Perth, Australia)*, pp. 1942–1948, IEEE Service Center, Piscataway, NJ, USA, 1995.

[14] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the Congress on Evolutionary Computation*, pp. 81–86, IEEE Press, New York, NY, USA, May 2001.

[15] B. Basturk and D. Karaboga, "An artifical bee colony (ABC) algorithm for numeric function optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium*, pp. 12–14, Indianapolis, Ind, USA, 2006.

[16] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[17] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems," *Information Sciences. An International Journal*, vol. 183, pp. 1–15, 2012.

[18] R. V. Rao, "Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 7, no. 1, pp. 19–34, 2016.

[19] Q.-K. Pan, "An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling," *European Journal of Operational Research*, vol. 250, no. 3, pp. 702–714, 2016.

[20] J. A. Delgado-Osuna, M. Lozano, and C. García-Martínez, "An alternative artificial bee colony algorithm with destructive-constructive neighbourhood operator for the problem of composing medical crews," *Information Sciences*, vol. 326, pp. 215–226, 2016.

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

[22] S. Kirkpatrick, J. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *American Association for the Advancement of Science. Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[23] V. Černý, "Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.

[24] A. Y. S. Lam and V. O. K. Li, "Chemical-reaction-inspired metaheuristic for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 381–399, 2010.

[25] Y. H. Shi, "Brain storm optimization algorithm," in *Advances in Swarm Intelligence: Second International Conference, ICSI 2011, Chongqing, China, June 12–15, 2011, Proceedings, Part I*, vol. 6728 of *Lecture Notes in Computer Science*, pp. 303–309, Springer, Berlin, Germany, 2011.

[26] W. Guo, W. Li, Q. Zhang, L. Wang, Q. Wu, and H. Ren, "Biogeography-based particle swarm optimization with fuzzy

elitism and its applications to constrained engineering problems," *Engineering Optimization*, vol. 46, no. 11, pp. 1465–1484, 2014.

[27] C.-H. Mu, J. Xie, Y. Liu, F. Chen, Y. Liu, and L.-C. Jiao, "Memetic algorithm with simulated annealing strategy and tightness greedy optimization for community detection in networks," *Applied Soft Computing Journal*, vol. 34, pp. 485–501, 2015.

[28] S. Cheng, Y. H. Shi, Q. D. Qin, T. O. Ting, and R. Bai, "Maintaining population diversity in brain storm optimization algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '14)*, pp. 3230–3237, July 2014.

[29] X. Zhang and S. Y. Yuen, "A directional mutation operator for differential evolution algorithms," *Applied Soft Computing Journal*, vol. 30, pp. 529–548, 2015.

[30] L. Cui, G. Li, Q. Lin, J. Chen, and N. Lu, "Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations," *Computers and Operations Research*, vol. 67, pp. 155–173, 2016.

[31] J. Q. 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.

[32] 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.

[33] 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.

[34] Y. Wang, Z. X. Cai, and Q. F. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011.

[35] A. Ghosh, S. Das, A. Chowdhury, and R. Giri, "An improved differential evolution algorithm with fitness-based adaptation of the control parameters," *Information Sciences. An International Journal*, vol. 181, no. 18, pp. 3749–3765, 2011.

[36] G. H. Xiang, L. Y. Nan, L. J. Ling, S. Han, W. D. Yun, and C. X. Hong, "Differential evolution improved with self-adaptive control parameters based on simulated annealing," *Swarm and Evolutionary Computation*, vol. 19, pp. 52–67, 2014.

[37] Q. Fan and X. Yan, "Self-adaptive differential evolution algorithm with discrete mutation control parameters," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1551–1572, 2015.

[38] L. d. Coelho, H. V. Ayala, and V. C. Mariani, "A self-adaptive chaotic differential evolution algorithm using gamma distribution for unconstrained global optimization," *Applied Mathematics and Computation*, vol. 234, pp. 452–459, 2014.

[39] X. Lu, K. Tang, B. Sendhoff, and X. Yao, "A new self-adaptation scheme for differential evolution," *Neurocomputing*, vol. 146, pp. 2–16, 2014.

[40] A. R. Jordehi, "Enhanced leader PSO (ELPSO): a new PSO variant for solving global optimisation problems," *Applied Soft Computing Journal*, vol. 26, pp. 401–417, 2015.

[41] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Applied Mathematics and Computation*, vol. 274, pp. 292–305, 2016.

[42] W. Fang, J. Sun, H. Chen, and X. Wu, "A decentralized quantum-inspired particle swarm optimization algorithm with cellular structured population," *Information Sciences*, vol. 330, pp. 19–48, 2016.

[43] M. Taherkhani and R. Safabakhsh, "A novel stability-based adaptive inertia weight for particle swarm optimization," *Applied Soft Computing Journal*, vol. 38, pp. 281–295, 2016.

[44] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[45] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '00)*, vol. 1, pp. 84–88, July 2000.

[46] J. Sun, X. Wu, V. Palade, W. Fang, C.-H. Lai, and W. Xu, "Convergence analysis and improvements of quantum-behaved particle swarm optimization," *Information Sciences. An International Journal*, vol. 193, pp. 81–103, 2012.

[47] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: a survey," *ACM Computing Surveys*, vol. 1, no. 1, pp. 1–33, 2011.

[48] W. Li, "A modified cloud particles differential evolution algorithm for real-parameter optimization," *Algorithms*, vol. 9, no. 4, pp. 1–19, 2016.

[49] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization," Tech. Rep., Zhengzhou University and Nanyang Technological University, 2013.

[50] P. N. Suganthan, N. Hansen, J. J. Liang et al., "Problem definitions and evaluation criteria for the CEC2005 special session on real-parameter optimization," 2005, http://www.ntu.edu.sg/home/EPNSugan.

[51] V. A. Gromov and A. N. Shulga, "Chaotic time series prediction with employment of ant colony optimization," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8474–8478, 2012.