Research Article

Cloud Particles Differential Evolution Algorithm: A Novel Optimization Method for Global Numerical Optimization

Wei Li,¹ Lei Wang,¹ Quanzhu Yao,¹ Qiaoyong Jiang,¹ Lei Yu,¹ Bin Wang,¹ and Xinhong Hei²

¹School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China

²Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an 710048, China

Correspondence should be addressed to Lei Wang; wangleeei@163.com

Received 2 August 2015; Revised 22 October 2015; Accepted 1 November 2015

Academic Editor: Carla Roque

Copyright © 2015 Wei Li 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.

We propose a new optimization algorithm inspired by the formation and change of the cloud in nature, referred to as Cloud Particles Differential Evolution (CPDE) algorithm. The cloud is assumed to have three states in the proposed algorithm. Gaseous state represents the global exploration. Liquid state represents the intermediate process from the global exploration to the local exploitation. Solid state represents the local exploitation. The best solution found so far acts as a nucleus. In gaseous state, the nucleus leads the population to explore by condensation operation. In liquid state, cloud particles carry out macrolocal exploitation by liquefaction operation. A new mutation strategy called cloud differential mutation is introduced in order to solve a problem that the misleading effect of a nucleus may cause the premature convergence. In solid state, cloud particles carry out microlocal exploitation by solidification operation. The effectiveness of the algorithm is validated upon different benchmark problems. The results have been compared with eight well-known optimization algorithms. The statistical analysis on performance evaluation of the different algorithms on 10 benchmark functions and CEC2013 problems indicates that CPDE attains good performance.

1. Introduction

Nowadays, there are many problems which can be treated as optimization problems in real-world application. These problems are always complex because they are multimodal, high dimensional, discontinuous, multiobjective, dynamic, and so forth. Traditional optimization methods often have poor results for some problems. Therefore, intelligent optimization algorithms are designed by imitating natural evolution. These algorithms which are called artificial evolution (AE) [1] solve the difficult design and optimization problems by building solutions that are more fit relative to desired properties. Many results show that the intelligent optimization algorithms are often superior to traditional optimization methods for solving practical optimization problems. Popular intelligent optimizations include Particle Swarm Optimization (PSO) [2], differential evolution (DE) [3], artificial bee colony algorithm (ABC) [4], artificial immune system (AIS) [5], and ant colony algorithm (ACO) [6].

Particle Swarm Optimization (PSO) is a bionic intelligent computing method proposed by Kennedy and Eberhart which imitates the flying and the foraging behavior of birds. The PSO's concept is based on the movement of particles and their personal and best individual experiences [2]. PSO has many advantages such as fast convergence speed, high quality solutions, and good robustness in multidimensional space function optimization, especially for engineering applications. In PSO, the particles adjust their speed and position dynamically by sharing information and experiences of the best particles. The information sharing strategy reduces the diversity of the particle swarm, because all particles except itself only share the optimal particle information while ignoring other particles information. Therefore, the algorithm is easy to fall into premature convergence, low search accuracy, and low efficiency in later iterations for solving complex problems. Researchers have studied and proposed many improvement strategies for solving these problems. Clerc and Kennedy proposed PSO with constriction factor (PSOcf) [7]

based on PSO by studying a particle's trajectory as it moves in discrete time. In PSOf, a five-dimensional depiction is developed for controlling the explosion that results from randomness in the system. Coefficients are applied to various parts of the formula in order to guarantee convergence, while encouraging exploration. Then, Mendes et al. proposed fully informed Particle Swarm Optimization (FIPS) and PSOWFIPS in [8]. FIPS is the fully informed particle swarm with W returning a constant, that is, where all contributions have the same value. PSOWFIPS is a full informed swarm where the contribution of each neighbor was weighted by the goodness of its previous best. Parsopoulos and Vrahatis proposed Unified Particle Swarm Optimization (UPSO) [9] based on PSO. The algorithm merges the local and global variants of PSO to improve their exploration and exploitation abilities without imposing additional requirements in terms of function evaluations. By employing multiple swarms to optimize different components of the solution vector cooperatively, a Cooperative Approach to Particle Swarm Optimization is proposed in [10] and is named as the cooperative particle swarm optimizer (CPSO). H-PSOA [11] is a hierarchical version of the Particle Swarm Optimization (PSO) metaheuristic. The algorithm uses a dynamic hierarchy to define a neighborhood structure. For dealing with different complex situations, self-learning particle swarm optimizer (SLPSO) is proposed in [12]. In SLPSO, each particle has a set of four strategies. The cooperation of the four strategies can enable a particle to choose the optimal strategy according to its own local fitness landscape. A PSO with an aging leader and challengers (ALC-PSO) [13] is proposed to overcome the problem of premature convergence without significantly impairing the fast-converging feature of PSO. The idea of the algorithm is to assign the leader of the swarm with a growing age and a lifespan and to allow the other individuals to challenge the leadership when the leader becomes aged. By improving the current existing teaching-learning-based optimization (TLBO) framework, a teaching and peer-learning PSO (TPLPSO) and a bidirectional teaching and peer-learning PSO (BTPLPSO) are proposed in [14, 15]. BTPLPSO uses two learning phases, namely, the teaching and peer-learning phases to solve the problems such as the particles fail to improve their fitness during the searching process. Recently, Jordehi [16] proposed enhanced leader PSO (ELPSO) based on a five-staged successive mutation strategy. Comprehensive learning particle swarm optimizer (CLPSO) [17] uses a novel learning strategy whereby all other particles' historical best information is used to update a particle's velocity. Dynamic Multiswarm Particle Swarm Optimization (DMS-PSO) [18] was constructed based on the local version of PSO. The whole population is divided into a large number of subswarms. These swarms use their own particles to search for better solutions.

Differential evolution (DE) based on individual differences, introduced by Storn and Price, is a simple evolutionary algorithm [3]. It is an effective, robust, and simple global optimization algorithm which only has a few control parameters [19]. In DE, an intermediate population is produced by restructuring the difference between the individuals of the current population. Then, the algorithm uses the parent

FIGURE 1: Cloud particles transformation process.

FIGURE 2: Three states change of water.

population and offspring population to guide the optimization process. The biggest difference between differential evolution algorithm and other evolutionary algorithms is the mutation operation. At the beginning of the evolution, the difference between the mutation individual and random individual is larger. Therefore, the global search ability is relatively strong. In the later evolution, there is little difference between individuals and the local search ability is improved. According to the experimental studies and theoretical analyses, the control parameter settings are important for the performance of differential evolution. Inappropriate control parameters may lead to premature convergence or stagnation in the evolutionary process. Therefore, many researchers have studied the choices of control parameters and mutation strategies for improving the performance of DE. Brest et al. proposed self-adapting control parameters in differential evolution (jDE) [20]. The parameter control technique of jDE is based on the self-adaptation of two parameters (F and CR), associated with the evolutionary process. Jingqiao and Arthur [21] proposed adaptive differential evolution with optional external archive (JADE). In JADE, a set of

FIGURE 3: Cloud gaseous phase.

FIGURE 4: Cloud liquid phase.

FIGURE 5: Cloud solid phase.

FIGURE 6: Optimization process of CPDE model.

recently explored inferior solutions is considered instead of the best solutions previously explored. These solutions are used to guide the optimum direction and to improve the population diversity. Moreover, a new greedy mutation strategy is introduced, “DE/current-to-pbest” with optional external archive. In addition, the proposed crossover rate repair technique is also incorporated into a self-adaptive DE (SaDE) [22] and an ensemble of mutation strategies and control parameters with the DE (EPSDE) [23]. Recently, researchers have made a number of improvements in DE. A binary learning differential evolution (BLDE) algorithm [24], which is inspired by the learning mechanism in Particle Swarm Optimization (PSO) algorithms, can efficiently locate the global optimal solutions by learning from the last population. Inspired by the phenomenon of fireworks explosion, an improved fireworks optimization method is proposed in [25]. An enhanced self-adaptive differential evolution (ESADE) [26] is proposed to solve the problem that users always have to set the control parameters of DE according to every single different problem. A self-adaptive DE algorithm with discrete mutation control parameters (DMPSADE) [27] is proposed to automatically adjust the control parameters during the entire evolution process. An adaptive hybrid DE algorithm (AH-DEa) is proposed in [28]. The algorithm employs a binomial crossover in early stages of evolution for exploration, while an exponential crossover is

employed for exploitation in later stages. Adaptive invasion-based migration model for dDE (AIM-dDE) [29] is a novel adaptive model endowed with three updating schemes for randomly setting DE control parameters. A self-adaptive DE algorithm based on Gaussian probability distribution, gamma distribution, and chaotic sequence (DEGC) [30] is proposed, where the control parameters are self-adaptive. A differential evolution algorithm with self-adaptive multi-combination strategies (SAS-DE) is proposed in [31]. SAS-DE combines the strengths of four mutations, two crossover operators, and two constraint handling techniques.

So far, differential mutation operator can adaptively adjust the search step according to the landscape of the objective function and convergence state for solving simple problems. However, the simple differential mutation is not good at guiding the population evolution for solving complex problems. Therefore, many parameter control mechanisms are proposed in the aforementioned references to improve the exploration ability and exploitation ability of the algorithm. Inspired by the formation and change of cloud in nature, we introduce a new algorithm, namely, Cloud Particles Differential Evolution (CPDE). The essence of CPDE is to provide a phase transformation mechanism for promoting the nucleus (the leader) to lead the swarm through the

FIGURE 7: Continued.

FIGURE 7: Convergence performance of the nine different algorithms for ten 30-dimensional functions.

evolution. An individual of the population corresponds to a cloud particle. The best solution found so far acts as a nucleus. We use the nucleus to generate new population by cloud generator. Then all the individuals will round the nucleus, thereby providing a more promising searching direction toward the global optimum. However, the misleading effect of a nucleus positioned at a local optimum point may cause the premature convergence. To resolve the issue, we introduce a new mutation strategy called cloud differential mutation. The cloud differential mutation can improve the population diversity so that the premature convergence can be alleviated.

The rest of the paper is organized as follows. Section 2 describes the basic procedure of differential evolution. Section 3 states the basic idea and principle of CPDE in detail. Simulation results are presented and are discussed in Section 4 for the comparison of CPDE with other eight evolutionary algorithms. Finally, the conclusions and ideas for future research are given in Section 5.

2. Classical Differential Evolution

DE is a simple but effective population-based stochastic search technique for solving global optimization problems. Mathematically, an OP (optimization problem) can be formulated as follows:

$$\begin{aligned} \min \quad & f(x_1, x_2, \dots, x_D) \\ \text{s.t.} \quad & x_j^L \leq x_j \leq x_j^U \quad j = 1, 2, \dots, D, \end{aligned} \quad (1)$$

where D is the number of decision variables and x_j^L and x_j^U denote the minimum and maximum bounds of x_j , respectively.

2.1. Initialization of the Population. DE begins with a randomly initiated population of NP D -dimensional real-valued parameter vectors: $\{x_i(0) \mid x_{ji}^L \leq x_{ji}(0) \leq x_{ji}^U, i =$

$1, 2, \dots, N, j = 1, 2, \dots, D\}$. The initial value of the j th component in the i th vector is produced by

$$x_{ji}(0) = x_{ji}^L + \text{rand}(0, 1) \cdot (x_{ji}^U - x_{ji}^L), \quad (2)$$

$$i = 1, 2, \dots, N, \quad j = 1, 2, \dots, D,$$

where N denotes the population size and $\text{rand}(0, 1)$ represents a uniformly distributed random variable within the range $[0, 1]$.

2.2. Mutation. After initialization, DE employs the mutation operation and arithmetic recombination to generate a mutant vector $\mathbf{V}_{i,G}$ with respect to each individual $\mathbf{X}_{i,G}$, so-called target vector, in the current population at the generation G . There are many mutation strategies available in the literature [21, 32, 33]; the classical one is “DE/rand/1”:

$$\mathbf{V}_{i,G} = \mathbf{X}_{r_1,G} + F \cdot (\mathbf{X}_{r_2,G} - \mathbf{X}_{r_3,G}). \quad (3)$$

The indices r_1^i , r_2^i , and r_3^i are mutually exclusive integers randomly generated within the range $[1, N]$, which are also different from the index i . The scaling factor F is a positive control parameter for scaling the difference vector.

2.3. Crossover. After mutation, crossover operation is applied to increase the potential diversity of the population. A trial vector $\mathbf{U}_{i,G} = (u_{i,G}^1, u_{i,G}^2, \dots, u_{i,G}^D)$ is produced through crossover operation. In the basic version, DE employs the binomial (uniform) crossover defined as follows:

$$u_{i,G}^j = \begin{cases} v_{i,G}^j & \text{if } (\text{rand}_j(0, 1) \leq \text{Cr or } j = j_{\text{rand}}), \\ x_{i,G}^j & \text{otherwise,} \end{cases} \quad (4)$$

$$j = 1, 2, \dots, D.$$

In (4), the crossover rate Cr is a user-specified constant and appears as a control parameter of DE. $\text{rand}_j(0, 1) \in [0, 1]$

FIGURE 8: Continued.

FIGURE 8: Convergence performance of the nine different algorithms for ten 50-dimensional functions.

is a uniformly distributed random number. $j_{\text{rand}} \in [1, D]$ is a randomly chosen index. The binomial crossover is performed on each of the D variables. If $\text{rand}_j(0, 1) \leq \text{Cr}$ or $j = j_{\text{rand}}$, the binomial crossover operator copies the j th component of the mutant vector $\mathbf{V}_{i,G}$ to the corresponding element in the trial vector $\mathbf{U}_{i,G}$. Otherwise, it is copied from the corresponding target vector $\mathbf{X}_{i,G}$. The aim of introducing the condition $j = j_{\text{rand}}$ is to ensure that there will be some differences between the trial vector $\mathbf{U}_{i,G}$ and the target vector $\mathbf{X}_{i,G}$ by at least one component.

2.4. Selection. DE adopts greedy algorithm to select the population for the next generation. The selection operation can be expressed as follows:

$$\mathbf{X}_{i,G+1} = \begin{cases} \mathbf{U}_{i,G}, & \text{if } f(\mathbf{U}_{i,G}) \leq f(\mathbf{X}_{i,G}), \\ \mathbf{X}_{i,G}, & \text{otherwise.} \end{cases} \quad (5)$$

DE includes four steps which are initialization, mutation, crossover, and selection.

3. Cloud Particles Differential Evolution Algorithm

The classic DE has two control parameters F and Cr that need to be tuned by the user. Better control parameter values are helpful to increase the population diversity and avoid trapping to a local minimum. However, these parameters are problem-dependent and thus a trial-and-error is required to find their appropriate values for each specific problem. Obviously, it may expend a huge amount of computational costs. Motivated by these observations, we develop a CPDE algorithm, in which the populations evolve toward the direction of optimum according to their previous experiences of generating promising solutions.

3.1. Cloud Formation. The idea of the proposed CPDE is inspired from cloud formation in nature. The state of the cloud particles' changing process [34] is shown in Figure 1.

When the water vapor is saturated, it will adsorb on the cloud condensation nuclei (CCN). Then, the cloud particles are produced. Cloud particles are condensed and sublimated continuously by absorbing water vapor around the cloud. In the process of the condensation, the cloud particles are getting closer and are colliding to be a bigger one, which is called large cloud particle. Then, a large number of supercooled water droplets are generated from the changes of large cloud particles. Ice crystals begin to grow up as supercooled water droplets evaporate. When ice crystals are getting bigger and bigger, they will be able to overcome the air resistance and buoyancy, and they will fall to the ground. Therefore, snow or hailstone is formed. The snow or hailstone in our planet will be irradiated by the sunshine, and it will become the water vapor again in the air.

3.2. Description of the Proposed Algorithm. There are various kinds of substance in nature. Substances commonly exist in three states: gaseous, liquid, and solid. For example, Figure 2 is a simple diagram to show how water transforms among the three states. Liquefaction, solidification, and sublimation are the three ways for water transforming. In order to describe the different forms of the substance, the phase can be used as a symbol for states of substances. A phase is defined as a homogenous portion in a material that has its own physical and chemical characteristics and properties [35]. A phase transition is the transformation of a thermodynamic system from one phase or state of matter to another one.

After a serious study of the cloud change in nature and three states of change of water, the optimization process of the cloud particles can be divided into three stages: cloud gaseous phase stage, cloud liquid phase stage, and cloud solid phase stage. Cloud gaseous phase corresponds to the initial state of the cloud particles. Cloud liquid phase corresponds to

FIGURE 9: Continued.

FIGURE 9: Convergence performance of the nine different algorithms for ten 100-dimensional functions.

the supercooled water droplets in nature. Cloud solid phase corresponds to the hailstones in nature.

The proposed algorithm which is similar to other metaheuristic algorithms begins with an initial population called the cloud particles. At the beginning, let the initial state be cloud gaseous phase which is composed of many cloud particles (shown in Figure 3). The best individual (best cloud particle) is chosen as a nucleus. Then, the cloud particles are going to have a condensational growth or collisions growth to form the larger cloud particles (shown in Figure 4). If the population evolves successfully, the cloud particles are going to change from cloud gaseous phase to liquid phase by liquefaction. This process is called phase transformation. When cloud particles are in liquid phase, cloud particles transform from global exploration to local exploitation gradually. In this process, the local exploitation ability is improved. The cloud particles carry out macrolocal exploitation in liquid phase. When the optimal solution area is found, it will change from the liquid phase to solid phase by phase transformation. When cloud particles are in solid phase (shown in Figure 5), the algorithm begins to accelerate convergence to find the optimal solution. The cloud particles perform microlocal exploitation in solid phase. Here, the hailstone represents the optimal solution.

The optimization process of CPDE model is shown in Figure 6.

In CPDE, the initialization, condensation operation, liquefaction operation, cloud differential mutation, and solidification operation are described as follows.

(1) *Initialization.* When the metaheuristic methods are used to solve the optimization problem, one population is formed at first. Then, the variables which are involved in the optimization problem can be represented as an array. In this algorithm, one single solution is called one “cloud particle.” The cloud particles have three parameters [36]. They are expectation (E_x), entropy (E_n), and hyperentropy (H_e). E_x

is the expectation values of the distribution for all cloud particles in the domain. E_n is the range of domain which can be accepted by linguistic values (qualitative concept). In other words, E_n is ambiguity. H_e is the dispersion degree of entropy (E_n). That is to say, H_e is the entropy of entropy. In addition, H_e can be also defined as the uncertainty measure for entropy. In D dimensional optimization problems, a cloud particle is an array of $1 \times D$. This array is defined as follows:

$$\text{Cloud particle} = [x_1, x_2, \dots, x_D]. \quad (6)$$

To start the optimization algorithm, a matrix of cloud particles whose size is $N \times D$ is generated by cloud generator (i.e., population of cloud particles). Cloud particles generation is described as follows. Firstly, a set of data (S) is achieved by normal distribution, among which E_n is expectation and H_e is variance. Secondly, normal distribution is used again to generate the cloud particles, among which E_x is expectation and S is variance.

The cloud particle X generated by cloud generator is given as (rows and column are the number of population and the number of design variables, resp.)

$$\text{Population of cloud particles} = \begin{bmatrix} \text{Cloudparticle}_1 \\ \text{Cloudparticle}_2 \\ \vdots \\ \text{Cloudparticle}_N \end{bmatrix} \quad (7)$$

$$= \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & \cdots & x_D^1 \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_D^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^N & x_2^N & x_3^N & \cdots & x_D^N \end{bmatrix}.$$

Each of the decision variable values ($x_1, x_2, \dots, x_D$) can represent a real value for continuous or discrete problems.

FIGURE 10: Continued.

FIGURE 10: Convergence performance of the nine different algorithms for ten 500-dimensional functions.

The fitness of a cloud particle corresponds to the objective function value. In (7), N and D stand for the number of cloud particles and the number of decision variables, respectively. The cloud particles whose number is N are generated by the cloud generator. The cloud particle which has the optimal value is selected as the nucleus. The total population consists of N_p subpopulations. Each subpopulation of cloud particles has a nucleus. Therefore, there are N_p nuclei.

(2) *Condensation Operation.* The population realizes the global exploration in cloud gaseous phase by condensation operation. Here, we use the differential evolution theory. The algorithm generates a new cloud particle $\vec{x}_i^{\text{new}}$ for each $\vec{x}_i$ in the population by adding the weighted difference between two randomly selected cloud particles to a new one:

$$\vec{x}_i^{\text{new}} = \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}), \quad (8)$$

where the indexes r_1, r_2 , and r_3 are mutually exclusive integers randomly chosen from the range $(r_1, r_2, r_3 \in \{1, 2, \dots, N\})$.

Then, we obtain a trial vector $\mathbf{u}_i$ by mixing the components of $\vec{x}_i^{\text{new}}$ and $\mathbf{x}_i$:

$$u_i^j = \begin{cases} \vec{x}_i^{\text{new}}, & \text{if } (\text{rand}_j(0, 1) \leq \text{Cr} \text{ or } j = j_{\text{rand}}), \\ \mathbf{x}_i^j, & \text{otherwise,} \end{cases} \quad (9)$$

$$j = 1, 2, \dots, D,$$

where D is the dimension of decision variables and $j_{\text{rand}} = \lfloor \text{rand}(0, 1) \times D \rfloor + 1$.

The suggested choices by Storn in [3, 37] are $F \in [0.5, 1]$ and $\text{Cr} \in [0.8, 1]$. Here, we have used $F = 0.5$ and $\text{Cr} = 0.9$. Our decision for using those values is based on proposed values from the literature [3, 37, 38].

Finally, $\mathbf{x}_i$ is replaced by $\mathbf{u}_i$ if $\mathbf{u}_i$ is better according to (5).

(3) Liquefaction Operation is

$$\text{nucleus}_i = \mathbf{x}_i^*, \quad i = 1, 2, \dots, N_p, \quad (10)$$

$$E_{n_i} = \begin{cases} \frac{|\text{nucleus}_i|}{lf}, & \text{if } |\text{nucleus}_i| > \varepsilon, \\ |\text{nucleus}_i|, & \text{otherwise,} \end{cases} \quad (11)$$

$$H_{e_i} = \frac{E_{n_i}}{\sqrt{N^2}}. \quad (12)$$

$\mathbf{x}_i^*$ represents the best solution of i th subpopulation. E_{n_i} is the entropy of the i th subpopulation. H_{e_i} is hyperentropy of the i th subpopulation. lf is the factor of liquefaction. ε is the threshold. N_p represents the number of the subpopulations. N stands for the population size.

We use cloud generator to produce the cloud particles with nucleus_i , E_{n_i} and H_{e_i} .

In liquid phase, the individual is updated by cloud differential mutation.

(4) Cloud Differential Mutation is

$$\text{nucleus} = \mathbf{x}_{r_1}, \quad (13)$$

$$E_n = \frac{\text{nucleus}}{(2 * N)}, \quad (14)$$

$$H_e = \frac{E_n}{N^2}, \quad (15)$$

$$\vec{x}_i^{\text{new}} = \begin{cases} \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) - \left| \sqrt[3]{0.5 - r} \times (c_2 - c_1) \right|, & \text{if } r < 0.5, \\ \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}), & \text{otherwise,} \end{cases} \quad (16)$$

FIGURE II: Continued.

FIGURE 11: Convergence performance of the nine different algorithms for ten 1000-dimensional functions.

where $r = \text{rand}(0, 1)$ and $\text{rand}(0, 1)$ is a uniformly distributed random number. c_1 and c_2 are two cloud particles generated by cloud generator with nucleus, E_n and H_e .

(5) *Solidification Operation.* The algorithm realizes convergence operation in cloud solid phase by solidification operation. sf is the solidification factor:

$$E_{n_i} = \begin{cases} \frac{|\text{nucleus}_i|}{sf}, & \text{if } |\text{nucleus}_i| > \varepsilon, \\ |\text{nucleus}_i|, & \text{otherwise,} \end{cases} \quad (17)$$

$$H_{e_i} = \frac{E_{n_i}}{\sqrt{N^2}}, \quad (18)$$

where $i = 1, 2, \dots, N_p$. N_p represents the number of the subpopulations.

The pseudo-code of CPDE is illustrated in Algorithms 1–4. N stands for the population size. *Childnum* represents the number of individuals of every subpopulation. *Runmax* represents the number of total runs. *NFES* represents the number of function evaluations. *MaxFES* represents the maximum number of function evaluations. The search space is $[-r, r]^D$.

4. Experiments and Discussions

In this section, the performance of the proposed CPDE is tested with a comprehensive set of benchmark functions which include 10 different global optimization problems and CEC2013 benchmark problems. The details of 10 benchmark functions are shown in Table 1. The searching range and theory optima for all functions are also shown in Table 1. For 10 different global optimization problems, f_1 – f_5 are unimodal functions and f_6 – f_{10} are multimodal functions. The CEC2013 benchmark functions [39] broadly fall into four different groups (based on their characteristics): namely, f_1 – f_5

are unimodal functions, f_6 – f_{18} are multimodal functions, f_{19} – f_{20} are expanded multimodal functions, and f_{21} – f_{28} are hybrid composition functions. To reduce the occurrence of statistical errors, each function is independently simulated in 30 runs for comparison. The value of each function is defined as the fitness value in the algorithm. For a thorough comparison, we not only compare the mean best solution, standard deviation, maximum, and minimum of 9 algorithms for each function, but also carry out the t -test [40]. The T values are presented on every function of this two-tailed test with a significant level of 0.05 between CPDE and another algorithm. “General merit over contender” displays the difference between the number of better results and the number of worse results, which is used to give an overall comparison between the two algorithms. All the algorithms are implemented on the same machine with a T2450 2.4 GHz CPU, 1.5 GB memory, and windows XP operating system with MATLAB R2009b. For DE, CMAES, JADE, jDE, PSOf, PSOWFIPS, UPSO, and ABC algorithms, the training parameters are the same as those used in the corresponding references, except for the maximal evolutionary generation and population size. The parameter values of CPDE are set to be $E_n = (2/3)r$, (the search space is $[-r, r]^D$), $H_e = E_n/1000$, and $E_x = (1/2)r$ used in [41]. The population size was set to 100 for all the algorithms. *MaxFES* was set to 60000 for 30-dimensional and 50-dimensional problems. *MaxFES* was set to 100000 for 100-dimensional and CEC2013 problems. *MaxFES* was set to 300000 for 500-dimensional and 1000-dimensional problems.

4.1. Comparison of Different Algorithms for 30-Dimensional Problems. Each column of Table 2 shows the mean best solution (F_{mean}), standard deviation (SD), maximum (Max), and minimum (Min) of error value between the best fitness values found in each run and the true optimal value. Table 2 indicates that CPDE outperforms the other algorithms in terms of the mean best solution, standard deviation, and

FIGURE 12: Continued.

FIGURE 12: Continued.

FIGURE 12: Continued.

FIGURE 12: Convergence performance of the nine different algorithms for CEC2013 problems.

```

(1) if State == 0 then
(2)   for  $i = 1$  to  $N$ 
(3)     Generate  $\tilde{x}_i^{\text{new}}$  according to (8)
(4)     Vector  $\mathbf{V}_{i,G}$  is generated according to (4) and (9)
(5)     Select the population ( $\mathbf{X}_{i,G+1}$ ) for the next generation
(6)   end for
(7)   if  $NFES == Runmax * MaxFES/N$  then      %Phase Transformation from gaseous to liquid
(8)     for  $i = 1$  to  $N_p$                       % Liquefaction Operation
(9)       Calculate the nucleus $_i$  for each subpopulation according to (10)
(10)      Calculate  $E_{n_i}$  for each subpopulation according to (11)
(11)      Calculate  $H_{e_i}$  for each subpopulation according to (12)
(12)    end for
(13)    Use cloud generator to produce cloud particles
(14)    State = 1;
(15)  end if
(16) end if

```

ALGORITHM 1: Evolutionary algorithm of the cloud gaseous phase.

```

(1) if State == 1 then
(2)   for  $i = 1$  to  $N$                                      % Cloud Differential Mutation
(3)     Calculate the nucleus according to (13)
(4)     Calculate  $E_n$  according to (14)
(5)     Calculate  $H_e$  according to (15)
(6)     Generate  $c_1$  and  $c_2$  with cloud generator
(7)     Generate  $\tilde{x}_i^{\text{new}}$  according to (16)
(8)     Vector  $V_{i,G}$  is generated according to (4) and (9)
(9)     Select the population ( $X_{i,G+1}$ ) for the next generation
(10)   end for
(11) if  $NFES == Runmax * MaxFES * 0.025$  then %Phase Transformation from liquid to solid
(12)    $N_p = N_p/2$ 
(13)   Archive =  $X_G$ 
(14)    $\lambda = f_{\min}(X_G)$ 
(15)   State = 2
(16) end if
(17) end if

```

ALGORITHM 2: Evolutionary algorithm of the cloud liquid phase.

```

(1) if State == 2 then
(2)   if  $\lambda \leq f_{\min}(X_G)$  then                               % Solidification Operation
(3)     for  $i = 1$  to  $N_p$ 
(4)       Calculate the nucleus $_i$  for each subpopulation according to (10)
(5)       Calculate  $E_{n_i}$  for each subpopulation according to (17)
(6)       Calculate  $H_{e_i}$  for each subpopulation according to (18)
(7)     end for
(8)     Use cloud generator to produce cloud particles ( $X_G$ )
(9)     Evaluate Population
(10)    Archive =  $X_G$ 
(11)     $\lambda = f_{\min}(X_G)$ 
(12)  else
(13)     $X_G = \text{Archive}$ 
(14)    for  $i = 1$  to  $N$                                      % Cloud Differential Mutation
(15)      Calculate the nucleus according to (13)
(16)      Calculate  $E_n$  according to (14)
(17)      Calculate  $H_e$  according to (15)
(18)      Generate  $c_1$  and  $c_2$  with cloud generator
(19)      Generate  $\tilde{x}_i^{\text{new}}$  according to (16)
(20)      Vector  $V_{i,G}$  is generated according to (4) and (9)
(21)      Select the population ( $X_{i,G+1}$ ) for the next generation
(22)    end for
(23)  end if
(24) end if

```

ALGORITHM 3: Evolutionary algorithm of the cloud solid state.

maximum and minimum for the functions f_1, f_2, f_6, f_7, f_8 , and f_9 . For function f_3 , DE, CMAES, JADE, jDE, UPSO, ABC, and CPDE have the same solutions. For function f_4 , CMAES outperforms the other algorithms in terms of the mean best solution, standard deviation, maximum, and minimum. For function f_5 , the mean best solution, standard deviation, and maximum of CMAES are smaller than those of other algorithms. CPDE has the best performance in terms of minimum for function f_5 . For function f_{10} , the mean best solution, maximum, and minimum of CMAES are smaller

than those of other algorithms. The standard deviation of jDE is the smallest.

Different algorithms have different convergence characteristics and processes for different functions. The convergence graphs of different benchmark functions in terms of the median fitness values achieved by each of 9 algorithms for 30 runs are shown in Figure 7. Figure 7 indicates that the convergence speed of CPDE is the quickest compared to that of the other algorithms for functions f_1, f_2, f_6, f_7, f_8 , and f_9 . For function f_3 , all the algorithms can quickly converge

```

(1) Initialize  $D$  (number of dimensions),  $N$  (number of population);  $N_p = 10$ ;  $Childnum = N/N_p$ ;
(2)  $\varepsilon = 5$ ;  $lf = 3$ ;  $sf = 40$ ;  $E_n = 2/3r$ ;  $H_e = E_n/\sqrt{N^3}$ ;
(3) for  $i = 1: N_p$ 
(4)   for  $j = 1: D$ 
(5)     if  $j$  is even
(6)        $E_{x_{ij}} = r/2$ 
(7)     else
(8)        $E_{x_{ij}} = -r/2$ 
(9)     end if
(10)  end for
(11) end for
(12) Use cloud generator to produce cloud particles
(13) State = 0
(14) while requirements are not satisfied do
(15)   Evolutionary algorithm of the cloud gaseous state
(16)   Evolutionary algorithm of the cloud liquid state
(17)   Evolutionary algorithm of the cloud solid state
(18) end while
      State = 0: Cloud Gaseous Phase, State = 1: Cloud Liquid Phase, State = 2: Cloud Solid Phase

```

ALGORITHM 4: Cloud particles evolution algorithm: CPDE.

except PSOf and PSOWFIPS. For functions f_4 and f_5 , the convergence speed of CMAES is the quickest. For function f_{10} , none of the algorithms converge to the global optimum.

Table 2 and Figure 7 conclude that the CPDE has a good performance of the solution accuracy for global optimization problems with 30 dimensions.

4.2. Comparison of Different Algorithms for 50-Dimensional Problems. Table 3 shows the results in terms of the mean best solution, standard deviation, and maximum and minimum of the solutions obtained in the 30 independent runs by each algorithm on 10 test functions. The best results among the algorithms are shown in bold.

From Table 3 it can be observed that CPDE outperforms the other algorithms in terms of the mean best solution, standard deviation, maximum, and minimum for the functions f_1 , f_2 , f_5 , f_6 , f_7 , and f_8 . For function f_3 , jDE, UPSO, and CPDE have the same solutions. CMAES performs better for function f_4 . CMAES and CPDE perform well for function f_9 . CMAES performs better in terms of the mean best solution, maximum, and minimum for function f_{10} . Table 3 also gives the information about the t -value on 10 test functions. Figure 8 presents the convergence graphs of different benchmark functions in terms of the median fitness values achieved by each of 9 algorithms for 50-dimensional problems.

Table 3 and Figure 8 conclude that the CPDE generally offered better performance than those of other algorithms in the paper though it performed weaker for some functions with 50 dimensions.

4.3. Comparison of Different Algorithms for 100-Dimensional Problems. To test the performance of the CPDE for high-dimensional problems, we simulated the 10 benchmark functions in Table 4 in 100 dimensions. The information about the

t -value on 10 test functions is presented in Table 4. The best results among the algorithms are shown in bold.

From Table 4 it can be observed that CPDE outperforms the other algorithms in terms of the mean best solution, standard deviation, maximum, and minimum for the functions f_1 , f_2 , f_3 , f_6 , f_7 , and f_8 . CMAES performs better for function f_4 . jDE performs better in terms of the mean best solution, maximum, and minimum for function f_5 . CPDE performs well in terms of the mean best solution and minimum for function f_9 . CMAES performs better in terms of the mean best solution, maximum, and minimum for function f_{10} . Figure 9 presents the convergence graphs of different benchmark functions in terms of the median fitness values achieved by each of 9 algorithms for 100-dimensional problems.

Table 4 and Figure 9 conclude that the CPDE performs better than those of other algorithms for most 100-dimensional functions.

4.4. Comparison of Different Algorithms for 500-Dimensional Problems. Table 5 lists the results in terms of the mean best solution, standard deviation, maximum, and minimum of the solutions and the t -value for 500-dimensional functions. The best results among the algorithms are shown in bold.

From Table 5 it can be observed that CPDE outperforms the other algorithms for all functions except f_4 , f_5 , f_6 , and f_{10} . CPDE performs better in terms of the mean best solution and minimum for function f_4 . jDE performs better in terms of the mean best solution, maximum, and minimum for function f_5 . ABC performs well for function f_6 . CMAES performs better in terms of the mean best solution, standard deviation, and maximum for function f_{10} . The minimum of CPDE algorithm performs well for function f_{10} . Figure 10 presents the convergence graphs of different benchmark functions in terms of the median fitness values achieved by each of 9 algorithms for 500-dimensional problems.

TABLE 1: Details of numerical benchmarks used in the experiments.

Function	Formula	Range	Optima
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$	0
Sum Square	$f_2(x) = \sum_{i=1}^D ix_i^2$	$[-100, 100]^D$	0
Step	$f_3(x) = \sum_{i=1}^D ([x_i + 0.5])^2$	$[-100, 100]^D$	0
Schwefel 1.2	$f_4(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^D$	0
Zakharov	$f_5(x) = \sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5ix_i \right)^2 + \left(\sum_{i=1}^D 0.5ix_i \right)^4$	$[-100, 100]^D$	0
Ackley	$f_6(x) = 20 - 20 \exp \left(-\frac{1}{5} \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + e$	$[-32.768, 32.768]^D$	0
Rastrigin	$f_7(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^D$	0
Weierstrass	$f_8(x) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k \times 0.5)]$, $a = 0.5, b = 3, k_{\max} = 20$	$[-0.5, 0.5]^D$	0
Griewank	$f_9(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]^D$	0
Rosenbrock	$f_{10}(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	$[-2.048, 2.048]$	0

Table 5 and Figure 10 conclude that the CPDE is a challenging method for high-dimensional global optimization.

4.5. Comparison of Different Algorithms for 1000-Dimensional Problems. The mean best solution, standard deviation, maximum, and minimum of the solutions and the t -value for 1000-dimensional functions obtained from 30 independent runs are listed in Table 6. The best results among the algorithms are shown in bold.

From Table 6 it can be observed that CPDE outperforms the other algorithms for the functions f_1, f_2, f_3, f_7, f_8 , and f_9 . jDE performs better for functions f_4 and f_5 . ABC performs well in terms of the mean best solution, maximum, and minimum for function f_6 . CMAES performs better for function f_{10} . Figure 11 presents the convergence graphs of different benchmark functions in terms of the median fitness values achieved by each of 9 algorithms for 1000-dimensional problems.

4.6. Comparison of Different Algorithms for CEC2013 Problems. To further verify the performance of CPDE, a set of recently proposed CEC2013 rotated benchmark problems are used. In this section, t -test has also been carried out to show the differences between CPDE and the other algorithm. The function parameters are the same as those in [35], except that the stop condition of each algorithm is the maximal number

of $NFES$. The maximal $NFES$ for all algorithms are 100,000. The algorithm parameters are the same as the previous experiments. Table 7 gives the information about the mean best solution, standard deviation, maximum, and minimum of the solutions and t -value of 30 runs of 9 algorithms on 28 test functions with $D = 10$. The best results among the algorithms are shown in bold.

Figure 12 presents the convergence graphs of different benchmark functions in terms of the median fitness values achieved by each of 9 algorithms for CEC2013 problems.

First, f_1 to f_5 are the five unimodal benchmark functions. From Table 7, one can observe that CPDE performs well for functions f_1 to f_5 . JADE performs well for functions f_1 to f_5 except function f_3 . DE, jDE, and UPSO perform well for functions f_1 and f_5 . ABC performs well for function f_1 . Based on these observations, it may be inferred that CPDE provides significantly better solutions for unimodal functions.

Next, f_6 to f_{18} are multimodal benchmark functions and f_{19} to f_{20} are expanded multimodal benchmark functions. CPDE performs better for functions f_6, f_8, f_{12}, f_{13} , and f_{20} . DE performs better for functions f_7 and f_9 . JADE performs better for functions f_{11}, f_{15} , and f_{17} . jDE performs better for functions f_{11} and f_{14} . ABC performs better for function f_{19} .

Finally, f_{21} to f_{28} are hybrid composition functions. CPDE performs better for functions f_{23} and f_{26} . JADE

TABLE 2: The mean best solutions, standard deviation, maximum, and minimum of the 30 trials obtained by various methods for 30-dimensional functions.

F	DE	CMAES	JADE	jDE	PSOcf	PSOWFIPS	UPSO	ABC	CPDE
f_1	F_{mean}	3.56e-04(+)	1.19e-20(+)	6.40e-09(+)	1.92e-12(+)	1.25e+02(+)	4.17e-07(+)	8.46e-16(+)	4.02e-53
	SD	1.67e-04	2.17e-20	4.62e-09	2.90e-12	2.30e+01	2.18e-07	1.54e-16	1.35e-52
	Max	7.53e-04	8.95e-20	2.26e-08	1.42e-11	1.67e+02	1.00e-06	1.16e-15	6.09e-52
	Min	1.23e-04	3.41e-22	1.22e-09	7.28e-14	7.75e+01	1.27e-07	4.77e-16	5.20e-58
f_2	F_{mean}	3.99e-03(+)	2.87e+05(+)	9.68e-08(+)	1.20e+04(+)	1.47e+03(+)	5.47e-06(+)	9.24e-16(+)	6.89e-51
	SD	2.27e-03	1.68e+05	7.53e-08	2.05e+04	2.09e+02	2.51e-06	2.24e-16	2.06e-50
	Max	1.02e-02	8.69e+05	3.32e-07	8.00e+04	1.91e+03	1.22e-05	1.42e-15	8.51e-50
	Min	1.64e-03	1.22e+05	1.39e-08	2.08e-12	1.10e+03	2.13e-06	4.96e-16	5.64e-58
f_3	F_{mean}	0.00e+00(≈)	0.00e+00(≈)	0.00e+00(≈)	1.10e+00(+)	1.30e+02(+)	0.00e+00(≈)	0.00e+00(≈)	0.00e+00
	SD	0.00e+00	0.00e+00	0.00e+00	9.22e-01	2.43e+01	0.00e+00	0.00e+00	0.00e+00
	Max	0.00e+00	0.00e+00	0.00e+00	3.00e+00	1.77e+02	0.00e+00	0.00e+00	0.00e+00
	Min	0.00e+00	0.00e+00	0.00e+00	0.00e+00	6.90e+01	0.00e+00	0.00e+00	0.00e+00
f_4	F_{mean}	4.21e+02(+)	3.43e+04(+)	9.50e+01(-)	4.24e+03(+)	9.68e+03(+)	1.78e+03(+)	6.74e+03(+)	1.75e+02
	SD	1.68e+02	1.00e+04	8.94e+01	4.97e+03	1.44e+03	5.94e+02	1.61e+03	5.41e+01
	Max	9.04e+02	5.27e+04	5.11e+02	1.52e+04	1.20e+04	3.33e+03	1.00e+04	3.41e+02
	Min	1.44e+02	1.48e+04	3.35e+01	3.32e+01	6.82e+03	9.82e+02	1.88e+03	9.75e+01
f_5	F_{mean}	6.18e+01(+)	2.34e+02(+)	4.37e+00(+)	1.11e+01(+)	2.50e+02(+)	1.32e+02(+)	5.60e+02(+)	4.64e-22
	SD	1.46e+01	1.20e+02	1.03e+01	1.88e+01	3.69e+01	2.50e+01	7.93e+01	1.40e-21
	Max	9.24e+01	7.87e+02	5.77e+01	1.06e+02	3.22e+02	1.91e+02	7.21e+02	5.55e-21
	Min	3.65e+01	1.29e+02	4.31e-01	1.32e+00	1.45e+02	9.76e+01	4.03e+02	6.79e-29
f_6	F_{mean}	5.94e-03(+)	1.94e+01(+)	1.79e-05(+)	2.24e-01(+)	4.16e+00(+)	4.26e-04(+)	6.80e-14(+)	0.00e+00
	SD	1.44e-03	1.43e-01	4.34e-06	5.35e-01	2.88e-01	1.90e-04	1.41e-14	0.00e+00
	Max	9.85e-03	1.97e+01	2.59e-05	2.11e+00	4.67e+00	1.17e-03	9.94e-14	0.00e+00
	Min	3.58e-03	1.91e+01	1.08e-05	1.09e-07	3.56e+00	1.99e-04	4.97e-14	0.00e+00
f_7	F_{mean}	2.00e+02(+)	4.59e+00(+)	1.01e+01(+)	7.38e+01(+)	1.57e+02(+)	7.62e+01(+)	3.82e-07(≈)	0.00e+00
	SD	9.62e+00	1.21e+00	2.29e+00	2.58e+01	1.02e+01	1.35e+01	1.48e-06	0.00e+00
	Max	2.13e+02	7.01e+00	1.41e+01	1.47e+02	1.76e+02	1.08e+02	6.86e-06	0.00e+00
	Min	1.76e+02	2.39e+00	4.50e+00	3.18e+01	1.28e+02	3.74e+01	0.00e+00	0.00e+00
f_8	F_{mean}	5.92e-01(+)	2.81e+01(+)	2.51e+00(+)	3.58e+00(+)	6.95e+00(+)	3.66e-02(+)	8.76e-15(+)	0.00e+00
	SD	1.26e-01	3.87e+00	1.62e+00	2.44e+00	4.20e-01	9.03e-03	1.24e-14	0.00e+00
	Max	9.63e-01	3.52e+01	8.20e+00	9.83e+00	7.49e+00	6.86e-02	4.97e-14	0.00e+00
	Min	3.39e-01	2.02e+01	8.08e-01	7.53e-02	5.73e+00	2.19e-02	0.00e+00	0.00e+00

TABLE 2: Continued.

F	DE	CMAES	JADE	jDE	PSOcf	PSOWFIPS	UPSO	ABC	CPDE
f_9									
F_{mean}	$1.14e-03(+)$	$2.13e-03(+)$	$4.42e-01(+)$	$5.20e-08(+)$	$9.59e-03(+)$	$2.09e+00(+)$	$3.67e-05(\approx)$	$2.81e-04(\approx)$	0.00e+00
SD	$7.51e-04$	$3.64e-03$	$1.32e-01$	$1.37e-07$	$1.34e-02$	$1.77e-01$	$9.85e-05$	$1.52e-03$	0.00e+00
Max	$4.09e-03$	$9.85e-03$	$8.00e-01$	$7.49e-07$	$5.16e-02$	$2.59e+00$	$4.75e-04$	$8.33e-03$	0.00e+00
Min	$2.07e-04$	0.00e+00	$1.86e-01$	$4.21e-09$	$2.48e-13$	$1.75e+00$	$4.92e-07$	$4.44e-16$	0.00e+00
f_{10}									
F_{mean}	$2.31e+01(\approx)$	6.62e-01(-)	$9.15e+00(-)$	$2.42e+01(+)$	$3.62e+01(\approx)$	$4.15e+01(+)$	$2.39e+01(\approx)$	$1.41e+01(-)$	$2.34e+01$
SD	$6.17e-01$	$1.40e+00$	$1.10e+00$	5.90e-01	$3.78e+01$	$2.92e+00$	$2.47e+00$	$8.76e+00$	$6.79e-01$
Max	$2.43e+01$	3.98e+00	$1.21e+01$	$2.59e+01$	$2.07e+02$	$4.67e+01$	$2.57e+01$	$2.67e+01$	$2.64e+01$
Min	$2.16e+01$	6.20e-18	$6.77e+00$	$2.29e+01$	$1.49e+01$	$3.58e+01$	$1.55e+01$	$2.35e-01$	$2.42e+01$
General merit over contender	8	4	7	7	9	10	7	5	

" $+$ ", " $\approx$ ", " $-$ ", and " $\approx$ " denote that the performance of CPDE ($D = 30$) is better than, worse than, and similar to that of other algorithms.

TABLE 3: The mean solutions, standard deviation, maximum, and minimum of the 30 trials obtained by various methods for 50-dimensional functions.

F	DE	CMAES	JADE	jDE	PSOcf	PSOWFIPS	UPSO	ABC	CPDE
f_1	F_{mean}	5.01e-01(+)	1.07e-28(+)	7.49e-17(+)	6.95e-05(+)	6.66e+02(+)	2.25e+03(+)	5.73e-03(+)	2.86e-15(≈)
	SD	2.02e-01	2.34e-29	1.06e-16	4.20e-05	1.53e+03	3.16e+02	1.13e-03	9.42e-15
	Max	9.26e-01	1.53e-28	5.02e-16	2.06e-04	1.00e+04	2.99e+03	9.17e-03	1.46e-14
	Min	2.15e-01	5.47e-29	5.81e-18	2.18e-05	4.49e-06	1.46e+03	3.80e-03	1.53e-15
f_2	F_{mean}	8.63e+00(+)	3.21e-27(+)	9.94e+05(+)	1.36e-03(+)	9.26e+04(+)	4.47e+04(+)	1.38e-01(+)	3.32e-15(≈)
	SD	3.42e+00	5.54e-28	4.49e+05	7.00e-04	1.15e+05	5.79e+03	3.24e-02	9.79e-15
	Max	1.66e+01	4.31e-27	2.29e+06	2.96e-03	5.10e+05	5.96e+04	2.09e-01	1.36e-14
	Min	2.61e+00	1.70e-27	3.76e+05	3.58e-04	1.71e-04	3.53e+04	8.13e-02	1.13e-15
f_3	F_{mean}	3.33e-02(≈)	4.00e-01(+)	3.33e-02(≈)	0.00e+00(≈)	1.34e+03(+)	2.26e+03(+)	0.00e+00(≈)	1.00e+00(+)
	SD	1.82e-01	6.74e-01	1.82e-01	0.00e+00	3.45e+03	3.05e+02	6.94e-01	6.94e-01
	Max	1.00e+00	2.00e+00	1.00e+00	0.00e+00	1.00e+04	2.94e+03	2.00e+00	2.00e+00
	Min	0.00e+00	0.00e+00	0.00e+00	0.00e+00	2.00e+00	1.69e+03	0.00e+00	0.00e+00
f_4	F_{mean}	1.64e+04(+)	2.52e-09(-)	1.07e+05(+)	2.17e+03(+)	2.75e+04(+)	4.38e+04(+)	1.81e+04(+)	3.06e+04(+)
	SD	4.67e+03	1.98e-09	2.23e+04	1.11e+03	8.09e+03	6.30e+03	2.96e+03	5.35e+03
	Max	2.54e+04	7.45e-09	1.53e+05	6.44e+03	4.52e+04	5.20e+04	2.30e+04	4.21e+04
	Min	9.50e+03	3.29e-10	4.01e+04	9.98e+02	1.37e+04	2.42e+04	1.18e+04	1.88e+04
f_5	F_{mean}	6.19e+02(+)	2.64e+01(≈)	6.32e+02(+)	4.27e+01(+)	3.09e+02(+)	8.27e+02(+)	7.09e+02(+)	1.15e+03(+)
	SD	1.08e+02	7.89e+01	1.68e+02	3.95e+01	6.54e+01	1.07e+02	8.95e+01	8.16e+01
	Max	7.82e+02	4.14e+02	1.27e+03	2.46e+02	5.12e+02	9.97e+02	8.54e+02	1.30e+03
	Min	3.73e+02	4.69e-02	4.17e+02	1.47e+01	2.06e+02	5.40e+02	4.59e+02	1.02e+03
f_6	F_{mean}	2.22e-01(+)	1.94e+01(+)	1.75e-09(+)	1.50e-03(+)	3.51e+00(+)	8.82e+00(+)	2.14e-01(+)	1.22e-10(+)
	SD	5.67e-02	1.06e-01	1.08e-09	4.60e-04	4.34e+00	4.00e-01	4.80e-01	1.24e-10
	Max	3.57e-01	1.96e+01	5.17e-09	3.21e-03	1.31e+01	9.50e+00	2.49e+00	5.09e-10
	Min	1.19e-01	1.90e+01	5.38e-010	9.48e-04	3.93e-04	8.18e+00	2.39e-02	1.71e-11
f_7	F_{mean}	3.87e+02(+)	3.86e+02(+)	5.04e+01(+)	6.40e+01(+)	2.03e+02(+)	3.64e+02(+)	1.80e+02(+)	1.71e-01(+)
	SD	2.05e+01	5.43e+01	4.36e+00	7.03e+00	5.01e+01	1.91e+01	2.48e+01	3.75e-01
	Max	4.15e+02	4.84e+02	5.79e+01	7.79e+01	3.35e+02	3.92e+02	2.50e+02	9.95e-01
	Min	3.47e+02	2.73e+02	4.25e+01	4.87e+01	1.19e+02	2.99e+02	1.30e+02	1.31e-13
f_8	F_{mean}	4.18e+00(+)	5.80e+00(+)	5.59e+01(+)	1.06e+01(+)	1.36e+01(+)	2.50e+01(+)	1.03e+00(+)	5.13e-10(+)
	SD	7.79e-01	2.53e+00	7.28e+00	3.45e+00	4.12e+00	1.26e+00	2.86e-01	1.07e-09
	Max	6.52e+00	1.04e+01	7.78e+01	2.22e+01	2.06e+01	2.78e+01	1.95e+00	4.66e-09
	Min	3.04e+00	1.75e+00	4.10e+01	4.26e+00	5.74e+00	2.29e+01	6.93e-01	1.29e-12

TABLE 3: Continued.

F	DE	CMAES	JADE	jDE	PSOcf	PSOWFIPS	UPSO	ABC	CPDE
f_9									
F_{mean}	4.04e-01(+)	0.00e+00($\approx$)	4.52e-01(+)	9.74e-05(+)	1.73e-05(+)	2.17e+01(+)	1.24e-02(+)	3.41e-10($\approx$)	0.00e+00
SD	1.30e-01	0.00e+00	1.89e-01	6.47e-05	1.29e-05	2.14e+00	7.79e-03	9.93e-10	0.00e+00
Max	7.04e-01	0.00e+00	8.69e-01	2.72e-04	4.48e-05	2.63e+01	4.25e-02	4.59e-09	0.00e+00
Min	1.81e-01	0.00e+00	1.26e-01	2.28e-05	2.43e-06	1.70e+01	4.90e-03	5.55e-16	0.00e+00
f_{10}									
F_{mean}	4.70e+01(+)	2.41e+01(-)	3.49e+01(-)	4.53e+01(+)	6.13e+01(+)	2.52e+02(+)	4.60e+01(+)	4.60e+01(+)	3.85e+01
SD	9.42e-01	1.64e+00	1.80e+00	4.04e-01	3.00e+01	2.61e+01	1.99e+00	1.88e+01	5.58e-01
Max	4.93e+01	2.87e+01	4.01e+01	4.62e+01	1.65e+02	3.19e+02	4.91e+01	9.41e+01	3.99e+01
Min	4.48e+01	2.15e+01	3.32e+01	4.46e+01	4.11e+01	2.01e+02	3.77e+01	2.49e+01	3.77e+01
General merit over contender	9	4	7	9	10	10	9	7	

" $+$ ", " $\approx$ ", " $-$ ", and " $\approx$ " denote that the performance of CPDE ($D = 50$) is better than, worse than, and similar to that of other algorithms.

TABLE 4: The mean solutions, standard deviation, maximum, and minimum of the 30 trials obtained by various methods for 100-dimensional functions.

F	DE	CMAES	JADE	jDE	PSOcf	PSOWFIPS	UFSO	ABC	CPDE
f_1	F_{mean}	2.06e + 00(+)	2.47e - 28(+)	2.37e - 16(+)	2.35e - 05(+)	1.30e + 04(+)	1.29e + 04(+)	2.45e - 14(+)	7.04e - 112
	SD	1.54e + 00	3.69e - 29	1.89e - 16	1.01e - 05	9.15e + 03	8.76e + 02	4.60e - 14	3.58e - 111
	Max	9.50e + 00	3.37e - 28	7.71e - 16	4.49e - 05	3.00e + 04	1.47e + 04	2.54e - 13	1.96e - 110
	Min	7.06e - 01	1.84e - 28	3.26e - 17	5.26e - 06	9.80e - 03	1.13e + 04	4.31e - 15	4.48e - 126
f_2	F_{mean}	7.69e + 01(+)	1.36e - 26(+)	6.56e + 06(+)	9.15e - 04(+)	8.14e + 05(+)	5.26e + 05(+)	1.20e - 13(+)	1.71e - 109
	SD	4.09e + 01	2.14e - 27	2.34e + 06	4.00e - 04	5.23e + 05	4.35e + 04	1.37e - 13	9.37e - 109
	Max	2.05e + 02	1.85e - 26	1.17e + 07	1.89e - 03	1.83e + 06	6.08e + 05	5.30e - 13	5.13e - 108
	Min	3.35e + 01	9.07e - 27	1.32e + 06	2.53e - 04	9.00e + 04	4.18e + 05	6.74e - 15	5.19e - 121
f_3	F_{mean}	4.53e + 00(+)	1.66e - 01(+)	4.73e + 00(+)	3.33e - 02(+)	1.27e + 04(+)	1.26e + 04(+)	7.53e + 00(+)	0.00e + 00
	SD	4.59e + 00	3.61e - 01	2.85e + 00	8.82e - 02	9.27e + 03	1.18e + 03	2.23e + 00	0.00e + 00
	Max	2.50e + 01	2.00e + 00	1.20e + 01	1.00e + 00	3.01e + 04	1.52e + 04	1.20e + 01	0.00e + 00
	Min	0.00e + 00	0.00e + 00	0.00e + 00	0.00e + 00	2.90e + 01	1.02e + 04	3.00e + 00	0.00e + 00
f_4	F_{mean}	1.32e + 05(+)	1.15e + 01(-)	8.43e + 05(+)	1.06e + 04(+)	1.10e + 05(+)	1.97e + 05(+)	1.17e + 05(+)	2.50e + 03
	SD	2.05e + 04	4.70e + 00	1.64e + 06	5.23e + 03	2.71e + 04	1.57e + 04	1.35e + 04	3.92e + 02
	Max	1.80e + 05	2.28e + 01	9.25e + 06	3.57e + 04	1.76e + 05	2.27e + 05	1.44e + 05	3.54e + 03
	Min	9.74e + 04	3.46e + 00	2.11e + 05	5.92e + 03	6.66e + 04	1.56e + 05	8.61e + 04	1.99e + 03
f_5	F_{mean}	1.83e + 03(+)	1.23e + 05(≈)	3.06e + 06(≈)	2.20e + 02(-)	1.57e + 03(≈)	2.14e + 03(+)	2.58e + 03(+)	1.67e + 03
	SD	1.65e + 02	6.67e + 05	1.60e + 07	1.81e + 02	1.58e + 02	1.48e + 02	9.88e + 01	2.35e + 02
	Max	2.14e + 03	3.65e + 06	8.79e + 07	1.15e + 03	1.88e + 03	2.48e + 03	2.76e + 03	2.02e + 03
	Min	1.52e + 03	1.01e + 03	1.06e + 03	1.18e + 02	1.30e + 03	1.82e + 03	2.37e + 03	1.09e + 03
f_6	F_{mean}	4.48e - 01(+)	1.95e + 01(+)	1.00e + 00(+)	5.51e - 04(+)	1.10e + 01(+)	1.24e + 01(+)	7.25e - 08(+)	0.00e + 00
	SD	1.99e - 01	7.76e - 02	4.93e - 01	1.53e - 04	4.29e + 00	2.54e - 01	6.02e - 08	0.00e + 00
	Max	9.80e - 01	1.96e + 01	1.68e + 00	9.17e - 04	1.68e + 01	1.29e + 01	2.99e - 07	0.00e + 00
	Min	1.83e - 01	1.93e + 01	4.82e - 09	3.24e - 04	2.78e + 00	1.16e + 01	1.32e - 08	0.00e + 00
f_7	F_{mean}	8.59e + 02(+)	7.55e + 02(+)	1.57e + 02(+)	1.81e + 02(+)	5.47e + 02(+)	9.05e + 02(+)	1.28e + 00(+)	0.00e + 00
	SD	2.65e + 01	7.99e + 01	7.30e + 00	1.95e + 01	7.44e + 01	3.43e + 01	1.26e + 00	0.00e + 00
	Max	9.13e + 02	8.99e + 02	1.75e + 02	2.25e + 02	6.83e + 02	9.48e + 02	5.03e + 00	0.00e + 00
	Min	7.89e + 02	5.58e + 02	1.44e + 02	1.37e + 02	3.91e + 02	4.01e + 02	2.74e - 08	0.00e + 00
f_8	F_{mean}	6.25e + 00(+)	1.94e + 01(+)	1.28e + 02(+)	2.93e + 01(+)	5.63e + 01(+)	7.02e + 01(+)	2.13e - 05(+)	0.00e + 00
	SD	8.54e - 01	3.99e + 00	1.61e + 01	7.69e + 00	8.70e + 00	1.27e + 00	1.71e - 05	0.00e + 00
	Max	8.65e + 00	2.53e + 01	1.64e + 02	5.66e + 01	6.94e + 01	7.28e + 01	8.67e - 05	0.00e + 00
	Min	4.74e + 00	9.78e + 00	1.03e + 02	1.43e + 01	3.12e + 01	6.80e + 01	2.85e - 06	0.00e + 00

TABLE 4: Continued.

F	DE	CMAES	JADE	jDE	PSOcf	PSOWFIPS	UPSO	ABC	CPDE
f_9									
F_{mean}	$6.65e-01(+)$	$3.28e-04(\approx)$	$3.03e-01(+)$	$1.45e-05(\approx)$	$1.05e+02(+)$	$1.19e+02(+)$	$3.17e-02(+)$	$2.00e-03(\approx)$	$1.15e-05$
SD	$1.13e-01$	$1.79e-03$	$2.58e-01$	$7.36e-06$	$8.23e+01$	$7.90e+00$	$9.78e-03$	$5.44e-03$	$2.99e-05$
Max	$9.20e-01$	$9.85e-03$	$9.21e-01$	$3.63e-05$	$2.70e+02$	$1.32e+02$	$5.49e-02$	$2.21e-02$	$9.86e-05$
Min	$4.57e-01$	$0.00e+00$	$6.50e-04$	$3.75e-06$	$1.60e-02$	$1.00e+02$	$1.82e-02$	$9.43e-15$	$0.00e+00$
f_{10}									
F_{mean}	$9.77e+01(-)$	$7.47e+01(-)$	$8.30e+01(-)$	$1.01e+02(-)$	$8.88e+02(+)$	$1.23e+03(+)$	$1.02e+02(-)$	$1.53e+02(\approx)$	$1.70e+02$
SD	$1.49e+00$	$2.86e+00$	$2.13e+00$	$2.06e+01$	$8.36e+02$	$9.04e+01$	$1.56e+01$	$2.92e+01$	$1.38e+02$
Max	$1.04e+02$	$8.06e+01$	$9.06e+01$	$2.02e+02$	$3.31e+03$	$1.43e+03$	$1.53e+02$	$2.37e+02$	$5.92e+02$
Min	$9.55e+01$	$7.04e+01$	$7.99e+01$	$9.39e+01$	$9.87e+01$	$1.02e+03$	$9.26e+01$	$9.71e+01$	$9.72e+01$
General merit over contender	8	4	7	5	9	10	8	8	

TABLE 5: The mean solutions, standard deviation, maximum, and minimum of the 30 trials obtained by various methods for 500-dimensional functions.

F	DE	CMAES	JADE	jDE	PSOcf	PSOWFIPS	UPSO	ABC	CPDE
f_1	F_{mean}	4.40e + 03(+)	2.26e - 27(+)	4.18e - 01(+)	3.14e + 01(+)	2.22e + 05(+)	3.58e + 05(+)	9.23e + 00(+)	3.75e - 07(+)
	SD	1.24e + 03	2.20e - 28	9.80e - 01	5.41e + 01	5.04e + 04	7.56e + 03	1.31e + 00	5.27e - 07
	Max	8.13e + 03	2.65e - 27	5.33e + 00	2.38e + 02	3.60e + 05	3.75e + 05	1.19e + 01	2.57e - 06
	Min	2.15e + 03	1.75e - 27	3.21e - 02	8.31e - 02	1.22e + 05	3.42e + 05	7.06e + 00	4.44e - 08
f_2	F_{mean}	9.04e + 05(+)	4.93e - 04(+)	2.02e + 08(+)	3.27e + 03(+)	5.62e + 07(+)	7.63e + 07(+)	1.94e + 03(+)	1.55e - 04(+)
	SD	2.92e + 05	3.66e - 04	3.23e + 07	7.98e + 03	1.09e + 07	1.82e + 06	2.23e + 02	3.43e - 04
	Max	1.74e + 06	1.50e - 03	3.11e + 08	4.14e + 04	7.97e + 07	7.96e + 07	2.48e + 03	2.07e - 03
	Min	4.98e + 05	1.01e - 05	1.63e + 08	2.62e + 01	3.89e + 07	7.27e + 07	1.37e + 03	1.44e - 06
f_3	F_{mean}	6.60e + 03(+)	6.66e - 01(+)	3.67e + 03(+)	9.02e + 02(+)	2.56e + 05(+)	3.54e + 05(+)	2.62e + 02(+)	2.90e + 02(+)
	SD	1.58e + 03	8.44e - 01	9.93e + 02	6.13e + 02	4.25e + 04	7.14e + 03	5.29e + 01	8.40e + 01
	Max	1.04e + 04	4.00e + 00	5.29e + 03	3.05e + 03	3.46e + 05	3.65e + 05	4.48e + 02	4.72e + 02
	Min	3.72e + 03	0.00e + 00	1.84e + 03	1.07e + 02	1.64e + 05	3.28e + 05	1.78e + 02	1.65e + 02
f_4	F_{mean}	3.64e + 06(+)	3.11e + 05(+)	8.33e + 07(+)	1.99e + 05(≈)	1.84e + 06(+)	4.11e + 06(+)	1.34e + 06(+)	2.17e + 06(+)
	SD	4.46e + 05	3.14e + 04	7.47e + 07	4.48e + 04	2.45e + 05	2.56e + 05	7.40e + 04	1.30e + 05
	Max	4.48e + 06	3.95e + 05	3.03e + 08	3.19e + 05	2.47e + 06	4.59e + 06	1.60e + 06	2.45e + 06
	Min	2.44e + 06	2.68e + 05	1.20e + 07	1.36e + 05	1.37e + 06	3.64e + 06	1.23e + 06	1.92e + 06
f_5	F_{mean}	1.14e + 04(≈)	7.75e + 06(+)	3.85e + 09(+)	1.96e + 03(-)	1.64e + 04(+)	1.37e + 04(+)	1.46e + 04(+)	1.20e + 04
	SD	1.40e + 03	1.67e + 06	2.10e + 09	1.36e + 03	2.23e + 03	5.89e + 02	8.23e + 02	2.43e + 02
	Max	1.42e + 04	1.45e + 08	1.15e + 11	9.08e + 03	2.09e + 04	1.44e + 04	1.54e + 04	1.52e + 04
	Min	8.68e + 03	1.53e + 04	7.16e + 03	1.36e + 03	1.27e + 04	1.18e + 04	1.21e + 04	1.42e + 04
f_6	F_{mean}	6.93e + 00(-)	1.94e + 01(+)	6.78e + 00(-)	1.56e + 00(-)	1.89e + 01(+)	1.82e + 01(+)	1.93e + 01(+)	2.01e - 03(-)
	SD	5.73e - 01	5.07e - 02	4.55e - 01	3.26e - 01	1.25e - 01	1.00e - 01	1.01e + 00	1.22e - 03
	Max	8.11e + 00	1.95e + 01	7.79e + 00	2.15e + 00	1.91e + 01	1.84e + 01	1.99e + 01	6.37e - 03
	Min	5.96e + 00	1.93e + 01	6.04e + 00	8.56e - 01	1.86e + 01	1.80e + 01	1.44e + 01	7.10e - 04
f_7	F_{mean}	2.30e + 03(+)	2.71e + 03(+)	1.68e + 03(+)	1.78e + 03(+)	3.46e + 03(+)	6.34e + 03(+)	3.21e + 03(+)	5.16e + 01(+)
	SD	1.11e + 03	1.77e + 03	4.54e + 01	1.22e + 02	2.22e + 02	7.35e + 01	1.43e + 02	6.82e + 00
	Max	4.33e + 03	3.12e + 03	1.79e + 03	2.04e + 03	3.91e + 03	6.45e + 03	3.56e + 03	6.44e + 01
	Min	1.06e + 03	2.40e + 03	1.58e + 03	1.41e + 03	2.94e + 03	6.17e + 03	2.95e + 03	2.87e + 01
f_8	F_{mean}	1.60e + 02(+)	1.94e + 02(+)	7.92e + 02(+)	1.92e + 02(+)	5.78e + 02(+)	6.08e + 02(+)	4.11e + 02(+)	4.17e - 02(+)
	SD	1.18e + 01	1.70e + 01	2.98e + 01	7.53e + 01	3.08e + 01	5.83e + 01	3.92e + 01	1.49e - 02
	Max	1.85e + 02	2.32e + 02	8.87e + 02	3.96e + 02	6.37e + 02	6.19e + 02	4.88e + 02	1.18e - 01
	Min	1.39e + 02	1.62e + 02	7.51e + 02	4.22e + 02	5.01e + 02	5.95e + 02	3.36e + 02	3.22e - 02

TABLE 5: Continued.

F	DE	CMAES	JADE	jDE	PSOcf	PSOWFIPS	UPSO	ABC	CPDE
f_9									
F_{mean}	$3.87e + 01(+)$	$9.85e - 04(\approx)$	$1.46e + 00(+)$	$1.45e + 00(+)$	$2.04e + 03(+)$	$3.19e + 03(+)$	$7.51e - 01(+)$	$8.46e - 04(\approx)$	0.00e + 00
SD	$1.35e + 01$	$3.07e - 03$	$5.58e - 01$	$1.68e + 00$	$3.88e + 02$	$8.63e + 01$	$6.14e - 02$	$4.55e - 03$	0.00e + 00
Max	$6.99e + 01$	$1.23e - 02$	$3.02e + 00$	$7.67e + 00$	$2.94e + 03$	$3.35e + 03$	$8.72e - 01$	$2.49e - 02$	0.00e + 00
Min	$1.69e + 01$	$7.77e - 16$	$4.53e - 01$	$1.20e - 02$	$1.36e + 03$	$2.99e + 03$	$6.07e - 01$	$5.90e - 09$	0.00e + 00
f_{10}									
F_{mean}	$1.07e + 03(+)$	4.72e + 02($\approx$)	$4.90e + 02(\approx)$	$7.22e + 02(\approx)$	$2.38e + 04(+)$	$3.63e + 04(+)$	$5.85e + 02(\approx)$	$1.14e + 03(+)$	$5.58e + 02$
SD	$1.15e + 02$	7.86e + 00	$1.02e + 01$	$1.07e + 02$	$4.62e + 03$	$1.53e + 03$	$1.08e + 02$	$1.05e + 02$	$4.51e + 02$
Max	$1.30e + 03$	4.99e + 02	$5.41e + 02$	$9.22e + 02$	$3.67e + 04$	$3.83e + 04$	$8.84e + 02$	$1.36e + 03$	$1.43e + 03$
Min	$8.86e + 02$	$4.61e + 02$	$4.84e + 02$	$5.58e + 02$	$1.22e + 04$	$3.27e + 04$	$4.97e + 02$	$9.11e + 02$	1.22e + 02
General merit over contender	7	8	7	4	10	10	9	7	

TABLE 6: The mean solutions, standard deviation, maximum, and minimum of the 30 trials obtained by various methods for 1000-dimensional functions.

F	DE	CMAES	JADE	jDE	PSOcf	PSOWFIPS	UPSO	ABC	CPDE
f_1	F_{mean}	7.21e + 04(+)	2.32e + 03(+)	1.03e + 04(+)	7.91e + 05(+)	1.26e + 06(+)	3.03e + 03(+)	1.37e − 02(+)	3.13e − 282
	SD	6.48e + 04	6.96e + 02	2.53e + 03	7.49e + 04	1.79e + 04	2.41e + 02	5.08e − 03	0.00e + 00
	Max	8.67e + 04	4.03e + 03	1.45e + 04	1.05e + 06	1.30e + 06	3.55e + 03	2.75e − 02	6.98e − 281
	Min	5.80e + 04	1.09e + 03	5.43e + 03	6.85e + 05	1.23e + 06	2.61e + 03	6.31e − 03	1.77e − 295
f_2	F_{mean}	2.93e + 07(+)	9.90e + 08(+)	4.03e + 06(+)	3.67e + 07(+)	5.58e + 08(+)	1.22e + 06(+)	6.32e + 00(+)	3.73e − 266
	SD	3.32e + 06	1.29e + 08	1.11e + 06	3.40e + 07	9.47e + 06	1.07e + 05	4.95e + 00	0.00e + 00
	Max	3.50e + 07	1.36e + 09	7.27e + 06	4.44e + 08	5.75e + 08	1.41e + 06	2.95e + 01	8.90e − 265
	Min	2.20e + 07	1.02e + 02	7.68e + 08	3.08e + 08	5.34e + 08	1.03e + 06	3.45e + 00	2.97e − 288
f_3	F_{mean}	7.94e + 04(+)	3.27e + 04(+)	1.66e + 04(+)	8.53e + 05(+)	1.26e + 06(+)	5.18e + 03(+)	1.34e + 03(+)	0.00e + 00
	SD	7.09e + 03	4.47e + 03	5.11e + 03	8.34e + 04	1.86e + 04	4.89e + 02	6.00e + 02	0.00e + 00
	Max	9.05e + 04	4.29e + 04	3.54e + 04	1.05e + 06	1.30e + 06	6.16e + 03	2.71e + 03	0.00e + 00
	Min	6.61e + 04	1.90e + 01	2.47e + 04	7.03e + 05	1.20e + 06	4.35e + 03	4.67e + 02	0.00e + 00
f_4	F_{mean}	1.47e + 07(+)	5.10e + 08(+)	8.07e + 05(−)	7.02e + 06(+)	1.65e + 07(+)	5.36e + 06(+)	7.26e + 06(+)	1.21e + 06
	SD	2.16e + 06	5.68e + 08	1.79e + 05	9.30e + 05	8.55e + 05	3.09e + 05	5.53e + 05	2.23e + 05
	Max	1.92e + 07	2.87e + 09	1.54e + 06	9.69e + 06	1.81e + 07	5.93e + 06	8.29e + 06	1.95e + 06
	Min	7.92e + 06	1.66e + 06	8.89e + 07	5.77e + 06	1.48e + 07	4.80e + 06	6.40e + 06	9.36e + 05
f_5	F_{mean}	2.49e + 04(≈)	4.25e + 16(+)	5.28e + 03(−)	3.68e + 04(+)	2.85e + 04(+)	2.88e + 04(+)	3.05e + 04(+)	2.64e + 04
	SD	2.37e + 03	2.33e + 17	3.44e + 03	6.03e + 03	1.25e + 03	1.77e + 03	2.69e + 02	3.41e + 03
	Max	2.97e + 04	3.85e + 10	1.27e + 18	4.84e + 04	3.08e + 04	3.16e + 04	3.09e + 04	3.30e + 04
	Min	2.12e + 04	3.22e + 04	1.60e + 04	2.83e + 04	2.54e + 04	2.49e + 04	2.99e + 04	1.82e + 04
f_6	F_{mean}	1.17e + 01(−)	1.02e + 01(−)	6.46e + 00(−)	1.94e + 01(+)	1.98e + 01(+)	1.99e + 01(+)	1.79e + 00(−)	1.60e + 01
	SD	2.94e − 01	3.60e − 01	8.90e − 01	6.86e − 02	4.12e − 02	1.03e − 02	1.69e − 01	2.32e − 01
	Max	1.23e + 01	1.07e + 01	8.24e + 00	1.96e + 01	1.99e + 01	2.00e + 01	2.24e + 00	1.65e + 01
	Min	1.11e + 01	1.94e + 01	9.54e + 00	1.92e + 01	1.97e + 01	1.99e + 01	1.52e + 00	1.56e + 01
f_7	F_{mean}	3.45e + 03(+)	4.86e + 03(+)	5.55e + 03(+)	8.41e + 03(+)	1.39e + 04(+)	7.41e + 03(+)	4.39e + 02(+)	0.00e + 00
	SD	2.81e + 02	2.26e + 02	9.02e + 01	1.83e + 02	1.03e + 02	2.87e + 02	2.81e + 01	0.00e + 00
	Max	4.18e + 03	5.36e + 03	5.72e + 03	8.70e + 03	1.41e + 04	7.93e + 03	5.12e + 02	0.00e + 00
	Min	2.93e + 03	4.46e + 03	5.40e + 03	7.94e + 03	1.36e + 04	6.91e + 03	3.76e + 02	0.00e + 00
f_8	F_{mean}	5.64e + 02(+)	4.88e + 02(+)	1.68e + 03(+)	1.32e + 03(+)	1.45e + 03(+)	9.51e + 02(+)	1.06e + 01(+)	0.00e + 00
	SD	1.84e + 01	2.64e + 01	5.11e + 01	4.10e + 01	7.43e + 00	6.63e + 01	2.17e + 00	0.00e + 00
	Max	6.13e + 02	5.34e + 02	1.79e + 03	1.40e + 03	1.46e + 03	1.10e + 03	1.56e + 01	0.00e + 00
	Min	5.38e + 02	4.23e + 02	1.58e + 03	1.25e + 03	1.43e + 03	8.09e + 02	7.24e + 00	0.00e + 00

TABLE 6: Continued.

F	DE	CMAES	JADE	jDE	PSOcf	PSOWFIPS	UPSO	ABC	CPDE
f_9									
F_{mean}	$6.50e + 02(+)$	$1.31e - 03(\approx)$	$2.48e + 01(+)$	$9.52e + 01(+)$	$7.17e + 03(+)$	$1.13e + 04(+)$	$2.91e + 01(+)$	$8.38e - 02(+)$	$1.86e - 04$
SD	$5.83e + 01$	$4.51e - 03$	$6.04e + 00$	$2.61e + 01$	$5.34e + 02$	$1.57e + 02$	$2.69e + 00$	$1.55e - 01$	$4.11e - 04$
Max	$7.82e + 02$	$2.21e - 02$	$3.91e + 01$	$1.52e + 02$	$8.22e + 03$	$1.16e + 04$	$3.44e + 01$	$8.03e - 01$	$1.32e - 03$
Min	$5.23e + 02$	$4.19e - 08$	$1.38e + 01$	$5.09e + 01$	$6.01e + 03$	$1.10e + 04$	$2.44e + 01$	$1.00e - 03$	$0.00e + 00$
f_{10}									
F_{mean}	$5.41e + 03(\approx)$	$9.85e + 02(-)$	$1.32e + 03(-)$	$2.82e + 03(-)$	$8.61e + 04(+)$	$1.39e + 05(+)$	$1.95e + 03(-)$	$3.20e + 03(-)$	$6.73e + 03$
SD	$3.62e + 02$	$2.22e + 01$	$9.07e + 01$	$3.58e + 02$	$9.06e + 03$	$3.51e + 03$	$2.28e + 02$	$1.76e + 02$	$7.42e + 03$
Max	$6.11e + 03$	$1.07e + 03$	$1.46e + 03$	$3.51e + 03$	$1.07e + 05$	$1.44e + 05$	$2.66e + 03$	$3.57e + 03$	$3.64e + 04$
Min	$4.54e + 03$	$9.71e + 02$	$1.15e + 03$	$1.83e + 03$	$6.51e + 04$	$1.31e + 05$	$1.68e + 03$	$2.79e + 03$	$3.82e + 03$
General merit over contender	6	7	6	2	10	10	8	6	

TABLE 7: The mean solutions, standard deviation, maximum, and minimum of the 30 trials obtained by various methods for CEC2013 problems.

F	DE	CMAES	JADE	jDE	PSOcf	PSOwFIPS	UPSO	ABC	CPDE
f_1	F_{mean}	$0.00e+00(\approx)$	$0.00e+00(\approx)$	$0.00e+00(\approx)$	$1.23e+02(+)$	$5.77e-09(+)$	$0.00e+00(\approx)$	$0.00e+00(\approx)$	$0.00e+00$
	SD	$2.11e-13$	$0.00e+00$	$0.00e+00$	$3.19e+02$	$2.57e-09$	$0.00e+00$	$0.00e+00$	$0.00e+00$
	Max	$4.54e-13$	$0.00e+00$	$0.00e+00$	$9.38e+02$	$1.03e-08$	$0.00e+00$	$0.00e+00$	$0.00e+00$
	Min	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.26e-09$	$0.00e+00$	$0.00e+00$	$0.00e+00$
f_2	F_{mean}	$1.44e-13(+)$	$0.00e+00(\approx)$	$3.38e-09(+)$	$5.43e+05(+)$	$6.06e+05(+)$	$2.20e+05(+)$	$2.70e+06(+)$	$0.00e+00$
	SD	$1.26e-13$	$0.00e+00$	$8.30e-09$	$8.28e+05$	$3.67e+05$	$1.17e+05$	$1.14e+06$	$0.00e+00$
	Max	$4.54e-13$	$0.00e+00$	$6.94e-09$	$3.97e+06$	$1.53e+06$	$5.57e+05$	$5.10e+06$	$0.00e+00$
	Min	$0.00e+00$	$0.00e+00$	$0.00e+00$	$2.12e+04$	$1.53e+05$	$4.89e+04$	$4.95e+05$	$0.00e+00$
f_3	F_{mean}	$6.13e+04(+)$	$1.71e+01(+)$	$1.09e+00(+)$	$2.59e+08(+)$	$6.79e+05(+)$	$8.63e+05(+)$	$7.93e+06(+)$	$2.02e-01$
	SD	$9.33e+04$	$2.09e+01$	$1.93e+00$	$4.94e+08$	$8.12e+05$	$1.33e+06$	$8.84e+06$	$3.45e-01$
	Max	$9.34e+05$	$7.85e+01$	$6.35e+00$	$2.19e+09$	$3.18e+06$	$5.30e+06$	$3.24e+07$	$1.71e+00$
	Min	$3.95e-11$	$0.00e+00$	$2.12e-03$	$2.75e+04$	$1.14e+03$	$1.61e+01$	$6.46e+05$	$0.00e+00$
f_4	F_{mean}	$1.37e-01(\approx)$	$0.00e+00(\approx)$	$3.35e-09(+)$	$2.36e+03(+)$	$1.00e+04(+)$	$9.84e+03(+)$	$1.02e+04(+)$	$0.00e+00$
	SD	$3.83e-01$	$0.00e+00$	$1.80e-09$	$2.58e+03$	$3.06e+03$	$3.83e+03$	$2.46e+03$	$0.00e+00$
	Max	$1.93e+00$	$0.00e+00$	$9.90e-09$	$9.39e+03$	$1.77e+04$	$1.94e+04$	$1.45e+04$	$0.00e+00$
	Min	$1.07e-05$	$0.00e+00$	$0.00e+00$	$2.14e+02$	$3.90e+03$	$4.07e+03$	$5.59e+03$	$0.00e+00$
f_5	F_{mean}	$9.09e-06(+)$	$0.00e+00(\approx)$	$0.00e+00(\approx)$	$9.26e+00(+)$	$2.02e-06(+)$	$0.00e+00(\approx)$	$1.40e-13(+)$	$0.00e+00$
	SD	$7.68e-06$	$0.00e+00$	$0.00e+00$	$2.86e+00$	$7.40e-07$	$0.00e+00$	$4.89e-14$	$0.00e+00$
	Max	$2.57e-05$	$0.00e+00$	$0.00e+00$	$1.11e+01$	$3.82e-06$	$0.00e+00$	$2.27e-13$	$0.00e+00$
	Min	$6.90e-07$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$8.60e-07$	$0.00e+00$	$1.13e-13$	$0.00e+00$
f_6	F_{mean}	$7.52e+00(+)$	$8.83e+00(+)$	$3.08e-02(+)$	$2.98e+01(+)$	$8.03e+00(+)$	$4.98e+00(+)$	$1.86e+00(+)$	$3.35e-09$
	SD	$4.22e+00$	$2.99e+00$	$4.96e-02$	$3.70e+01$	$3.37e+00$	$3.86e+00$	$3.26e+00$	$1.24e-08$
	Max	$9.81e+00$	$9.81e+00$	$2.42e-01$	$1.10e+02$	$9.83e+00$	$1.01e+01$	$9.81e+00$	$6.69e-08$
	Min	$0.00e+00$	$0.00e+00$	$1.59e-03$	$1.66e+00$	$6.72e-01$	$9.85e-02$	$2.17e-02$	$0.00e+00$
f_7	F_{mean}	$4.42e+01(+)$	$2.30e-01(+)$	$6.07e-03(\approx)$	$3.28e+01(+)$	$8.71e+00(+)$	$2.79e+01(+)$	$3.21e+01(+)$	$1.91e-03$
	SD	$4.23e+01$	$4.65e-01$	$1.11e-02$	$1.93e+01$	$5.99e+00$	$1.07e+01$	$1.05e+01$	$1.61e-03$
	Max	$1.60e+02$	$2.20e+00$	$4.56e-02$	$8.79e+01$	$2.66e+01$	$5.10e+01$	$5.68e+01$	$7.58e-03$
	Min	$8.87e-01$	$3.74e-04$	$1.76e-04$	$1.31e+01$	$1.42e+00$	$2.13e+00$	$1.10e+01$	$3.63e-05$
f_8	F_{mean}	$2.12e+01(+)$	$2.03e+01(\approx)$	$2.03e+01(\approx)$	$2.03e+01(\approx)$	$2.03e+01(\approx)$	$2.03e+01(\approx)$	$2.03e+01(\approx)$	$2.03e+01$
	SD	$2.66e-01$	$6.64e-02$	$6.58e-02$	$7.42e-02$	$6.10e-02$	$7.76e-02$	$6.82e-02$	$6.08e-02$
	Max	$2.15e+01$	$2.05e+01$	$2.04e+01$	$2.04e+01$	$2.04e+01$	$2.04e+01$	$2.05e+01$	$2.04e+01$
	Min	$2.04e+01$	$2.02e+01$	$2.02e+01$	$2.02e+01$	$2.02e+01$	$2.02e+01$	$2.02e+01$	$2.01e+01$

TABLE 7: Continued.

F	DE	CMAES	JADE	jDE	PSOcf	PSOWFIPS	UPSO	ABC	CPDE
f_9	F_{mean}	4.36e-01(-)	3.83e+00(+)	5.73e+00(+)	4.58e+00(+)	3.56e+00(+)	4.60e+00(+)	5.22e+00(+)	9.76e-01
	SD	5.47e-01	6.79e-01	9.10e-01	1.49e+00	8.79e-01	9.23e-01	6.15e-01	8.22e-01
	Max	1.87e+00	5.12e+00	8.16e+00	7.12e+00	5.05e+00	6.31e+00	6.41e+00	2.42e+00
	Min	1.10e-06	2.46e+00	4.16e+00	1.69e+00	1.88e+00	2.64e+00	3.91e+00	2.94e-04
f_{10}	F_{mean}	3.53e-01(-)	1.86e-02(-)	4.96e-02(-)	3.22e+01(+)	6.45e-01(+)	2.82e-01(-)	1.31e+00(+)	4.81e-01
	SD	1.59e-01	8.39e-03	3.15e-02	6.82e+01	1.50e-01	1.60e-01	2.13e-01	1.15e-01
	Max	5.78e-01	3.24e-02	1.18e-01	3.39e+02	8.87e-01	6.62e-01	1.94e+00	6.04e-01
	Min	1.47e-02	1.38e-03	0.00e+00	8.61e-02	2.97e-01	7.38e-02	1.03e+00	2.60e-02
f_{11}	F_{mean}	1.74e+01(+)	0.00e+00(-)	0.00e+00(-)	7.63e+00(+)	3.54e+00(+)	6.11e+00(+)	1.89e-15(-)	6.54e-01
	SD	3.61e+00	0.00e+00	0.00e+00	9.29e+00	1.52e+00	2.08e+00	1.03e-14	1.25e+00
	Max	2.49e+01	0.00e+00	0.00e+00	3.38e+01	7.64e+00	1.19e+01	5.68e-14	4.02e+00
	Min	8.35e+00	0.00e+00	0.00e+00	0.00e+00	1.04e+00	2.98e+00	0.00e+00	0.00e+00
f_{12}	F_{mean}	2.56e+01(+)	4.67e+00(+)	1.21e+01(+)	2.50e+01(+)	2.40e+01(+)	1.28e+01(+)	2.29e+01(+)	3.68e+00
	SD	4.04e+00	1.03e+00	2.88e+00	1.05e+01	4.45e+00	4.52e+00	5.08e+00	1.27e+00
	Max	3.30e+01	6.69e+00	1.73e+01	5.76e+01	3.08e+01	2.28e+01	3.79e+01	5.96e+00
	Min	1.80e+01	2.29e+00	7.07e+00	1.09e+01	1.08e+01	4.97e+00	1.37e+01	1.34e+00
f_{13}	F_{mean}	2.54e+01(+)	4.66e+00(≈)	1.34e+01(+)	2.85e+01(+)	2.13e+01(+)	1.95e+01(+)	3.25e+01(+)	4.48e+00
	SD	3.28e+00	1.83e+00	4.51e+00	1.37e+01	5.84e+00	5.07e+00	7.98e+00	8.07e-01
	Max	3.29e+01	1.04e+01	2.47e+01	5.50e+01	3.61e+01	3.20e+01	4.59e+01	5.03e+00
	Min	1.70e+01	4.52e-01	5.94e+00	2.13e+00	1.08e+01	9.48e+00	1.28e+01	1.98e+00
f_{14}	F_{mean}	1.01e+03(+)	1.85e+03(+)	6.24e-03(-)	3.03e+02(+)	5.11e+02(+)	3.01e+02(+)	1.25e-01(-)	1.75e+01
	SD	1.21e+02	4.02e+02	1.90e-02	1.90e+02	1.96e+02	1.12e+02	5.93e-02	1.87e+01
	Max	1.18e+03	2.83e+03	6.24e-02	7.25e+02	8.65e+02	5.81e+02	2.49e-01	8.86e+01
	Min	6.62e+02	0.00e+00	0.00e+00	1.02e+01	1.94e+02	7.84e+01	7.35e-06	2.11e-01
f_{15}	F_{mean}	1.31e+03(+)	4.66e+02(≈)	1.13e+03(+)	7.63e+02(+)	1.27e+03(+)	5.88e+02(+)	6.75e+02(+)	4.73e+02
	SD	1.52e+02	1.32e+02	1.96e+02	2.33e+02	1.90e+02	1.75e+02	1.53e+02	1.97e+02
	Max	1.58e+03	7.12e+02	1.46e+03	1.24e+03	1.62e+03	9.61e+02	9.44e+02	9.22e+02
	Min	9.32e+02	8.04e+01	6.58e+02	3.69e+02	7.72e+02	3.19e+02	3.63e+02	1.81e+02

TABLE 7: Continued.

F	DE	CMAES	JADE	jDE	PSOcf	PSOWFIPS	UPSO	ABC	CPDE
f_{16}									
	F_{mean}	1.07e + 00(≈)	1.15e + 00(+)	1.17e + 00(+)	5.33e − 01(−)	1.16e + 00(+)	3.06e − 01(−)	7.64e − 01(−)	1.03e + 00
	SD	2.09e − 01	1.88e − 01	1.98e − 01	2.54e − 01	2.20e − 01	1.13e − 01	1.54e − 01	2.21e − 01
	Max	1.42e + 00	1.40e + 00	1.53e + 00	1.11e + 00	1.53e + 00	6.61e − 01	1.04e + 00	1.49e + 00
	Min	5.88e − 01	7.27e − 01	7.35e − 01	1.80e − 01	7.06e − 01	1.11e − 01	5.01e − 01	6.54e − 01
f_{17}									
	F_{mean}	2.94e + 01(+)	1.01e + 01(−)	1.01e + 01(−)	1.16e + 01(−)	2.77e + 01(≈)	1.66e + 01(−)	9.07e + 00(−)	2.76e + 01
	SD	3.31e + 00	0.00e + 00	8.40e − 11	3.92e + 00	3.31e + 00	3.83e + 00	2.06e + 00	3.44e + 00
	Max	3.42e + 01	1.01e + 01	1.01e + 01	1.90e + 01	3.56e + 01	2.26e + 01	1.01e + 01	3.28e + 01
	Min	2.17e + 01	1.01e + 01	1.01e + 01	1.20e + 00	2.08e + 01	7.87e + 00	3.41e + 00	1.95e + 01
f_{18}									
	F_{mean}	3.48e + 01(≈)	1.87e + 01(−)	3.24e + 01(−)	2.27e + 01(−)	3.66e + 01(+)	1.80e + 01(−)	3.86e + 01(+)	3.47e + 01
	SD	4.01e + 00	1.89e + 00	4.02e + 00	8.23e + 00	3.35e + 00	2.80e + 00	6.00e + 00	3.50e + 00
	Max	4.17e + 01	2.13e + 01	3.94e + 01	4.03e + 01	4.39e + 01	2.30e + 01	4.67e + 01	4.04e + 01
	Min	2.60e + 01	1.46e + 01	2.42e + 01	7.23e + 00	3.09e + 01	1.29e + 01	2.26e + 01	2.79e + 01
f_{19}									
	F_{mean}	1.95e + 00(≈)	3.38e − 01(−)	4.37e − 01(−)	1.29e + 00(≈)	1.55e + 00(−)	5.99e − 01(−)	3.51e − 02(−)	1.98e + 00
	SD	2.89e − 01	4.73e − 02	7.40e − 02	3.66e + 00	2.82e − 01	1.41e − 01	2.38e − 02	2.98e − 01
	Max	2.44e + 00	4.27e − 01	5.69e − 01	2.06e + 01	2.08e + 00	9.54e − 01	9.58e − 02	2.70e + 00
	Min	1.34e + 00	2.07e − 01	2.97e − 01	1.19e − 01	7.92e − 01	4.06e − 01	9.85e − 05	1.35e + 00
f_{20}									
	F_{mean}	2.66e + 00(+)	2.36e + 00(≈)	3.17e + 00(+)	3.30e + 00(+)	3.05e + 00(+)	3.13e + 00(+)	3.20e + 00(+)	2.27e + 00
	SD	3.20e − 01	4.20e − 01	2.64e − 01	6.11e − 01	2.85e − 01	4.06e − 01	3.28e − 01	2.54e − 01
	Max	3.44e + 00	3.51e + 00	3.70e + 00	4.01e + 00	3.46e + 00	3.71e + 00	3.63e + 00	2.77e + 00
	Min	2.12e + 00	1.82e + 00	2.67e + 00	1.62e + 00	2.42e + 00	2.27e + 00	2.41e + 00	1.72e + 00
f_{21}									
	F_{mean}	3.66e + 02(≈)	3.93e + 02(+)	3.86e + 02(+)	3.66e + 02(≈)	2.73e + 02(−)	1.92e + 02(−)	1.56e + 02(−)	3.46e + 02
	SD	7.58e + 01	3.65e + 01	5.07e + 01	7.58e + 01	7.82e + 01	6.86e + 01	5.74e + 01	9.00e + 01
	Max	4.00e + 02	4.00e + 02	4.00e + 02	4.00e + 02	4.00e + 02	4.00e + 02	3.01e + 02	4.00e + 02
	Min	2.00e + 02	2.00e + 02	2.00e + 02	2.00e + 02	2.00e + 02	1.00e + 02	4.68e + 01	2.00e + 02
f_{22}									
	F_{mean}	1.07e + 03(+)	3.37e + 00(−)	5.67e + 01(−)	3.84e + 02(+)	4.26e + 02(+)	4.41e + 02(+)	9.87e + 00(−)	1.75e + 02
	SD	1.60e + 02	3.54e + 00	1.66e + 01	1.66e + 02	1.41e + 02	1.52e + 02	6.39e + 00	9.90e + 01
	Max	1.31e + 03	9.22e + 00	1.04e + 02	7.03e + 02	7.42e + 02	6.78e + 02	2.82e + 01	4.28e + 02
	Min	7.02e + 02	2.38e − 07	2.66e + 01	4.86e + 01	1.81e + 02	3.90e + 01	1.17e − 02	2.41e + 01
f_{23}									
	F_{mean}	1.25e + 03(+)	5.32e + 02(≈)	1.36e + 03(+)	9.68e + 02(+)	1.34e + 03(+)	7.95e + 02(+)	9.96e + 02(+)	4.92e + 02
	SD	1.56e + 02	1.52e + 02	2.68e + 02	2.70e + 02	2.07e + 02	1.49e + 02	1.54e + 02	2.73e + 02
	Max	1.72e + 03	8.01e + 02	1.79e + 03	1.50e + 03	1.61e + 03	1.12e + 03	1.28e + 03	1.07e + 03
	Min	9.69e + 02	1.90e + 02	7.36e + 02	2.39e + 02	6.68e + 02	5.56e + 02	6.37e + 02	1.17e + 02

TABLE 7: Continued.

F	DE	CMAES	JADE	jDE	PSOcf	PSOWFIPS	UPSO	ABC	CPDE
f_{24}									
F_{mean}	2.03e + 02(≈)	4.04e + 02(+)	2.03e + 02(≈)	2.10e + 02(+)	2.19e + 02(+)	2.02e + 02(≈)	2.09e + 02(≈)	1.32e + 02(-)	2.01e + 02
SD	3.36e + 02	1.77e + 02	5.26e + 00	1.42e + 01	4.56e + 00	1.93e + 01	2.20e + 01	6.72e + 00	1.54e + 01
Max	2.09e + 02	7.52e + 02	2.12e + 02	2.20e + 02	2.27e + 02	2.16e + 02	2.21e + 02	1.46e + 02	2.10e + 02
Min	2.00e + 02	1.16e + 02	1.87e + 02	1.69e + 02	2.08e + 02	1.35e + 02	1.30e + 02	1.21e + 02	1.21e + 02
f_{25}									
F_{mean}	2.01e + 02(≈)	2.59e + 02(+)	2.00e + 02(≈)	2.18e + 02(≈)	2.18e + 02(≈)	2.12e + 02(≈)	2.16e + 02(≈)	1.56e + 02(≈)	2.01e + 02
SD	2.28e + 02	6.76e + 01	9.67e + 00	1.88e + 00	4.47e + 00	2.39e + 00	2.42e + 00	1.58e + 01	2.25e + 02
Max	2.04e + 02	4.80e + 02	2.08e + 02	2.21e + 02	2.27e + 02	2.15e + 02	2.19e + 02	1.98e + 02	2.04e + 02
Min	2.00e + 02	2.00e + 02	1.49e + 02	2.12e + 02	2.05e + 02	2.07e + 02	2.09e + 02	1.29e + 02	2.00e + 02
f_{26}									
F_{mean}	1.58e + 02(+)	2.53e + 02(≈)	1.36e + 02(+)	1.88e + 02(+)	2.15e + 02(+)	1.31e + 02(+)	1.27e + 02(+)	1.28e + 02(+)	1.17e + 02
SD	3.75e + 01	1.51e + 02	4.32e + 01	2.91e + 01	6.40e + 01	6.99e + 00	1.14e + 01	8.28e + 00	1.08e + 01
Max	2.00e + 02	8.12e + 02	2.00e + 02	2.00e + 02	3.23e + 02	1.53e + 02	1.60e + 02	1.42e + 02	1.30e + 02
Min	1.10e + 02	1.05e + 02	1.02e + 02	1.05e + 02	1.11e + 02	1.14e + 02	1.07e + 02	1.16e + 02	1.01e + 02
f_{27}									
F_{mean}	3.17e + 02(≈)	3.45e + 02(+)	3.01e + 02(≈)	4.74e + 02(+)	5.59e + 02(+)	4.94e + 02(+)	5.31e + 02(+)	3.92e + 02(+)	3.16e + 02
SD	5.45e + 01	5.59e + 01	4.57e + 00	1.85e + 01	5.57e + 01	3.83e + 01	2.98e + 01	1.29e + 01	5.18e + 01
Max	4.81e + 02	5.13e + 02	3.25e + 02	5.02e + 02	6.52e + 02	5.36e + 02	5.69e + 02	4.00e + 02	5.06e + 02
Min	3.00e + 02	3.00e + 02	3.00e + 02	4.22e + 02	4.43e + 02	3.18e + 02	4.06e + 02	3.55e + 02	3.00e + 02
f_{28}									
F_{mean}	2.86e + 02(≈)	1.39e + 03(+)	3.00e + 02(+)	2.93e + 02(≈)	3.81e + 02(+)	2.83e + 02(≈)	2.46e + 02(≈)	1.63e + 02(-)	2.66e + 02
SD	5.07e + 01	1.22e + 03	0.00e + 00	3.65e + 01	1.43e + 02	5.03e + 01	8.99e + 01	7.80e + 01	7.58e + 01
Max	3.00e + 02	3.88e + 03	3.00e + 02	3.00e + 02	7.20e + 02	3.00e + 02	3.00e + 02	3.00e + 02	3.00e + 02
Min	1.00e + 02	3.00e + 02	3.00e + 02	1.00e + 02	3.00e + 02	1.19e + 02	1.00e + 02	1.00e + 02	1.00e + 02
General merit over contender	10	17	2	8	18	19	10	7	

performs better for functions f_{22} and f_{27} . ABC performs better for functions f_{21} , f_{24} , f_{25} , and f_{28} .

5. Conclusion

In this paper, we have described a new algorithm. Its principle of operation is based on the transformation of the cloud particles three states. It employs the phase transformation to realize better exploitation and employs cloud differential mutation to make the cloud particles capable of tackling the premature convergence. The performance of CPDE has been evaluated by using the set of benchmark functions (unimodal, basic multimodal, expanded multimodal, and hybrid composition). As shown in the simulation results, CPDE generally performs better than other algorithms for unimodal and multimodal problems. However, it may be noted that CPDE has not performed well on a few hybrid composition functions. The shortcoming may be due to the worse phase transformation mechanism of the cloud particles. In future work, we will focus on designing adaptive strategies for phase transformation to effectively provide better performance for the case of hybrid composite functions. We will also derive new ways to improve the exploration of the algorithm with the lowest expense.

Conflict of Interests

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

Acknowledgments

The authors express their sincere thanks to the reviewers for their significant contributions to the improvement of the final paper. This research is partly supported by the support of the National Natural Science Foundation of China (no. 61272283), Science & Research Plan Project of Shaanxi Province Department of Education (no. 12JK0737), and Xi'an Science and Technology Project (no. CXY1437 (5)).

References

- [1] 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.
- [2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, IEEE, Perth, Australia, December 1995.
- [3] R. Storn and K. V. 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.
- [4] B. Basturk and D. Karaboga, "An artificial bee colony (ABC) algorithm for numeric function optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '06)*, pp. 12–14, Indianapolis, Ind, USA, May 2006.
- [5] L. N. De Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, Springer, Berlin, Germany, 2002.
- [6] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, no. 6, pp. 39–42, 2000.
- [7] 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.
- [8] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.
- [9] K. E. Parsopoulos and M. N. Vrahatis, "UPSO—a unified particle swarm optimization scheme," in *Lecture Series on Computational Sciences*, vol. 1, pp. 868–873, VSP International Science Publishers, Zeist, The Netherlands, 2004.
- [10] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.
- [11] S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer and its adaptive variant," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 6, pp. 1272–1282, 2005.
- [12] C. Li, S. Yang, and T. T. Nguyen, "A self-learning particle swarm optimizer for global optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 3, pp. 627–646, 2012.
- [13] W.-N. Chen, J. Zhang, Y. Lin et al., "Particle swarm optimization with an aging leader and challengers," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 2, pp. 241–258, 2013.
- [14] W. H. Lim and N. A. M. Isa, "Teaching and peer-learning particle swarm optimization," *Applied Soft Computing*, vol. 18, pp. 39–58, 2014.
- [15] W. H. Lim and N. A. Mat Isa, "Bidirectional teaching and peer-learning particle swarm optimization," *Information Sciences*, vol. 280, pp. 111–134, 2014.
- [16] A. R. Jordehi, "Enhanced leader PSO (ELPSO): a new PSO variant for solving global optimisation problems," *Applied Soft Computing*, vol. 26, pp. 401–417, 2015.
- [17] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [18] S. Z. Zhao, J. J. Liang, P. N. Suganthan, and M. F. Tasgetiren, "Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 3845–3852, Hong Kong, China, June 2008.
- [19] 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.
- [20] 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.
- [21] Z. Jingqiao and C. S. Arthur, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–957, 2009.

- [22] 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.
- [23] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 1679–1696, 2011.
- [24] Y. Chen, W. Xie, and X. Zou, "A binary differential evolution algorithm learning from explored solutions," *Neurocomputing*, vol. 149, pp. 1038–1047, 2015.
- [25] Y.-J. Zheng, X.-L. Xu, H.-F. Ling, and S.-Y. Chen, "A hybrid fireworks optimization method with differential evolution operators," *Neurocomputing*, vol. 148, pp. 75–82, 2015.
- [26] H. Guo, Y. Li, J. Li, H. Sun, D. Wang, and X. Chen, "Differential evolution improved with self-adaptive control parameters based on simulated annealing," *Swarm and Evolutionary Computation*, vol. 19, pp. 52–67, 2014.
- [27] 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.
- [28] M. Asafuddoula, T. Ray, and R. Sarker, "An adaptive hybrid differential evolution algorithm for single objective optimization," *Applied Mathematics and Computation*, vol. 231, pp. 601–618, 2014.
- [29] I. de Falco, A. Della Cioppa, D. Maisto, U. Scafuri, and E. Tarantino, "An adaptive invasion-based model for distributed differential evolution," *Information Sciences*, vol. 278, pp. 653–672, 2014.
- [30] L. d. Coelho, H. V. Ayala, and V. C. Mariani, "A self-adaptive chaotic differential evolution algorithm using γ distribution for unconstrained global optimization," *Applied Mathematics and Computation*, vol. 234, pp. 452–459, 2014.
- [31] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "A self-adaptive combined strategies algorithm for constrained optimization using differential evolution," *Applied Mathematics and Computation*, vol. 241, pp. 267–282, 2014.
- [32] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Springer, Berlin, Germany, 2005.
- [33] A. W. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution," in *AI 2004: Advances in Artificial Intelligence*, vol. 3339 of *Lecture Notes in Comput. Sci.*, pp. 861–872, Springer, Berlin, Germany, 2004.
- [34] F. A. Wang, *Cloud*, Meteorological Press, Beijing, China, 2002 (Chinese).
- [35] H. Hermawan, D. Ramdan, and J. R. P. Djuansjah, "Metals for biomedical applications," in *Biomedical Engineering—From Theory to Applications*, R. Fazel-Rezai, Ed., chapter 17, InTech, Rijeka, Croatia, 2011.
- [36] D. Y. Li, "Uncertainty in knowledge representation," *Engineering Sciences*, vol. 2, no. 10, pp. 73–79, 2000 (Chinese).
- [37] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, vol. 9, no. 6, pp. 448–462, 2005.
- [38] M. M. Ali and A. Törn, "Population set-based global optimization algorithms: some modifications and numerical studies," *Computers & Operations Research*, vol. 31, no. 10, pp. 1703–1725, 2004.
- [39] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernandez-Daz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," Tech. Rep. 201212, Zhengzhou University, Zhengzhou, China; Nanyang Technological University, Singapore, 2013.
- [40] 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.
- [41] W. Li, L. Wang, Q. Jiang, X. Hei, and B. Wang, "Cloud particles evolution algorithm," *Mathematical Problems in Engineering*, vol. 2015, Article ID 434831, 21 pages, 2015.