

Research Article

Cloud Particles Evolution Algorithm

Wei Li, Lei Wang, Qiaoyong Jiang, Xinhong Hei, and Bin Wang

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

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

Received 10 November 2014; Revised 13 January 2015; Accepted 13 January 2015

Academic Editor: Jyh-Hong Chou

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.

Many evolutionary algorithms have been paid attention to by the researchers and have been applied to solve optimization problems. This paper presents a new optimization method called cloud particles evolution algorithm (CPEA) to solve optimization problems based on cloud formation process and phase transformation of natural substance. 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 cloud is composed of descript and independent particles in this algorithm. The cloud particles use phase transformation of three states to realize the global exploration and the local exploitation in the optimization process. Moreover, the cloud particles not only realize the survival of the fittest through competition mechanism but also ensure the diversity of the cloud particles by reciprocity mechanism. The effectiveness of the algorithm is validated upon different benchmark problems. The proposed algorithm is compared with a number of other well-known optimization algorithms, and the experimental results show that cloud particles evolution algorithm has a higher efficiency than some other algorithms.

1. Introduction

Many real-world problems which are classified as global optimization problems are very complex and are quite difficult to solve. Various optimization algorithms are developed to solve these problems. Most of these algorithms learn and imitate a variety of intelligent behavior in nature. These algorithms exploit a set of potential solutions and detect the optimal solution through cooperation and competition among the individuals of the population [1]. For example, Genetic algorithm [2] proposed by Professor Holland is a computational model of biological evolution which imitates natural selection and genetic mechanism. Ant colony algorithm [3] is a heuristic bionic optimization algorithm which imitates the foraging behavior of the ant colony. In the ant colony algorithm, an individual is called an artificial ant. The artificial ants search probabilistically in the solution space to create candidate solutions. The evaluating and the updating of these candidate solutions are based on the pheromone associated with each one of them. The candidate solution with the maximum amount of pheromone is considered to be the optimal solution of the problem. Particle swarm optimization (PSO) [4] is a bionic intelligent computing method proposed

by Kennedy and Eberhart which imitates the flying and the foraging behavior of birds. In PSO, the optimal solution is called a leader. The leader guides each particle to search. Besides, the particles cooperate with each other to realize exploration. These behaviors are directed by a combination of swarm's previous best (*gBest*) and their own previous best (*pBest*) [5]. The artificial bee colony algorithm (ABC) [6, 7] is a nonderivative optimization method based on self-organizational models of bee and swarm intelligence. In ABC, bees are divided into three groups which are called "employed" bees, "onlooker" bees, and "scouts" bees. "Onlookers" bees look for food sources. Food source is the search goal of bees. "Employed" bees work on food source. In addition, "scouts" bees search for the new food source near the hive. Artificial immune system (AIS) [8] is an adaptive system inspired by the immunology which imitates human immunological functions, principles, and models to solve complex problems. In AIS, the external organisms are called antigen and the internal organisms are called antibodies. Immune clonal operation provides multistrategy conditions for recombination and mutation [9]. The clonal selection selects the excellent antibodies from the subpopulation generated by clonal proliferation.

This paper which is inspired by the natural phenomena of cloud formation presents a novel optimization algorithm to solve optimization problems called cloud particles evolution algorithm (CPEA). The proposed algorithm uses phase transformation mechanism to imitate the formation and diversification of the cloud to solve the optimization problems. It is well known that the species evolution is influenced by individual fitness, living environment, competition, and cooperation between the individuals. Therefore, the algorithm introduces competition and reciprocal mechanism in the evolution process so as to strengthen the viability of the cloud particles. The performance of the cloud particles evolution algorithm (CPEA) is tested on the classical optimization problems. The results are compared with other algorithms in terms of best function value and the number of function evaluations.

The remainder of this paper is organized as follows. In Section 2, the proposed CPEA and the concepts are introduced in detail. In Section 3, the performance of the proposed algorithm is validated by different optimization problems. The conclusion and future research are drawn in Section 4.

2. Cloud Particles Evolution Algorithm

2.1. Cloud Formation. The idea of the proposed CPEA is inspired from cloud formation. The state of the cloud particles' changing process [10–12] is shown in Figure 1.

After the wet air rises, it will go through a set of changes because of some certain reasons temperature and pressure. Then, the water vapor is formed. Water vapor plays an important role in this process. When the water vapor is saturated, it will adsorb on the cloud condensation nuclei (CCN) and the initial cloud particles are formed. 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 colliding. Then the larger cloud particles are formed. Ice phase is generated when the temperature is below 0°C at this time. The emergence of ice crystals damages the stable status of cloud phase structure. Ice crystals grow up quickly to be the snow crystals because of the condensation of the water vapor. On the one hand, water vapor transfers to ice crystals and ice crystals begin to grow up as supercooled water droplets evaporate. On the other hand, the supercooled water droplets will be frozen after the colliding to make the snow particles grow bigger. If the supercooled water vapor which has been collided is too many, the snow crystals will be transferred to be solid substances. Then, the solid substances such as snow and hail will be dropped. If the solid substances descent in the area which has a higher temperature over 0°C , they will melt into raindrop. The water in our planet will be irradiated by the sunshine, and it will become the water vapor again in the air. The cloud is formed and changed like this.

2.2. Phase and Phase Transformation. There are various kinds of substance in nature. Substances commonly exist in three states: solid, liquid, and gaseous. Figure 2 is a simple diagram to show how water transforms among the three states. Evaporation, condensation, and melting are the three ways for water

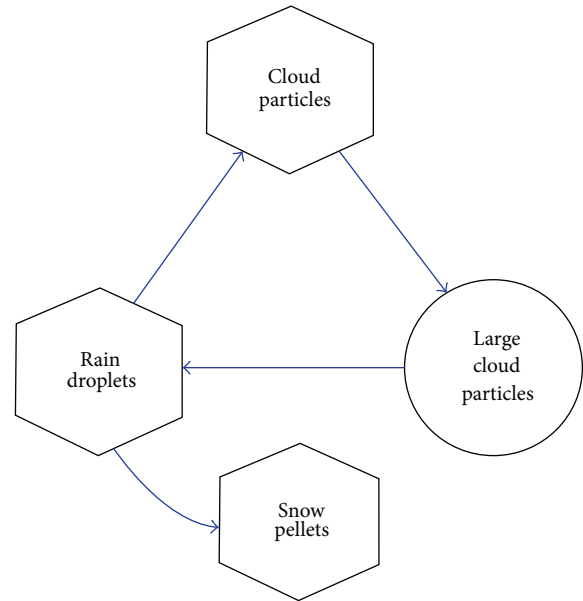


FIGURE 1: Cloud particles transformation process.

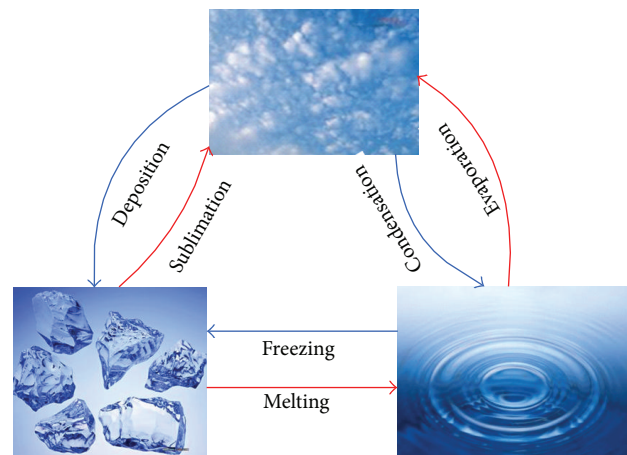


FIGURE 2: Three phases of water.

transforming. In order to describe the different forms of the substance, the phase can be used as a symbol for states of substances. In thermodynamics, phase is defined as some parts which have the same quality [13]. A phase transformation is the transition which is from one phase to another in thermodynamics.

Phase transformation is relevant to Gibbs free energy. In thermodynamics, the Gibbs free energy [14] is a thermodynamic potential that measures the “useful” or process-initiating work obtainable from a thermodynamic system at a constant temperature and pressure.

2.3. Population Evolution Mechanism. Darwin pointed out that mutation and natural selection are the two important factors for the evolution, which explained many natural phenomena successfully [15]. Nowak from Harvard University

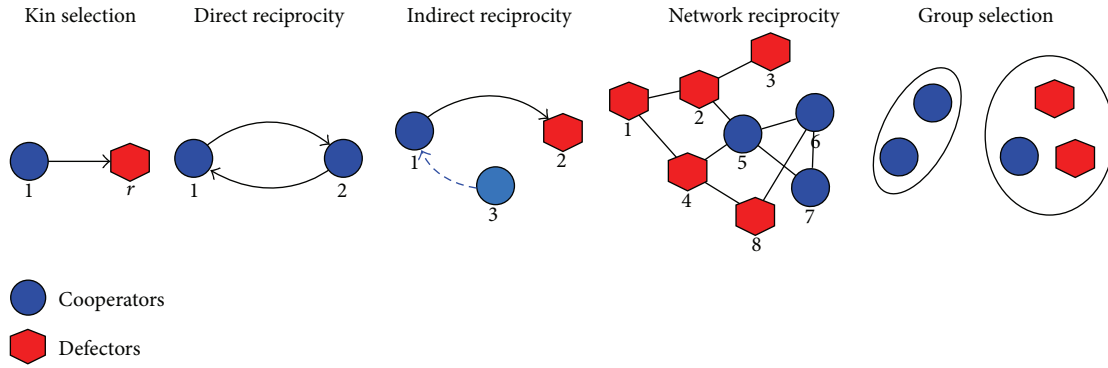


FIGURE 3: Five mechanisms of cooperation evolution [16].

ranked the cooperation as the third important factor of the species evolution. Evolution is a unity of the cooperation and the opposition [16]. Every gene, cell, and organism of the individual should strengthen its own evolution process at the expense of beating its rivals. Therefore, individuals often conflict with each other because of limited resources [17]. However, the complex life organizational structure in different levels is produced in the process of evolution, which indicates that the cooperation exists in the evolution. Therefore, Nowak summarizes five mechanisms for the cooperation evolution such as kin selection, direct reciprocity, indirect reciprocity, network reciprocity, and group selection (shown in Figure 3).

Haldane believes that the altruism exists in the evolution [18]. According to the study of the social insect, Hamilton [19] used mathematic model for quantitative description and defined the Hamilton rules. The rule states natural selection favors cooperation if the donor and the recipient have kin relationships. Direct reciprocity requires several repeated encounters between two individuals which can offer help with each other. The direct reciprocity theory framework is repeated Prisoner’s Dilemma. The classical game strategy is tit for tat, TFT [20]. The interactions among individuals in nature are often asymmetrical and fleeting. Therefore, indirect reciprocity is a more prevalent reciprocity form. Indirect reciprocity refers to establishing a good reputation by helping others, which will be rewarded by others later. In indirect reciprocity, one individual acts as donor while the other acts as recipient. Indirect reciprocity will favor the cooperation evolution, if q ; the probability of one individual’s reputation exceeds the cost-to-benefit ratio of the altruistic act, namely, $q > c/b$ [21]. Network reciprocity is a new form of reciprocity. Cooperators can form network clusters in order to help each other. The cooperation mechanism of this spatial reciprocity is called network reciprocity. Group selection is a minimalist stochastic model [22]. The population is divided into several groups. Cooperators help each other in their own groups while defectors do not help. The pure cooperator groups grow faster than the pure defector groups while the defector groups reproduce faster than the cooperator groups in any mixed group. Therefore, defectors are often selected in the lower level (within groups) while cooperators are often selected in the higher (between groups) level.

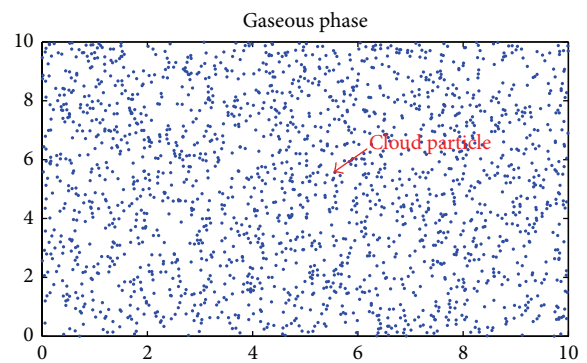


FIGURE 4: Cloud gaseous phase.

2.4. Description of the Proposed Algorithm. The main goal of the cloud particles evolution system is to realize the evolution of the cloud particles through phase transformation. In cloud particles evolution model, cloud particles have three phases: cloud gaseous phase, cloud liquid phase, and cloud solid phase. Cloud gaseous phase indicates gaseous cloud particles which are distributed in the entire search space. Cloud liquid phase refers to the rain in nature. Cloud solid phase represents hailstones in nature.

The proposed algorithm which is similar to other meta-heuristic 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 4). The best individual (best cloud particle) is chosen as a seed or nucleus. Then, the cloud particles are going to have a condensational growth or collisions and coalescence growth (shown in Figure 5). If the population evolves successfully, the cloud particles are going to change from cloud gaseous state to liquid state by condensation. Cloud changes from gaseous phase to cloud liquid phase. This process is called phase transformation. When cloud particles are in liquid state, cloud particles transform from global exploration to local exploitation gradually. In this process, according to the idea of the indirect reciprocity mentioned before, if $q > c/b$ the algorithm will favor the cooperation evolution or it will change from the liquid phase to solid phase by phase transformation. When cloud particles are in solid

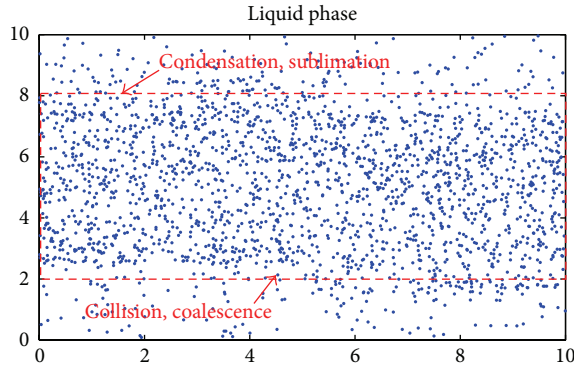


FIGURE 5: Cloud liquid phase.

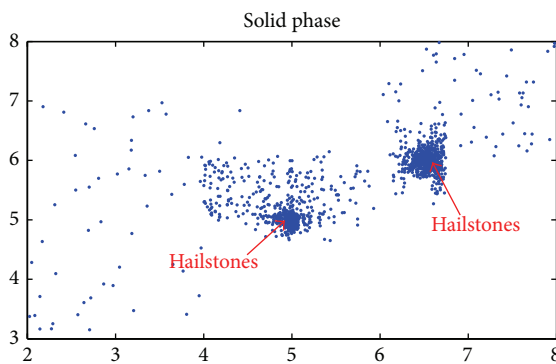


FIGURE 6: Cloud solid phase.

state (shown in Figure 6), the cloud particles condense into ice crystals at the beginning, and then ice crystals curdle into hailstones. The hailstone is the best solution.

It can be seen from the biological evolutionary behavior that the individuals compete to survive. However, competitive ability is always asymmetry because the different individual has different fitness. Usually, it is easier for disadvantaged individuals to compete with the individuals in the same situation while they seldom compete with the advantaged individuals [23]. Therefore, competition and cooperation which are two different survival strategies for living organisms are formed in order to adapt to the environment. In the CPEA model, according to the natural phenomena that the population develops cooperatively when they evolve, these two strategies can be shown as follows. The populations carry out the competitive evolution in cloud gaseous phase. In cloud liquid phase, the populations employ competitive mechanism for improving local exploitation ability and employ reciprocal mechanism to ensure the diversity of the population. Cloud solid phase indicates the algorithm has found the optimal solution area; therefore the algorithm realizes fast convergence by competitive evolution. The optimization process of CPEA model is shown in Figure 7.

In CPEA, the initialization, phase transformation driving force, condensation operation, reciprocity operation, and solidification operation are described as follows.

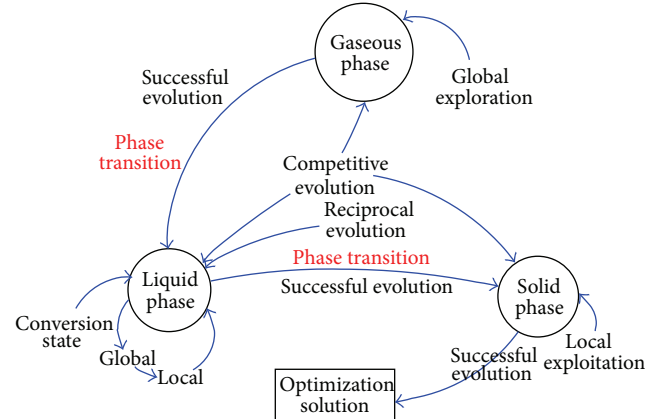


FIGURE 7: Optimization process of CPEA model.

2.4.1. 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 [24]. 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 another word, 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 m dimensional optimization problems, a cloud particle is an array of $1 \times m$. This array is defined as follows:

$$\text{Cloud particle} = [x_1, x_2, \dots, x_m]. \quad (1)$$

To start the optimization algorithm, a matrix of cloud particles which size is $N \times m$ is generated by cloud generator (i.e., population of cloud particles). Cloud particles generation is described as follows: $\Phi(\mu, \delta)$ is the normal random variable which has an expectation μ and a variance δ ; N is the number of cloud particle; X is the cloud particle generated by cloud generator. Consider

$$S = \{s_i \mid \Phi(E_n, H_e), i = 1 \dots N\}, \quad (2)$$

$$X = \{x_i \mid \Phi(E_x, s_i), s_i \in S, i = 1 \dots N\}.$$

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

Population of cloud particles

$$= \begin{bmatrix} \text{Cloudparticle}_1 \\ \text{Cloudparticle}_2 \\ \vdots \\ \text{Cloudparticle}_N \end{bmatrix} = \begin{bmatrix} X_1^1 & X_2^1 & X_3^1 & \cdots & X_m^1 \\ X_1^2 & X_2^2 & X_3^2 & \cdots & X_m^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_1^N & X_2^N & X_3^N & \cdots & X_m^N \end{bmatrix}. \quad (3)$$

Each of the decision variable values (x_1, x_2, \dots, x_m) can represent real values for continuous or discrete problems. The fitness of a cloud particle is calculated by the evaluation of objective function (F) given as

$$F_i = f(x_1^1, x_2^1, \dots, x_i^1) \quad i = 1, 2, 3, \dots, m. \quad (4)$$

In the function above, N and m are the number of cloud particles and the number of decision variables. The cloud particles which 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 selects a cloud particle as the nucleus. Therefore, there are N_p nuclei.

2.4.2. Phase Transformation Driving Force. The phase transformation driving force (PT) can reflect the evolution extent of the population. In the proposed algorithm, phase transformation driving force is the factor of deciding phase transform. During the evolution, the energy possessed by a cloud particle is called its evolution potential. Phase transformation driving force is the evolution potential energy difference between the cloud particles of new generation and the cloud particles of old generation. The phase transformation driving force is defined as follows:

$$EP_i = \text{bestindividual}_i \quad i = 1, 2, 3, \dots, \text{gen} \quad (5)$$

$$PT = \sqrt{\frac{(EP_{\text{genindex}-1} - \overline{EP})^2 + (EP_{\text{genindex}} - \overline{EP})^2}{2}} \quad (6)$$

$$\overline{EP} = \frac{EP_{\text{genindex}-1} + EP_{\text{genindex}}}{2}.$$

The cloud evolution system realizes phase transformation if $PT \leq \delta$. EP represents the evolution potential of cloud particles. *gen* represents the maximum generation of the algorithm. *bestindividual* represents the best solution of the current generation. *genindex* is the number of the current evolution.

2.4.3. Condensation Operation. The population realizes the global exploration in cloud gaseous phase by condensation operation. The population takes the best individual as the nucleus in the process of the condensation operation. In the cloud gaseous phase, the condensation growth space of cloud particle is calculated by the following equations. *cd* is the condensation factor. *generation* is the number of current evolution. E_{n_i} is the entropy of the i th subpopulation. H_{e_i} is hyperentropy of the i th subpopulation. Consider

$$E_{n_i} = \frac{|\text{nucleus}_i|}{\sqrt{\text{generation}}}, \quad (7)$$

$$H_{e_i} = \frac{E_{n_i}}{\text{cd}}. \quad (8)$$

2.4.4. Reciprocity Operation. If the evolution potential of cloud particles between the two generations is less than a

given threshold in cloud liquid phase, the algorithm will determine whether to employ the reciprocal evolution. If the reciprocal evolution condition is met, the reciprocal evolution is employed. Otherwise the algorithm carries out phase transformation. In reciprocity operation, the probability of one individual's reputation is calculated by u . $u \in [0, 1]$ is a uniformly random number. The cost-to-benefit ratio of the altruistic act is $(1 - 0.618) \approx 0.38$ and 0.618 is the golden ratio. N_p represents the number of the subpopulations. nucleus_i is the nucleus of the i th subpopulation. Consider

$$\text{nucleus}_{N_p} = \frac{\sum_{i=1}^{N_p-1} \text{nucleus}_i}{N_p - 1}. \quad (9)$$

Generally, the optima value will act as the nucleus. However, the nucleus of the last subpopulation is the mean of the optima values coming from the rest of the subpopulations in reciprocity operation. The purpose of the reciprocity mechanism is to ensure the diversity of the cloud particles. For example, circle represents the location of the global optimum and pentagram represents the location of the local optima in Figure 8. Figure 8(a) displays the position of nucleus before reciprocity operator. Figure 8(b) shows the individuals generated by cloud generator based on Figure 8(a). Figure 8(c) shows the position of nucleus with reciprocity operator based on Figure 8(a). Figure 8(d) shows the individuals generated by cloud generator based on Figure 8(c). By comparing Figure 8(b) with Figure 8(d), it can be seen that the distribution of the individuals in Figure 8(d) is more widely than those in Figure 8(b).

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

$$E_{n_{ij}} = \begin{cases} \frac{|\text{nucleus}_{ij}|}{\text{sf}}, & \text{if } |\text{nucleus}_{ij}| > 10^{-2}, \\ |\text{nucleus}_{ij}|, & \text{otherwise,} \end{cases} \quad (10)$$

$$H_{e_{ij}} = \frac{E_{n_{ij}}}{100}, \quad (11)$$

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

The pseudocode of CPEA is illustrated in Algorithms 1, 2, 3, and 4. N_p represents the number of the subpopulation. *Childnum* represents the number of individuals of every subpopulation. The search space is $[-r, r]$.

3. Experiments and Discussions

In this section, the performance of the proposed CPEA is tested with a comprehensive set of benchmark functions which include 40 different global optimization problems and CEC2013 benchmark problems. The definition of the benchmark functions and their global optimum(s) are listed in the Appendix. The benchmark functions are chosen from

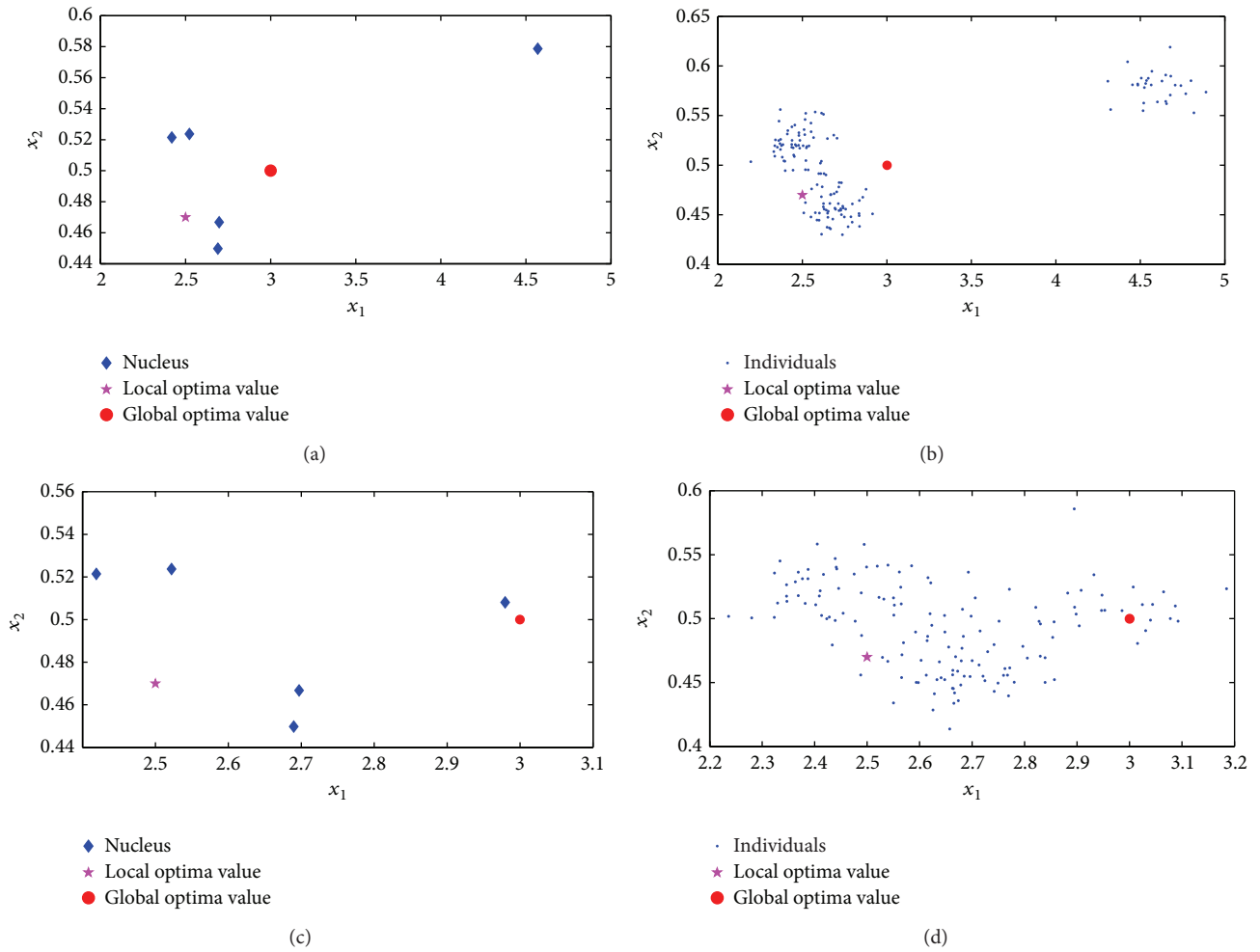


FIGURE 8: Reciprocity operator.

```

(1) if State == 0 then
(2)   Calculate the evolution potential of cloud particles according to (5)
(3)   Calculate the phase transformation driving force according to (6)
(4)   if  $PT \leq \delta$  then
(5)     % Phase Transformation from gaseous to liquid
(6)      $N_p = \lceil N_p/2 \rceil$ 
(7)      $Childnum = Childnum \times 2$ 
(8)     Select the nucleus of each sub-population
(9)      $\delta = \delta/10$ 
(10)    State = 1
(11)  else
(12)    % Condensation Operation
(13)    for  $i = 1: N_p$ 
(14)      Update  $E_{n_i}$  for each sub-population according to (7)
(15)      Update  $H_{e_i}$  for each sub-population according to (8)
(16)    endfor
(17)  endif
(18) endif
State = 0: Cloud Gaseous Phase, State = 1: Cloud Liquid Phase, State = 2: Cloud Solid Phase

```

ALGORITHM 1: Evolutionary algorithm of the cloud gaseous state.

```

(1) if State == 1 then
(2)   Calculate the evolution potential of cloud particles according to (5)
(3)   Calculate the phase transformation driving force according to (6)
(4)   for  $i = 1: N_p$ 
(5)     for  $j = 1: D$ 
(6)       if  $|nucleus_{ij}| > 1$ 
(7)          $E_{n_{ij}} = |nucleus_{ij}|/20$ 
(8)       elseif  $|nucleus_{ij}| > 0.1$ 
(9)          $E_{n_{ij}} = |nucleus_{ij}|/3$ 
(10)      else
(11)         $E_{n_{ij}} = |nucleus_{ij}|$ 
(12)      endif
(13)    endfor
(14)  endfor
(15)  for  $i = 1: N_p$ 
(16)    Update  $H_{e_i}$  for each sub-population according to (8)
(17)  endfor
(18)  if  $PT \leq \delta$  then
(19)    if rand >  $c/b$  then
(20)      Reciprocity Operation according to (9) % Reciprocity Operation
(21)    else
(22)      % Phase Transformation from liquid to solid
(23)       $N_p = \lceil N_p/2 \rceil$ 
(24)       $Childnum = Childnum \times 2$ 
(25)      Select the nucleus of each sub-population
(26)       $\delta = \delta/100$ 
(27)      State = 2
(28)    endif
(29)  endif
(30) endif

```

ALGORITHM 2: Evolutionary algorithm of the cloud liquid state.

```

(1) if State == 2 then
(2)   Calculate the evolution potential of cloud particles according to (5)
(3)   Calculate the phase transformation driving force according to (6)
(4)   % Solidification Operation
(5)   if  $PT \leq \delta$  then
(6)      $sf = sf \times 2$ 
(7)      $\delta = \delta/10$ 
(8)   endif
(9)   for  $i = 1: N_p$ 
(10)    for  $j = 1: D$ 
(11)      Update  $E_{n_{ij}}$  according to (10)
(12)      Update  $H_{e_{ij}}$  according to (11)
(13)    endfor
(14)  endfor
(15) endif

```

ALGORITHM 3: Evolutionary algorithm of the cloud solid state.

Ali et al. [25] and Rahnamayan et al. [26]. They are unimodal functions, multimodal functions, and rotate functions. 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.

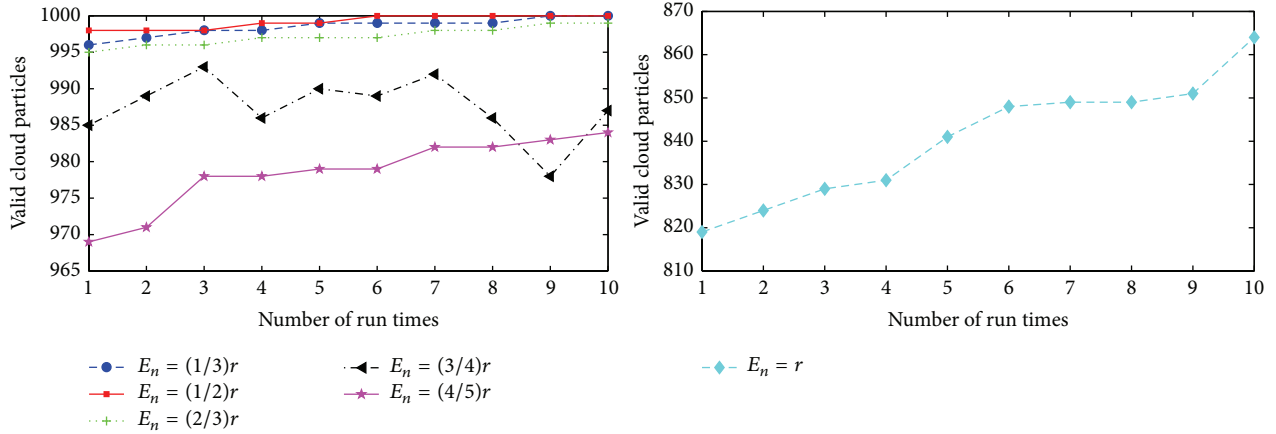
3.1. Parameter Study. In this section, the setting basis for initial values of some parameters involved in the cloud particles evolution algorithm model is described. Firstly, we hope that all the cloud particles which are generated by the cloud generator can fall into the search space as much as possible.

```

(1) Initialize  $D$  (number of dimensions),  $N_p = 5$ ;  $Childnum = 30$ ;  $E_n = 2/3r$ ;  $H_e = E_n/1000$ ;
(2) for  $i = 1: N_p$ 
(3)   for  $j = 1: D$ 
(4)     if  $j$  is even
(5)        $E_{x_{ij}} = r/2$ 
(6)     else
(7)        $E_{x_{ij}} = -r/2$ 
(8)     end if
(9)   end for
(10) end for
(11) Use cloud generator to produce cloud particles
(12) Evaluate Subpopulation
(13) Selectnucleus
(14) while requirements are not satisfied do
(15)   Use cloud generator to produce cloud particles
(16)   Evaluate Subpopulation
(17)   Selectnucleus
(18)   Evolutionary algorithm of the cloud gaseous state
(19)   Evolutionary algorithm of the cloud liquid state
(20)   Evolutionary algorithm of the cloud solid state
(21) end while

```

ALGORITHM 4: Cloud particles evolution algorithm: CPEA.

FIGURE 9: Valid cloud particles of different E_n .

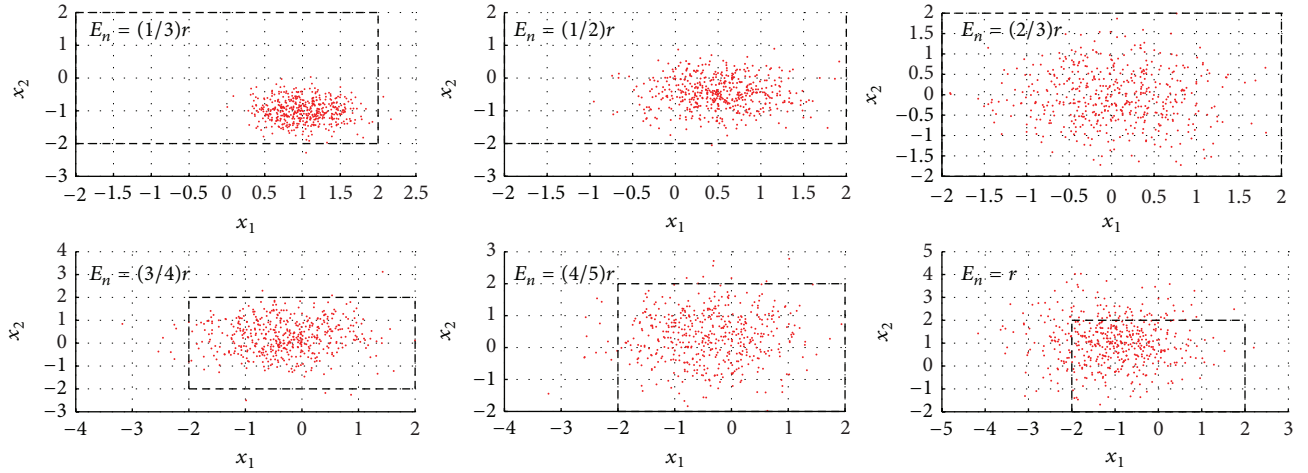
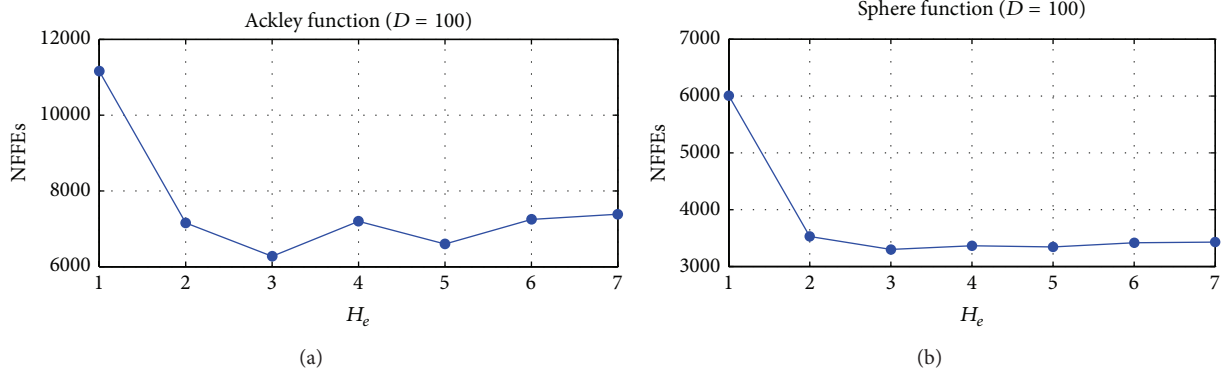
Secondly, the diversity of the population can be ensured effectively if the cloud particles are distributed in the search space uniformly. Therefore, some experiments are designed to analyze the feasibility of the selected scheme in order to illustrate the rationality of the selected parameters.

3.1.1. Definition of E_n . Let the search range S be $[-r, r]$ and r is the search radius. Set $r = 2$, $E_{x_1} = -2/3$, $E_{x_2} = 2/3$, $H_e = 0.001E_n$. Five hundred cloud particles are generated in every experiment. The experiment is independently simulated 30 times. The cloud particle is called a valid cloud particle if it falls into the search space. Figure 9 is a graph which shows the number of the valid cloud particles generated with different E_n , and Figure 10 is a graph which shows the distribution of valid cloud particles generated with different E_n .

It can be seen from Figure 9 that the number of the valid cloud particles reduces as the E_n increases. The average

number of valid cloud particles is basically the same when E_n is $(1/3)r$, $(1/2)r$, and $(2/3)r$, respectively. However, the number of valid cloud particles is relatively fewer when E_n is $(3/4)r$, $(4/5)r$, and r . It can be seen from Figure 10 that all the cloud particles fall into the research space when E_n is $(1/3)r$, $(1/2)r$, and $(2/3)r$, respectively. However, the cloud particles concentrate in a certain area of the search space when E_n is $(1/3)r$ or $(1/2)r$. By contrast, the cloud particles are widely distributed throughout all the search space $[-2, 2]$ when E_n is $(2/3)r$. Therefore, set $E_n = (2/3)r$. The dashed rectangle in Figure 10 represents the decision variables space.

3.1.2. Definition of H_e . Ackley Function (multimode) and Sphere Function (unimode) are selected to calculate the number of fitness evaluations (NFES) of the algorithm when the solution reaches f_{stop} with different H_e . Let the search space


 FIGURE 10: Distribution of cloud particles in different E_n .

 FIGURE 11: NFES of CPEA in different H_e .

be $[-32, 32]$ (Ackley) and $[-100, 100]$ (Sphere), $f_{\text{stop}} = 10^{-3}$. $H_{e1} = E_n$, $H_{e2} = 0.1E_n$, $H_{e3} = 0.01E_n$, $H_{e4} = 0.001E_n$, $H_{e5} = 0.0001E_n$, $H_{e6} = 0.00001E_n$, and $H_{e7} = 0.000001E_n$. The reported values are the average of the results for 50 independent runs. As shown in Figure 11, for Ackley Function, the number of fitness evaluations is least when H_e is equal to $0.001E_n$. For Sphere Function, the number of fitness evaluations is less when H_e is equal to $0.001E_n$. Therefore, H_e is equal to $0.001E_n$ in the cloud particles evolution algorithm.

3.1.3. Definition of E_x . Ackley Function (multimode) and Sphere Function (unimode) are selected to calculate the number of fitness evaluations (NFES) of the algorithm when the solution reaches f_{stop} with E_x being equal to r , $(1/2)r$, $(1/3)r$, $(1/4)r$, and $(1/5)r$, respectively. Let the search space be $[-r, r]$, $f_{\text{stop}} = 10^{-3}$. As shown in Figure 12, for Ackley Function, the number of fitness evaluations is least when E_x is equal to $(1/2)r$. For Sphere Function, the number of fitness evaluations is less when E_x is equal to $(1/2)r$ or $(1/3)r$. Therefore, E_x is equal to $(1/2)r$ in the cloud particles evolution algorithm.

3.1.4. Definition of c/b . Many things in nature have the mathematical proportion relationship between their parts. If the substance is divided into two parts, the ratio of the whole and the larger part is equal to the ratio of the larger part to the smaller part; this ratio is defined as golden ratio, expressed algebraically

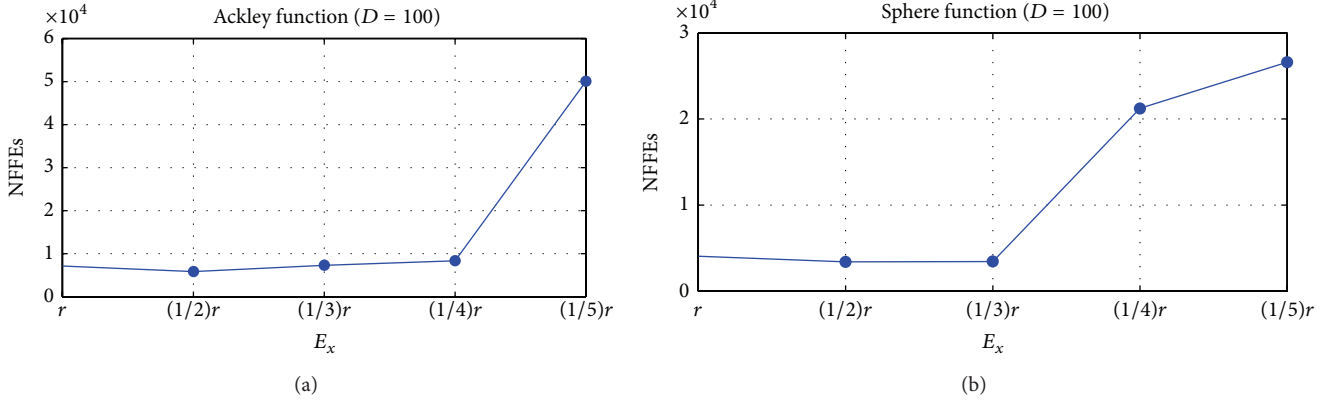
$$\frac{a+b}{a} = \frac{a}{b} \stackrel{\text{def}}{=} \varphi, \quad (12)$$

in which φ represents the golden ration. Its value is

$$\varphi = \frac{1 + \sqrt{5}}{2} = 1.6180339887, \quad (13)$$

$$\frac{1}{\varphi} = \varphi - 1 = 0.618 \dots$$

The golden section has a wide application not only in the fields of painting, sculpture, music, architecture, and management but also in nature. Studies have shown that plant leaves, branches, or petals are distributed in accordance with the golden ratio in order to survive in the wind, frost, rain, and snow. It is the evolutionary result or the best solution

FIGURE 12: NFES of CPEA in different E_x .

which adapts its own growth after hundreds of millions of years of long-term evolutionary process.

In this model, competition and reciprocity of the population depend on the fitness and are a dynamic relationship which constantly adjust and change. With competition and reciprocity mechanism, the convergence speed and the population diversity are improved. Premature convergence is avoided, and large search space and complex computational problems are solved. Indirect reciprocity is selected in this paper. Here, set $c/b = (1 - 0.618) \approx 0.38$.

3.2. Experiments Settings. For fair comparison, we set the parameters to be fixed. For CPEA, population size = 5, $childnum = 30$, $E_n = (2/3)r$, (the search space is $[-r, r]$), $H_e = E_n/1000$, $E_x = (1/2)r$, $cd = 1000$, $\delta = 10^{-2}$, and $sf = 1000$. For CMA-ES, the parameter values are chosen from [27]. The parameters of JADE come from [28]. For PSO, the parameter values are chosen from [29]. The parameters of DE are set to be $F = 0.3$ and $CR = 0.9$ used in [30]. For ABC, the number of colony size $NP = 20$ and the number of food sources $FoodNumber = NP/2$ and $limit = 100$. A food source will not be exploited anymore and is assumed to be abandoned when $limit$ is exceeded for the source. For CEBA, population size = 10, $childnum = 50$, $E_n = 1$, and $H_e = E_n/6$. The parameters of PSO-cf-Local and FDR-PSO come from [31, 32].

3.3. Comparison Strategies. Five comparison strategies come from the literature [33] and literature [26] to evaluate the algorithms. The comparison strategies are described as follows.

Successful Run. A run during which the algorithm achieves the fixed accuracy level within the max number of function evaluations for the particular dimension is shown as follows:

$$SR = \frac{\text{successful runs}}{\text{total runs}}. \quad (14)$$

Error. The error of a solution x is defined as $f(x) - f(x^*)$, where x^* is the global minimum of the function. The

minimum error is written when the max number of function evaluations is reached in 30 runs. Both the average and standard deviation of the error values are calculated.

Convergence Graphs. The convergence graphs show the median performance of the total runs with termination by either the max number of function evaluations or the termination error value.

Number of Function Evaluations (NFES). The number of function evaluations is recorded when the value-to-reach is reached. The average and standard deviation of the number of function evaluations are calculated.

Acceleration Rate (AR). The acceleration rate is defined as follows, based on the NFES for the two algorithms:

$$AR = \frac{NFES_{a1}}{NFES_{a2}}, \quad (15)$$

in which $AR > 1$ means Algorithm 2 ($a2$) is faster than Algorithm 1 ($a1$).

The average acceleration rate (AR_{ave}) and the average success rate (SR_{ave}) over n test functions are calculated as follows:

$$AR_{ave} = \frac{1}{n} \sum_{i=1}^n AR_i, \quad (16)$$

$$SR_{ave} = \frac{1}{n} \sum_{i=1}^n SR_i.$$

3.4. Comparison of CMA-ES, DE, RES, LOS, ABC, CEBA, and CPEA. This section compares CMA-ES, DE, RES, LOS, ABC, CEBA [34], and CPEA in terms of mean number of function evaluations for successful runs. Results of CMA-ES, DE, RES, and LOS are taken from literature [35]. Each run is stopped and regarded as successful if the function value is smaller than f_{stop} .

As shown in Table 1, CPEA can reach f_{stop} faster than other algorithms with respect to f_{07} and f_{17} . However, CPEA converges slower than CMA-ES with respect to f_{18} . When the

TABLE 1: Average number of function evaluations to reach f_stop of CPEA versus CMA-ES, DE, RES, LOS, ABC, and CEBA on f_{07} , f_{17} , f_{18} , and f_{33} function.

F	f_stop	init	D	CMA-ES	DE	RES	LOS	ABC	CEBA	CPEA
f_{07}	$1e-3$	$[-600, 600]^D$	20	3111	8691	—	—	12463	94412	2950
			30	4455	11410	$2e5$	—	29357	163545	4270
			100	12796	31796	—	—	34777	458102	7260
f_{17}	0.9	$[-5.12, 5.12]^D$ DE: $[-600, 600]^D$	20	68586	12971	—	$9.2e4$	11183	30436	2850
			30	147416	20150	$1.0e5$	$2.3e5$	26990	44138	6187
			100	1010989	73620	—	—	449404	60854	10360
f_{18}	$1e-3$	$[-30, 30]^D$	20	2667	—	—	$6.0e4$	10183	64400	3330
			30	3701	12481	$1.1e5$	$9.3e4$	17810	91749	4100
			100	11900	36801	—	—	59423	205470	12248
f_{33}	0.9	$[-5.12, 5.12]^D$	30	152000	$1.25e6$	—	—	—	50578	13100
			100	240899	5522	—	—	—	167324	14300

The bold entities indicate the best results obtained by the algorithm. D represents dimension.

dimension is 30, CPEA can reach f_stop relatively fast with respect to f_{33} . Nonetheless, CPEA converges slower while the dimension increases.

3.5. Comparison of CMA-ES, DE, PSO, ABC, CEBA, and CPEA. This section compares the error of a solution of CMA-ES, DE, PSO, ABC [36], CEBA, and CPEA under a given NFES. The best solution of every single function in different algorithms is highlighted with boldface.

As shown in Table 2, CPEA obtains better results in function f_{01} , f_{02} , f_{04} , f_{05} , f_{07} , f_{09} , f_{10} , f_{11} , and f_{12} while it obtains relatively bad results in other functions. The reason for this phenomenon is that the changing rule of E_n is set according to the priori information and background knowledge instead of adaptive mechanism in CPEA. Therefore, the results of some functions are relatively worse. Besides, E_n which can be optimized in order to get better results is an important controlling parameter in the algorithm.

3.6. Comparison of JADE, DE, PSO, ABC, CEBA, and CPEA. This section compares JADE, DE, PSO, ABC, CEBA, and CPEA in terms of SR and NFES for most functions at $D = 30$ except f_{03} ($D = 2$). The termination criterion is to find a value smaller than 10^{-8} . The best result of different algorithms for each function is highlighted in boldface.

As shown in Table 3, CPEA can reach the specified precision successfully except for function f_{14} which successful run is 96. In addition, CPEA can reach the specified precision faster than other algorithms in function f_{01} , f_{02} , f_{04} , f_{11} , f_{14} , and f_{17} . For other functions, CPEA spends relatively long time to reach the precision. The reason for this phenomenon is that, for some functions, the changing rule of E_n given in the algorithm makes CPEA to spend more time to get rid of the local optima and converge to the specified precision.

As shown in Table 4, the solution obtained by CPEA is closer to the best solution compared with other algorithms except for functions f_{13} and f_{20} . The reason is that the phase transformation mechanism introduced in the algorithm can improve the convergence of the population greatly. Besides,

the reciprocity mechanism of the algorithm can improve the adaptation and the diversity of the population in case the algorithm falls into the local optimum. Therefore, the algorithm can obtain better results in these functions.

As shown in Table 5, CPEA can find the optimal solution for the test functions. However, the stability of CPEA is worse than JADE and DE (f_{25} , f_{26} , f_{27} , f_{28} and f_{31}). This result shows that the setting of H_e needs to be further improved.

3.7. Comparison of CMAES, JADE, DE, PSO, ABC, CEBA, and CPEA. The convergence curves of different algorithms on some selected functions are plotted in Figures 13 and 14. The solutions obtained by CPEA are better than other algorithms with respect to functions f_{01} , f_{11} , f_{16} , f_{32} , and f_{33} . However, the solution obtained by CPEA is worse than CMA-ES with respect to function f_{04} . The solution obtained by CPEA is worse than JADE for high dimensional problems such as function f_{05} and function f_{07} . The reason is that the setting of initial population of CPEA is insufficient for high dimensional problems.

3.8. Comparison of DE, ODE, RDE, PSO, ABC, CEBA, and CPEA. The algorithms DE, ODE, RDE, PSO, ABC, CEBA, and CPEA are compared in terms of convergence speed. The termination criterion is to find error values smaller than 10^{-8} . The results of solving the benchmark functions are given in Tables 6 and 7. Data of all functions in DE, ODE, and RDE are from literature [26]. The best result of NEFS and the SR for each function are highlighted in boldface. The average success rates and the average acceleration rate on test functions are shown in the last row of the tables.

It can be seen from Tables 6 and 7 that CPEA outperforms DE, ODE, RDE, PSO, ABC, and CEBA on 38 of the functions. CPEA performs better than other algorithms on 23 functions. In Table 6, the average AR is 0.98, 0.86, 4.01, 6.36, 1.31, and 18.58 for ODE, RDE, PSO, ABC, CEBA, and CPEA, respectively. In Table 7, the average AR is 1.27, 0.88, 1.12, 2.31, 0.46, and 4.56 for ODE, RDE, PSO, ABC, CEBA, and CPEA, respectively. In Table 6, the average SR is 0.96, 0.98, 0.96, 0.67,

TABLE 2: Comparison on the ERROR values of CPEA versus CMA-ES, DE, PSO, ABC, and CEBA on function $f_{01} \sim f_{13}$.

F	NFES	D	CMA-ES mean/SD	DE mean/SD	PSO mean/SD	ABC mean/SD	CEBA mean/SD	CPEA mean/SD
f_{01}	30 k	30	5.87e - 29/ 1.56e - 29	3.20e - 07/ 2.51e - 07	5.44e - 12/ 1.56e - 11	9.71e - 16/ 2.26e - 16	1.74e - 02/ 1.55e - 02	3.19e - 33/ 6.21e - 33
	30 k	50	1.02e - 28/ 1.87e - 29	2.95e - 03/ 2.88e - 03	3.93e - 03/ 6.62e - 02	2.35e - 08/ 2.47e - 08	4.07e - 02/ 2.11e - 02	1.09e - 48/ 1.41e - 48
f_{02}	10 k	30	1.50e - 05/ 3.66e - 05	2.30e + 00/ 1.78e + 00	6.24e - 06/ 8.74e - 05	0.00e + 00/ 0.00e + 00	6.16e + 00/ 3.22e + 00	0.00e + 00/ 0.00e + 00
	50 k	50	0.00e + 00/ 0.00e + 00	1.65e + 00/ 2.05e + 00	3.87e + 00/ 2.65e + 00	0.00e + 00/ 0.00e + 00	6.08e + 00/ 5.34e + 00	0.00e + 00/ 0.00e + 00
f_{03}	10 k	2	2.65e - 30/ 8.08e - 31	0.00e + 00/ 0.00e + 00	5.01e - 14/ 8.52e - 14	4.83e - 04/ 1.40e - 03	5.83e - 04/ 8.32e - 04	1.52e - 13/ 2.44e - 13
f_{04}	100 k	30	8.28e - 27/ 1.69e - 27	2.47e - 03/ 2.50e - 03	1.69e - 02/ 2.57e - 02	1.36e + 00/ 1.93e + 00	1.59e + 00/ 7.93e + 00	3.31e - 14/ 1.42e - 13
	100 k	50	2.29e - 06/ 2.22e - 06	7.37e - 01/ 1.93e - 01	5.01e + 00/ 3.88e + 00	1.19e + 02/ 4.54e + 01	3.13e + 02/ 3.94e + 02	3.98e - 07/ 8.79e - 07
f_{05}	30 k	30	8.19e - 09/ 5.08e - 09	1.27e - 12/ 3.94e - 12	8.66e - 34/ 2.88e - 33	8.00e - 13/ 2.24e - 12	2.54e - 09/ 4.13e - 09	2.11e - 37/ 4.20e - 36
	50 k	50	2.68e - 08/ 1.67e - 09	1.93e - 13/ 2.01e - 13	4.37e - 54/ 7.58e - 54	9.54e - 12/ 1.55e - 11	1.34e - 15/ 2.01e - 15	1.07e - 23/ 1.85e - 23
f_{06}	30 k	2	0.00e + 00/ 0.00e + 00	0.00e + 00/ 0.00e + 00	0.00e + 00/ 0.00e + 00	3.60e - 09/ 1.05e - 08	1.1e - 03/ 8.2e - 04	7.71e - 09/ 4.52e - 09
f_{07}	30 k	30	1.88e - 12/ 1.40e - 12	3.08e - 06/ 5.41e - 06	1.64e - 09/ 2.85e - 08	1.54e - 09/ 2.78e - 08	2.07e - 06/ 5.57e - 05	5.32e - 16/ 7.59e - 16
	50 k	50	0.00e + 00/ 0.00e + 00	3.28e - 05/ 5.69e - 05	1.79e - 02/ 2.49e - 02	1.27e - 09/ 2.21e - 09	9.49e - 03/ 4.23e - 03	0.00e + 00/ 0.00e + 00
f_{08}	30 k	3	0.00e + 00/ 0.00e + 00	0.00e + 00/ 0.00e + 00	0.00e + 00/ 0.00e + 00	0.00e + 00/ 0.00e + 00	2.80e - 03/ 2.13e - 03	3.85e - 06/ 6.21e - 06
f_{09}	30 k	2	0.00e + 00/ 0.00e + 00	0.00e + 00/ 0.00e + 00	0.00e + 00/ 0.00e + 00	0.00e + 00/ 0.00e + 00	0.00e + 00/ 0.00e + 00	0.00e + 00/ 0.00e + 00
f_{10}	5 k	2	6.99e - 47/ 2.21e - 46	1.28e - 26/ 3.9e - 26	6.59e - 07/ 7.71e - 07	2.32e - 04/ 2.60e - 04	8.24e - 08/ 1.37e - 07	8.29e - 59/ 3.42e - 58
f_{11}	30 k	30	1.23e - 04/ 1.82e - 04	3.74e - 01/ 2.91e - 01	4.78e - 01/ 1.94e - 01	2.39e + 00/ 3.27e + 00	1.42e + 00/ 8.13e + 00	5.70e - 40/ 6.61e - 39
	50 k	50	5.58e - 01/ 1.55e - 01	3.36e + 00/ 1.55e + 00	3.15e + 01/ 1.11e + 01	5.71e + 01/ 6.98e + 00	2.05e + 01/ 9.40e + 00	2.33e - 05/ 3.58e - 05
f_{12}	30 k	30	3.19e - 14/ 1.45e - 14	1.20e - 08/ 2.24e - 08	2.15e - 05/ 5.18e - 04	3.56e - 10/ 1.22e - 10	1.93e - 07/ 6.45e - 08	2.33e - 23/ 2.50e - 22
	50 k	50	1.41e - 14/ 2.43e - 14	6.77e - 05/ 1.11e - 05	6.08e - 03/ 3.35e - 03	6.37e - 10/ 6.83e - 10	9.76e - 03/ 3.08e - 03	1.90e - 72/ 3.29e - 72
f_{13}	30 k	30	3.94e - 11/ 1.15e - 10	1.34e - 31/ 3.21e - 31	1.34e - 31/ 0.00e + 00	6.57e - 04/ 3.77e - 03	2.09e - 02/ 7.34e - 02	1.20e - 12/ 2.82e - 12
	50 k	50	3.08e - 03/ 3.80e - 03	1.34e - 31/ 0.00e + 00	1.34e - 31/ 0.00e + 00	1.41e - 02/ 9.29e - 03	4.22e - 02/ 2.53e - 02	2.36e - 12/ 1.04e - 12

“SD” stands for standard deviation.

0.79, 0.89, and 0.99 for DE, ODE, RDE, PSO, ABC, CBEA, and CPEA, respectively. In Table 7, the average SR is 0.84, 0.84, 0.84, 0.73, 0.89, 0.77, and 0.98 for DE, ODE, RDE, PSO, ABC, CBEA, and CPEA, respectively. These results clearly demonstrate that CPEA has better performance.

3.9. Test on the CEC2013 Benchmark Problems. To further verify the performance of CPEA, a set of recently proposed CEC2013 rotated benchmark problems are used. In the experiment, t -test [37] has been carried out to show the differences

between CPEA and the other 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. Table 8 gives the information about the average error, standard deviation, and t -value of 30 runs of 7 algorithms over 300,000 FEs on 20 test functions with $D = 10$. The best results among the algorithms are shown in bold.

Table 8 consists of unimodal problems ($F_{01} \sim F_{05}$) and basic multimodal problems ($F_{06} \sim F_{20}$). From the results of

TABLE 3: Comparison on the SR and the NFES of CPEA versus JADE, DE, PSO, ABC, and CEBA for each function.

F		f_{01} mean/ SD	f_{02} mean/ SD	f_{03} mean/ SD	f_{04} mean/ SD	f_{07} mean/ SD	f_{11} mean/ SD	f_{14} mean/ SD	f_{16} mean/ SD	f_{17} mean/ SD
JADE	SR	1	1	1	1	1	0.98	0.94	1	1
	NFES	2.8e + 4/ 6.0e + 2	1.0e + 4/ 3.5e + 2	5.7e + 3/ 6.9e + 2	6.9e + 4/ 3.0e + 3	3.3e + 3/ 2.6e + 2	6.6e + 4/ 4.2e + 3	4.0e + 4/ 2.1e + 3	1.3e + 3/ 3.8e + 2	1.3e + 5/ 1.8e + 3
DE	SR	1	1	1	1	0.96	1	0.94	1	0
	NFES	3.3e + 4/ 6.6e + 2	1.1e + 4/ 2.3e + 3	1.6e + 3/ 1.1e + 2	1.9e + 5/ 1.2e + 4	3.4e + 4/ 1.6e + 3	1.4e + 5/ 1.0e + 4	2.6e + 4/ 1.6e + 4	2.1e + 4/ 8.1e + 3	—
PSO	SR	1	1	1	0	0.86	0	0	1	0
	NFES	2.9e + 4/ 7.5e + 2	1.6e + 4/ 1.4e + 3	7.7e + 3/ 9.6e + 2	—	2.8e + 4/ 9.8e + 2	—	—	1.8e + 4/ 7.2e + 2	—
ABC	SR	1	1	1	0	0.96	0	0	1	1
	NFES	1.7e + 4/ 1.0e + 3	6.3e + 3/ 1.5e + 3	1.6e + 4/ 2.6e + 3	—	1.9e + 4/ 2.3e + 3	—	—	6.5e + 3/ 3.9e + 2	3.9e + 4/ 1.5e + 4
CEBA	SR	1	1	1	0	1	1	0.96	1	1
	NFES	1.3e + 5/ 5.8e + 3	5.9e + 4/ 4.6e + 3	4.2e + 4/ 6.1e + 3	—	2.2e + 5/ 1.2e + 4	1.3e + 5/ 5.8e + 3	2.9e + 5/ 1.4e + 4	5.4e + 4/ 3.8e + 3	1.1e + 5/ 7.3e + 3
CPEA	SR	1	1	1	1	1	1	0.96	1	1
	NFES	4.3e + 3/ 5.1e + 2	4.5e + 3/ 2.7e + 2	5.7e + 3/ 3.0e + 3	6.1e + 4/ 2.6e + 3	7.7e + 3/ 3.4e + 2	2.3e + 4/ 3.4e + 3	3.4e + 4/ 1.6e + 3	2.4e + 3/ 4.3e + 2	1.0e + 4/ 1.3e + 3

TABLE 4: Comparison on the ERROR values of CPEA versus JADE, DE, PSO, and ABC at $D = 50$ except f_{21} ($D = 2$) and f_{22} ($D = 2$) for each function.

F	NFES	JADE mean/SD	DE mean/SD	PSO mean/SD	ABC mean/SD	CPEA mean/SD
f_{13}	40 k	1.34e - 31/0.00e + 00	1.34e - 31/0.00e + 00	1.35e - 31/0.00e + 00	3.52e - 03/6.33e - 02	3.64e - 12/4.68e - 12
f_{15}	40 k	9.18e - 11/7.04e - 11	2.94e - 05/3.66e - 05	1.81e - 09/4.18e - 08	4.11e - 14/2.62e - 14	4.59e - 53/1.36e - 52
f_{16}	40 k	1.15e - 20/1.42e - 20	6.62e - 08/1.24e - 07	6.91e - 19/7.55e - 19	3.41e - 15/5.23e - 15	1.77e - 76/0.00e + 00
f_{17}	200 k	6.71e - 07/1.32e - 07	2.58e - 01/6.15e - 01	6.73e + 00/2.06e + 00	6.07e - 09/1.85e - 08	0.00e + 00/0.00e + 00
f_{18}	50 k	1.16e - 07/4.27e - 08	3.89e - 04/2.10e - 04	1.70e - 09/3.13e - 08	2.93e - 09/2.34e - 08	8.88e - 16/0.00e + 00
f_{19}	40 k	1.14e - 02/1.68e - 02	9.90e - 03/6.66e - 03	9.00e - 06/1.62e - 05	2.22e - 05/1.95e - 04	1.64e - 64/5.13e - 64
f_{20}	40 k	0.00e + 00/0.00e + 00	0.00e + 00/0.00e + 00	1.40e - 10/2.81e - 10	4.38e - 03/3.33e - 03	1.21e - 10/3.11e - 09
f_{21}	30 k	0.00e + 00/0.00e + 00	0.00e + 00/0.00e + 00	0.00e + 00/0.00e + 00	1.06e - 17/3.34e - 17	0.00e + 00/0.00e + 00
f_{22}	30 k	1.28e - 45/1.86e - 45	4.99e - 16/0.00e + 00	2.63e - 51/6.82e - 51	6.62e - 18/5.84e - 18	0.00e + 00/0.00e + 00
f_{23}	30 k	0.00e + 00/0.00e + 00	0.00e + 00/0.00e + 00	0.00e + 00/0.00e + 00	0.00e + 00/4.24e - 04	0.00e + 00/0.00e + 00

TABLE 5: Comparison on the optimum solution of CPEA versus JADE, DE, PSO, and ABC for each function.

F	NFES	JADE mean/SD	DE mean/SD	PSO mean/SD	ABC mean/SD	CPEA mean/SD
f_{08}	40 k	-3.86278/0.00e + 00	-3.86278/0.00e + 00	-3.86278/6.20e - 17	-3.86278/0.00e + 00	-3.86278/0.00e + 00
f_{24}	40 k	-3.31044/3.60e - 02	-3.24608/5.70e - 02	-3.25800/5.90e - 02	-3.33539/0.00e + 00	-3.33539/3.03e - 06
f_{25}	40 k	-10.1532/4.00e - 14	-10.1532/4.20e - 16	-5.79196/3.30e + 00	-10.1532/1.25e - 13	-10.1532/1.58e - 09
f_{26}	40 k	-10.4029/9.40e - 16	-10.4029/2.50e - 16	-7.14128/3.50e + 00	-10.4029/1.77e - 15	-10.4029/1.94e - 10
f_{27}	40 k	-10.5364/8.10e - 12	-10.5364/7.10e - 16	-7.02213/3.60e + 00	-10.5364/4.03e - 05	-10.5364/2.02e - 09
f_{28}	40 k	3.00000/1.10e - 15	3.00000/6.4e - 16	3.00000/1.30e - 15	3.00000/6.17e - 05	3.00000/2.01e - 15
f_{29}	40 k	2.55e - 20/4.79e - 20	0.00e + 00/0.00e + 00	2.10e - 03/4.28e - 03	1.14e - 18/1.62e - 18	0.00e + 00/0.00e + 00
f_{30}	40 k	-2.3458/0.00e + 00	-2.3458/0.00e + 00	-2.3458/0.00e + 00	-2.3458/0.00e + 00	-2.3458/0.00e + 00
f_{31}	40 k	4.78e - 31/1.07e - 30	3.59e - 55/1.13e - 54	1.22e - 36/3.85e - 36	1.40e - 11/1.38e - 11	8.69e - 25/1.14e - 24
f_{32}	40 k	1.49e - 02/4.23e - 03	3.46e - 02/5.14e - 03	6.58e - 02/1.22e - 02	2.95e - 01/1.13e - 01	5.70e - 03/3.58e - 03

Data of functions f_{08} , f_{24} , f_{25} , f_{26} , f_{27} , and f_{28} in JADE, DE, and PSO are from literature [28].

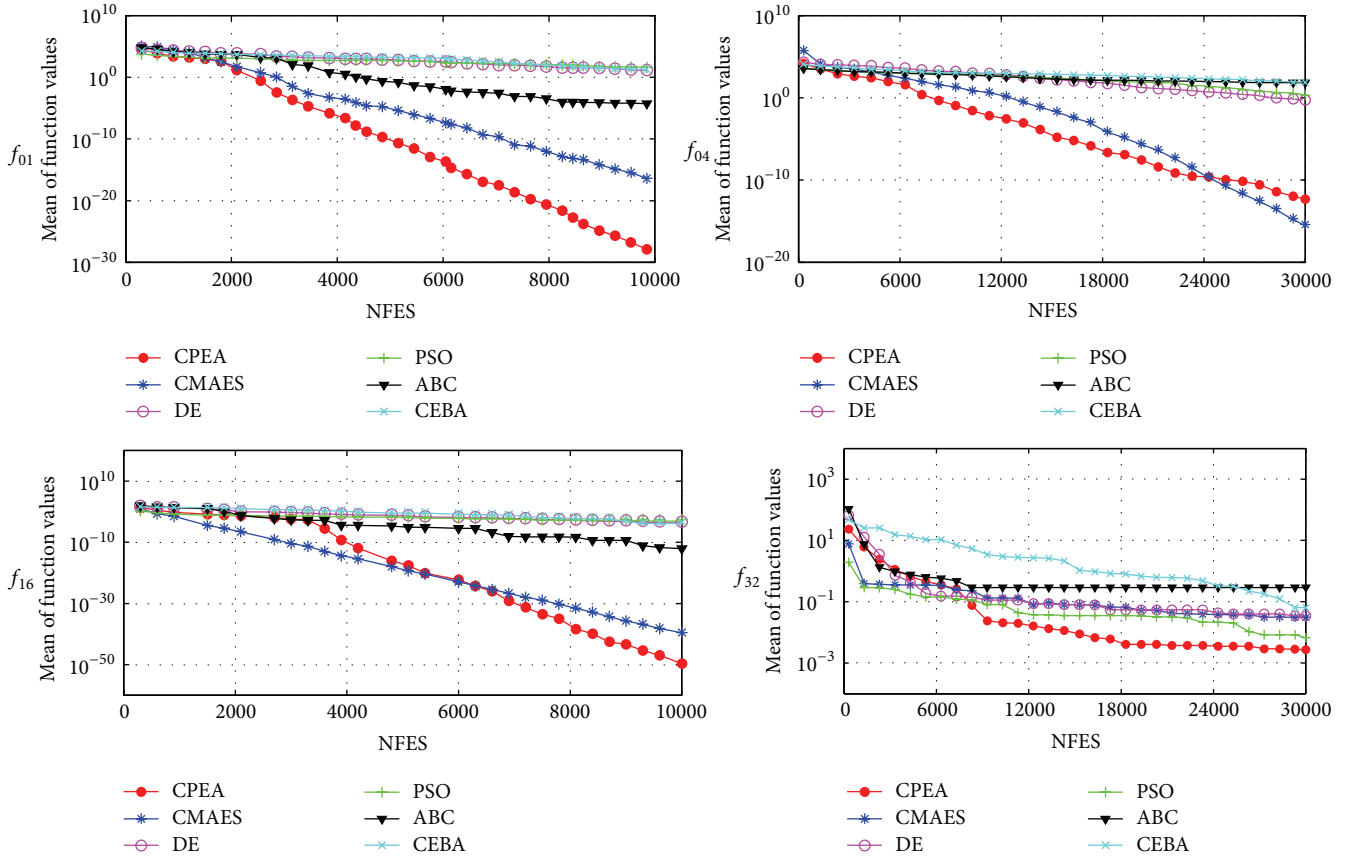


FIGURE 13: Convergence graph for test functions f_{01} , f_{04} , f_{16} , and f_{32} ($D = 30$).

Table 8, we can see that PSO, ABC, PSO-FDR, PSO-cf-Local, and CPEA work well for F_{01} , where PSO-FDR, PSO-cf-Local, and CPEA can always achieve the optimal solution in each run. CPEA outperforms other algorithms on F_{01} , F_{02} , F_{03} , F_{07} , F_{08} , F_{10} , F_{12} , F_{15} , F_{16} , and F_{18} ; the cause of the outstanding performance may be because of the phase transformation mechanism, which leads to faster convergence in the late stage of evolution.

4. Conclusions

This paper presents a new optimization method called cloud particles evolution algorithm (CPEA). The fundamental concepts and ideas come from the formation of the cloud and the phase transformation of the substances in nature. In this paper, CPEA solves 40 benchmark optimization problems which have difficulties in discontinuity, nonconvexity, multimode, and high-dimensionality. The evolutionary process of the cloud particles is divided into three states: the cloud gaseous state, the cloud liquid state, and the cloud solid state. The algorithm realizes the change of the state of the cloud with the phase transformation driving force according to the evolutionary process. The algorithm reduces the population number appropriately and increases the individuals of subpopulations in each phase transformation. Therefore, CPEA can improve the whole exploitation ability of the population greatly and enhance the convergence. There is a

larger adjustment for E_n to improve the exploration ability of the population in melting or gasification operation. When the algorithm falls into the local optimum, it improves the diversity of the population through the indirect reciprocity mechanism to solve this problem.

As we can see from experiments, it is helpful to improve the performance of the algorithm with appropriate parameter settings and change strategy in the evolutionary process. As we all know, the benefit is proportional to the risk for solving the practical engineering application problems. Generally, the better the benefit is, the higher the risk is. We always hope that the benefit is increased and the risk is reduced or controlled at the same time. In general, the adaptive system adapts to the dynamic characteristics change of the object and disturbance by correcting its own characteristics. Therefore, the benefit is increased. However, the adaptive mechanism is achieved at the expense of computational efficiency. Consequently, the settings of parameter should be based on the actual situation. The settings of parameters should be based on the prior information and background knowledge in some certain situations, or based on adaptive mechanism for solving different problems in other situations. The adaptive mechanism will not be selected if the static settings can ensure to generate more benefit and control the risk. Inversely, if the adaptive mechanism can only ensure to generate more benefit while it cannot control the risk, we will not choose it. The adaptive mechanism should be selected only if it can not only

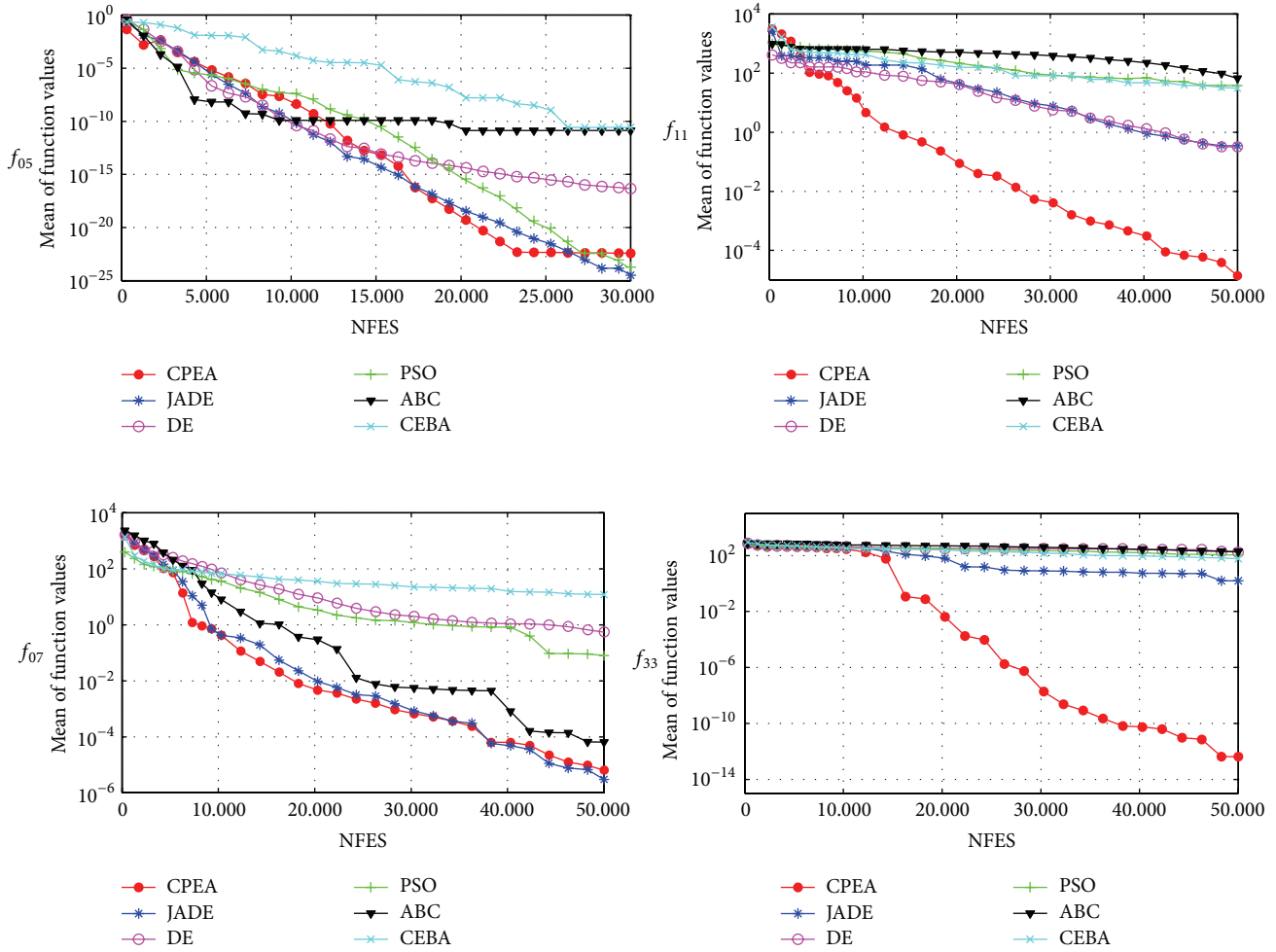


FIGURE 14: Convergence graph for test functions f_{05} , f_{11} , f_{33} ($D = 50$), and f_{07} ($D = 100$).

increase the benefit but also reduce the risk. In CPEA, E_n and H_e are statically set according to the priori information and background knowledge. As we can see from experiments, the results from some functions are better while the results from the others are relatively worse among the 40 tested functions.

As previously mentioned, parameter setting is an important focus. Then, the next step is the adaptive settings of E_n and H_e . Through adaptive behavior, these parameters can automatically modify the calculation direction according to the current situation while the calculation is running. Moreover, possible directions for future work include improving the computational efficiency and the quality of solutions of CPEA. Finally, CPEA will be applied to solve large scale optimization problems and engineering design problems.

Appendix

Test Functions

f_{01} Sphere model, $f_{01}(x) = \sum_{i=1}^n x_i^2$, $[-100, 100]$, $\min(f_{01}) = 0$.

f_{02} Step function $f_{02}(x) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$, $[-100, 100]$, $\min(f_{02}) = 0$.

f_{03} Beale function $f_{03}(x) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.625 - x_1(1 - x_2^3)]^2$, $[-4.5, 4.5]$, $\min(f_{03}) = 0$.

f_{04} Schwefel's problem 1.2 $f_{04}(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$, $[-65, 65]$, $\min(f_{04}) = 0$.

f_{05} Sum of different power $f_{05}(x) = \sum_{i=1}^n |x_i|^{i+1}$, $[-1, 1]$, $\min(f_{05}) = f(0, \dots, 0) = 0$.

f_{06} Easom function $f_{06}(x) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$, $[-100, 100]$, $\min(f_{06}) = -1$.

f_{07} Griewangk's function $f_{07}(x) = \sum_{i=1}^n (x_i^2/4000) - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$, $[-600, 600]$, $\min(f_{07}) = 0$.

f_{08} Hartmann function $f_{08}(x) = -\sum_{i=1}^4 a_i \exp(-\sum_{j=1}^3 A_{ij}(x_j - P_{ij})^2)$, $[0, 1]$, $\min(f_{08}) = -3.86278214782076$. Consider

$$a = [1 \quad 1.2 \quad 3 \quad 3.2],$$

$$A = \begin{bmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix},$$

TABLE 6: Comparison of DE, ODE, RDE, PSO, ABC, CEBA, and CPEA. The best result for each case is highlighted in boldface.

F	D	DE		ODE		AR		RDE		AR		PSO		AR		ABC		AR		CEBA		AR		CPEA			
		NFES	SR	NFES	SR	ODE	AR	RDE	SR	NFES	SR	AR	RDE	SR	NFES	SR	AR	ABC	SR	NFES	SR	AR	CEBA	SR	NFES	SR	AR
f_{01}	30	87748	1	47716	1	1.83	115096	1	0.76	4.32	18370	1	4.77	133381	1	0.66	5266	1	16.66	1	16.66	1	16.66	1	16.66	1	16.66
f_{02}	30	41588	1	23124	1	1.80	54316	1	0.77	3.60	6247	1	6.66	58416	1	0.71	2966	1	14.02	1	14.02	1	2966	1	2966	1	14.02
f_{03}	2	4324	1	4776	1	0.90	4904	1	0.88	0.71	18050	1	0.24	38278	1	0.11	4350	1	0.99	1	0.99	1	4350	1	4350	1	0.99
f_{04}	20	177880	1	168680	1	1.05	231152	1	0.77	—	—	0	—	—	0	—	—	0	16.07	1	16.07	1	11068	1	11068	1	16.07
f_{05}	30	25140	1	8328	1	3.01	29096	1	0.86	4.25	6134	1	4.10	18037	1	1.39	1700	1	14.79	1	14.79	1	1700	1	1700	1	14.79
f_{06}	2	8016	1	6608	1	1.21	8248	1	0.97	1.12	25407	1	0.32	66319	1	0.12	5033	1	1.59	1	1.59	1	5033	1	5033	1	1.59
f_{07}	30	113428	1	69342	1	1.64	149744	1	0.76	5.48	20930	0.96	5.42	209278	1	0.54	9941	1	11.41	1	11.41	1	9941	1	9941	1	11.41
f_{08}	3	3376	1	3580	1	0.94	3652	1	0.92	0.83	1492	1	2.26	206859	1	0.02	31874	1	0.11	1	0.11	1	31874	1	31874	1	0.11
f_{09}	2	5352	1	4468	1	1.20	6292	1	0.85	1.89	1077	1	4.97	9200	1	0.58	3856	1	1.39	1	1.39	1	9200	1	9200	1	1.39
f_{10}	2	3608	1	3288	1	1.09	3924	1	0.92	0.66	10891	1	0.33	6633	1	0.54	1920	1	1.88	1	1.88	1	6633	1	6633	1	1.88
f_{11}	30	385192	1	369104	1	1.04	498200	1	0.77	—	—	0	—	—	0	—	—	0	30.42	1	30.42	1	142564	1	142564	1	30.42
f_{12}	30	187300	1	155636	1	1.20	244396	1	0.76	—	—	0	—	—	0	—	—	0	18.39	1	18.39	1	157139	1	157139	1	18.39
f_{13}	30	101460	1	70408	1	1.44	138308	1	0.73	15.58	—	0	7.23	—	0	—	—	0	23.60	1	23.60	1	—	0	—	0	23.60
f_{14}	30	570290	0.28	72250	0.88	7.89	285108	1	2.00	—	—	0	—	—	0	—	—	0	18.33	1	18.33	1	292946	1	292946	1	18.33
f_{15}	30	96488	1	53304	1	1.81	126780	1	0.76	5.14	15150	1	6.37	103475	1	0.93	6466	1	14.92	1	14.92	1	103475	1	103475	1	14.92
f_{16}	2	1016	1	996	1	1.02	1084	1	0.94	2.37	301	1	3.38	3292	1	0.31	610	1	1.67	1	1.67	1	3292	1	3292	1	1.67
f_{17}	10	328844	1	70389	0.76	4.67	501875	0.96	0.66	—	—	0	41.44	48221	1	6.82	3725	1	88.28	1	88.28	1	48221	1	48221	1	88.28
f_{18}	30	169152	1	98296	1	1.72	222784	1	0.76	6.21	32398	1	5.22	184041	1	0.92	13400	1	12.62	1	12.62	1	184041	1	184041	1	12.62
f_{19}	30	411164	1	337532	1	1.22	927230	0.24	0.44	—	—	0	2.62	148014	1	2.78	6233	1	65.97	1	65.97	1	148014	1	148014	1	65.97
Ave			0.96		0.98			0.96	0.86	4.01	0.79	0.67	6.36	0.89	1.31			0.99						0.99			18.58

TABLE 7: Comparison of DE, ODE, RDE, RDE, PSO, ABC, CEBA, and CPEA. The best result for each case is highlighted in boldface.

F	D	DE		ODE		AR		RDE		AR		PSO		ABC		CEBA		AR		CPEA					
		NFES	SR	NFES	SR	ODE	AR	RDE	SR	NFES	SR	AR	PSO	NFES	SR	ABC	SR	NFES	SR	AR	CEBA	NFES	SR	CPEA	AR
f_{20}	5	2163	0.88	2024	1	1.07	0.96	0.74	0.96	1	0.18	1	11960	1	—	0	58252	1	0.04	4216	0.92	0.51	—	—	—
f_{21}	2	7976	1	5092	1	1.56	1	0.96	1	0.98	0.95	8372	0.98	15944	1	0.50	11129	1	0.72	1025	1	7.78	—	—	—
f_{22}	2	3780	1	3396	1	1.11	1	0.98	1	0.98	0.66	5755	1	2050	1	1.84	6635	1	0.57	855	1	4.42	—	—	—
f_{23}	10	19528	1	15704	1	1.24	1	0.84	1	0.84	2.10	9290	1	5845	1	3.34	17907	1	1.09	1130	1	17.28	—	—	—
f_{24}	6	—	0	—	0	—	0	—	0	—	—	7966	0.94	3018	1	—	—	0	—	42716	0.98	—	—	—	—
f_{25}	4	—	0	—	0	—	0	—	0	—	—	—	0	4280	1	—	—	0	—	5471	0.94	—	—	—	—
f_{26}	4	—	0	—	0	—	0	—	0	—	—	—	0	8383	1	—	—	0	—	5391	0.96	—	—	—	—
f_{27}	4	4576	1	4500	1	1.02	1	0.77	1	0.77	—	—	0	4850	1	0.94	—	0	—	5512	0.92	0.83	—	—	—
f_{28}	2	4780	1	4684	1	1.02	1	0.97	1	0.97	0.71	6770	1	6228	1	0.77	14248	1	0.34	4495	1	1.06	—	—	—
f_{29}	2	8412	1	5772	1	1.46	1	0.77	1	0.77	—	—	0	1914	1	4.40	37765	1	0.22	4950	1	1.70	—	—	—
f_{30}	2	3256	1	3120	1	1.04	1	0.98	1	0.98	0.99	3275	1	1428	1	2.28	114261	1	0.03	3025	1	1.08	—	—	—
f_{31}	4	1108	1	1232	1	0.90	1	0.80	1	0.80	1.12	990	1	9944	1	0.11	131217	1	0.01	2662	1	0.42	—	—	—
f_{34}	2	5232	1	5956	0.92	0.88	1	0.91	1	0.91	1.50	3495	1	1852	1	2.83	9926	1	0.53	4370	1	1.20	—	—	—
f_{35}	5	38532	1	16340	1	2.36	1	0.84	1	0.84	3.24	11895	1	25303	1	1.52	35292	1	1.09	2095	1	18.39	—	—	—
f_{36}	2	7888	1	4272	1	1.85	1	0.86	1	0.86	0.72	10910	1	8359	1	0.94	13529	1	0.58	1340	1	5.89	—	—	—
f_{37}	2	5284	1	4728	1	1.12	1	0.97	1	0.97	0.69	7685	1	1038	1	5.09	9056	1	0.58	1022	1	5.17	—	—	—
f_{38}	2	5280	1	4804	1	1.10	1	0.95	1	0.95	0.69	7680	1	1598	1	3.30	8905	1	0.59	990	1	5.33	—	—	—
f_{39}	10	37824	1	24206	1	1.56	1	0.80	1	0.80	—	—	0	—	0	—	153161	1	0.25	38420	0.94	0.98	—	—	—
f_{40}	2	2976	1	2872	1	1.03	1	0.93	1	0.93	1.07	2790	1	666	1	4.47	9032	1	0.33	3085	1	0.96	—	—	—
Ave			0.84		0.84	1.27	0.88	0.88	0.84	0.88	1.12	0.73	0.89	0.89	2.31	0.77	0.77	0.46	0.46	0.98	0.98	4.56	—	—	—

TABLE 8: Results for CEC2013 benchmark functions over 30 independent runs.

Function	Result	DE	PSO	ABC	CEBA	PSO-FDR	PSO-cf-Local	CPEA
F_{01}	F_{mean} SD	$1.64e - 02(+)$ / $3.98e - 02$	$1.51e - 14(\approx)$ / $5.76e - 14$	$1.51e - 14(\approx)$ / $5.76e - 14$	$4.99e + 02(+)$ / $1.46e + 02$	$0.00e + 00(\approx)$ / $0.00e + 00$	$0.00e + 00(\approx)$ / $0.00e + 00$	$0.00e+00/$ $0.00e + 00$
F_{02}	F_{mean} SD	$1.20e + 05(+)$ / $1.50e + 05$	$1.73e + 05(+)$ / $2.02e + 05$	$3.31e + 06(+)$ / $1.13e + 06$	$3.95e + 06(+)$ / $1.60e + 06$	$4.37e + 04(+)$ / $3.76e + 04$	$2.11e + 04(\approx)$ / $1.61e + 04$	$1.82e + 04/$ $1.12e + 04$
F_{03}	F_{mean} SD	$1.75e + 06(+)$ / $6.09e + 05$	$4.63e + 05(+)$ / $8.77e + 05$	$1.35e + 07(+)$ / $1.30e + 07$	$6.79e + 08(+)$ / $3.27e + 08$	$1.38e + 06(+)$ / $1.89e + 06$	$2.92e + 04(+)$ / $7.10e + 04$	$4.91e + 00/$ $2.52e + 01$
F_{04}	F_{mean} SD	$1.30e + 03(+)$ / $1.43e + 03$	$1.97e + 02(-)$ / $1.11e + 02$	$1.16e + 04(+)$ / $3.19e + 03$	$6.25e + 03(+)$ / $2.25e + 03$	$1.54e + 02(-)$ / $1.60e + 02$	$8.18e + 02(\approx)$ / $5.26e + 02$	$7.26e + 02/$ $3.26e + 02$
F_{05}	F_{mean} SD	$1.38e - 01(+)$ / $3.62e - 01$	$5.30e - 14(-)$ / $5.76e - 14$	$1.62e - 13(-)$ / $5.72e - 14$	$1.15e + 02(+)$ / $9.45e + 01$	$7.57e - 15(-)$ / $2.88e - 14$	$0.00e + 00(-)$ / $0.00e + 00$	$7.72e - 05/$ $2.23e - 05$
F_{06}	F_{mean} SD	$6.30e + 00(\approx)$ / $4.47e + 00$	$6.21e + 00(\approx)$ / $4.36e + 00$	$1.51e + 00(-)$ / $2.90e + 00$	$5.72e + 01(+)$ / $1.94e + 01$	$5.28e + 00(\approx)$ / $4.92e + 00$	$9.26e + 00(+)$ / $2.51e + 00$	$6.21e + 00/$ $4.80e + 00$
F_{07}	F_{mean} SD	$7.25e - 01(+)$ / $1.93e + 00$	$1.63e + 00(+)$ / $1.89e + 00$	$4.08e + 01(+)$ / $1.18e + 01$	$4.01e + 01(+)$ / $5.31e + 00$	$1.89e + 00(+)$ / $1.94e + 00$	$6.87e - 01(+)$ / $1.23e + 00$	$1.69e - 03/$ $3.05e - 03$
F_{08}	F_{mean} SD	$2.02e + 01(\approx)$ / $6.33e - 02$	$2.02e + 01(\approx)$ / $6.88e - 02$	$2.04e + 01(+)$ / $8.13e - 02$	$2.02e + 01(\approx)$ / $5.86e - 02$	$2.02e + 01(\approx)$ / $6.46e - 02$	$2.02e + 01(\approx)$ / $7.73e - 02$	$2.02e + 01/$ $4.62e - 02$
F_{09}	F_{mean} SD	$1.03e + 00(-)$ / $8.41e - 01$	$2.97e + 00(\approx)$ / $1.26e + 00$	$5.30e + 00(+)$ / $8.55e - 01$	$6.83e + 00(+)$ / $5.71e - 01$	$2.44e + 00(\approx)$ / $1.37e + 00$	$2.26e + 00(\approx)$ / $8.62e - 01$	$2.32e + 00/$ $1.29e + 00$
F_{10}	F_{mean} SD	$5.64e - 02(\approx)$ / $4.11e - 02$	$3.88e - 01(+)$ / $2.90e - 01$	$1.42e + 00(+)$ / $3.27e - 01$	$6.24e + 01(+)$ / $1.40e + 01$	$3.39e - 01(+)$ / $2.25e - 01$	$1.12e - 01(+)$ / $5.53e - 02$	$5.35e - 02/$ $2.81e - 02$
F_{11}	F_{mean} SD	$7.29e - 01(-)$ / $8.63e - 01$	$5.63e - 01(-)$ / $7.24e - 01$	$0.00e + 00(-)$ / $0.00e + 00$	$4.56e + 01(+)$ / $5.89e + 00$	$6.63e - 01(-)$ / $6.57e - 01$	$1.22e + 00(-)$ / $1.29e + 00$	$7.95e + 00/$ $3.30e + 00$
F_{12}	F_{mean} SD	$5.31e + 00(\approx)$ / $2.27e + 00$	$1.38e + 01(+)$ / $6.58e + 00$	$2.80e + 01(+)$ / $8.22e + 00$	$4.59e + 01(+)$ / $5.13e + 00$	$1.38e + 01(+)$ / $4.80e + 00$	$6.36e + 00(+)$ / $2.19e + 00$	$5.00e + 00/$ $1.21e + 00$
F_{13}	F_{mean} SD	$7.74e + 00(-)$ / $4.59e + 00$	$1.89e + 01(\approx)$ / $6.14e + 00$	$3.39e + 01(+)$ / $9.86e + 00$	$4.68e + 01(+)$ / $8.05e + 00$	$1.53e + 01(\approx)$ / $6.07e + 00$	$9.04e + 00(-)$ / $4.00e + 00$	$1.70e + 01/$ $8.72e + 00$
F_{14}	F_{mean} SD	$2.83e + 00(-)$ / $3.36e + 00$	$4.82e + 01(-)$ / $5.11e + 01$	$1.46e - 01(-)$ / $6.09e - 02$	$9.74e + 02(+)$ / $1.12e + 02$	$5.65e + 01(-)$ / $8.34e + 01$	$2.00e + 02(-)$ / $1.00e + 02$	$3.17e + 02/$ $1.53e + 02$
F_{15}	F_{mean} SD	$3.04e + 02(\approx)$ / $2.03e + 02$	$6.88e + 02(+)$ / $2.91e + 02$	$7.35e + 02(+)$ / $1.94e + 02$	$7.21e + 02(+)$ / $1.65e + 02$	$6.38e + 02(+)$ / $1.92e + 02$	$3.90e + 02(+)$ / $1.24e + 02$	$3.02e + 02/$ $1.15e + 02$
F_{16}	F_{mean} SD	$9.56e - 01(+)$ / $1.42e - 01$	$7.13e - 01(+)$ / $1.86e - 01$	$8.93e - 01(+)$ / $1.91e - 01$	$1.02e + 00(+)$ / $1.14e - 01$	$7.60e - 01(+)$ / $2.38e - 01$	$5.88e - 01(+)$ / $2.59e - 01$	$4.47e - 02/$ $4.10e - 02$
F_{17}	F_{mean} SD	$1.45e + 01(\approx)$ / $4.11e + 00$	$8.74e + 00(-)$ / $4.71e + 00$	$8.55e + 00(-)$ / $2.61e + 00$	$4.64e + 01(+)$ / $7.21e + 00$	$1.32e + 01(\approx)$ / $5.44e + 00$	$1.37e + 01(\approx)$ / $1.15e + 00$	$1.43e + 01/$ $1.62e + 00$
F_{18}	F_{mean} SD	$2.29e + 01(+)$ / $5.49e + 00$	$2.31e + 01(+)$ / $1.04e + 01$	$4.01e + 01(+)$ / $5.84e + 00$	$5.48e + 01(+)$ / $6.62e + 00$	$2.10e + 01(+)$ / $5.05e + 00$	$1.85e + 01(\approx)$ / $5.64e + 00$	$1.74e + 01/$ $4.65e + 00$
F_{19}	F_{mean} SD	$4.81e - 01(\approx)$ / $1.88e - 01$	$5.44e - 01(\approx)$ / $1.90e - 01$	$6.22e - 02(-)$ / $2.85e - 02$	$7.17e + 00(+)$ / $1.21e + 00$	$5.33e - 01(\approx)$ / $1.58e - 01$	$4.61e - 01(\approx)$ / $1.28e - 01$	$4.68e - 01/$ $1.69e - 01$
F_{20}	F_{mean} SD	$1.81e + 00(-)$ / $7.77e - 01$	$2.73e + 00(\approx)$ / $7.18e - 01$	$3.40e + 00(+)$ / $2.40e - 01$	$3.08e + 00(+)$ / $2.19e - 01$	$2.10e + 00(-)$ / $5.25e - 01$	$2.54e + 00(\approx)$ / $3.42e - 01$	$2.72e + 00/$ $5.83e - 01$
General merit over contender		3	3	7	19	3	3	

“+,” “-,” and “ \approx ” denote that the performance of CPEA ($D = 10$) is better than, worse than, and similar to that of other algorithms.

$$P = \begin{bmatrix} 0.36890 & 0.11700 & 0.26730 \\ 0.46990 & 0.43870 & 0.74700 \\ 0.1091 & 0.8732 & 0.55470 \\ 0.03815 & 0.57430 & 0.88280 \end{bmatrix}. \quad (\text{A.1})$$

f_{09} Six Hump Camel back function $f_{09}(x) = 4x_1^2 - 2.1x_1^4 + (1/3)x_1^6 + x_1x_2 - 4x_2^4 + 4x_2^6$, $[-5, 5]$, $\min(f_{09}) = -1.0316$.

f_{10} Matyas function (dimension = 2) $f_{10}(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$, $[-10, 10]$, $\min(f_{10}) = 0$.

f_{11} Zakharov function $f_{11}(x) = \sum_{i=1}^n x_i^2 + (\sum_{i=1}^n 0.5ix_i)^2 + (\sum_{i=1}^n 0.5ix_i)^4$, $[-5, 10]$, $\min(f_{11}) = 0$.

f_{12} Schwefel's problem 2.22 $f_{12}(x) = \sum_{i=1}^n |x_i| + \prod_{i=1}^n |x_i|$, $[-10, 10]$, $\min(f_{12}) = 0$.

f_{13} Levy function $f_{13}(x) = \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 \times (1 + \sin^2(3\pi x_{i+1})) + (x_n - 1)^2 (1 + \sin^2(2\pi x_n))$, $[-10, 10]$, $\min(f_{13}) = 0$.

f_{14} Schwefel's problem 2.21 $f_{14}(x) = \max\{|x_i|, 1 \leq i \leq n\}$, $[-100, 100]$, $\min(f_{14}) = 0$.

f_{15} Axis parallel hyperellipsoid $f_{15}(x) = \sum_{i=1}^n ix_i^2$, $[-5.12, 5.12]$, $\min(f_{15}) = 0$.

f_{16} De Jong's function 4 (no noise) $f_{16}(x) = \sum_{i=1}^n ix_i^4$, $[-1.28, 1.28]$, $\min(f_{16}) = 0$.

f_{17} Rastrigin's function $f_{17}(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$, $[-5.12, 5.12]$, $\min(f_{17}) = 0$.

f_{18} Ackley's path function $f_{18}(x) = -20 \exp(-0.2 \sqrt{(\sum_{i=1}^n x_i^2)/n}) - \exp(\sum_{i=1}^n \cos(2\pi x_i)/n) + 20 + e$, $[-32, 32]$, $\min(f_{18}) = 0$.

f_{19} Alpine function $f_{19}(x) = \sum_{i=1}^n |x_i \sin(x_i) + 0.1x_i|$, $[-10, 10]$, $\min(f_{19}) = 0$.

f_{20} Pathological function $f_{20}(x) = \sum_{i=1}^{n-1} (0.5 + (\sin^2 \sqrt{(100x_i^2 + x_{i+1}^2)} - 0.5)/(1 + 0.001(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2)))$, $[-100, 100]$, $\min(f_{20}) = 0$.

f_{21} Schaffer's function 6 (Dimension = 2) $f_{21}(x) = 0.5 + (\sin^2 \sqrt{(x_1^2 + x_2^2)} - 0.5)/(1 + 0.01(x_1^2 + x_2^2))$, $[-10, 10]$, $\min(f_{21}) = 0$.

f_{22} Camel Back-3 Three Hump Problem $f_{22}(x) = 2x_1^2 - 1.05x_1^4 + (1/6)x_1^6 + x_1x_2 + x_2^2$, $[-5, 5]$, $\min(f_{22}) = 0$.

f_{23} Exponential Problem $f_{23}(x) = -\exp(-0.5 \sum_{i=1}^n x_i^2)$, $[-1, 1]$, $\min(f_{23}) = -1$.

f_{24} Hartmann function 2 (Dimension = 6) $f_{24}(x) = -\sum_{i=1}^4 a_i \exp(-\sum_{j=1}^6 B_{ij}(x_j - Q_{ij})^2)$, $[0, 1]$,

$$a = [1 \ 1.2 \ 3 \ 3.2],$$

$$B = \begin{bmatrix} 10 & 3 & 17 & 3.05 & 1.7 & 8 \\ 0.05 & 17 & 10 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{bmatrix},$$

$$Q = \begin{bmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{bmatrix}, \tag{A.2}$$

$$\min(f_{24}) = -3.33539215295525.$$

f_{25} Shekel's Family_5 $m = 5$, $\min(f_{25}) = -10.1532$, $f_{25}(x) = -\sum_{i=1}^m [(x_i - a_i)(x_i - a_i)^T + c_i]^{-1}$,

$$a = \begin{bmatrix} 4 & 1 & 8 & 6 & 3 & 2 & 5 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 5 & 1 & 2 & 3.6 \\ 4 & 1 & 8 & 6 & 3 & 2 & 3 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 3 & 1 & 2 & 3.6 \end{bmatrix}^T, \tag{A.3}$$

$$c = [0.1 \ 0.2 \ 0.2 \ 0.4 \ 0.4 \ 0.6 \ 0.3 \ 0.7 \ 0.5 \ 0.5]^T.$$

TABLE 9

i	a_i	b_i	c_i	d_i
1	0.5	0.0	0.0	0.1
2	1.2	1.0	0.0	0.5
3	1.0	0.0	-0.5	0.5
4	1.0	-0.5	0.0	0.5
5	1.2	0.0	1.0	0.5

f_{26} Shekel's Family_7 $m = 7$, $\min(f_{26}) = -10.4029$,

f_{27} Shekel's Family_10 $m = 10$, $\min(f_{27}) = f(4, 4, 4, 4) = -10.5364$.

f_{28} Goldstein and Price $[-2, 2]$, $\min(f_{28}) = 3$

$$f_{28}(x) = [1 + (x_1 + x_2 + 1)^2 \cdot (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \cdot (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]. \tag{A.4}$$

f_{29} Becker and Lago Problem $f_{29}(x) = (|x_1| - 5)^2 - (|x_2| - 5)^2$, $[-10 \leq 10]$, $\min(f_{29}) = 0$.

f_{30} Hosaki Problem $f_{30}(x) = (1 - 8x_1 + 7x_1^2 - (7/3)x_1^3 + (1/4)x_1^4)x_2^2 \exp(-x_2)$, $0 \leq x_1 \leq 5$, $0 \leq x_2 \leq 6$, $\min(f_{30}) = -2.3458$.

f_{31} Miele and Cantrell Problem $f_{31}(x) = (\exp(x_1) - x_2)^4 + 100(x_2 - x_3)^6 + (\tan(x_3 - x_4))^4 + x_1^8$, $[-1, 1]$, $\min(f_{31}) = 0$.

f_{32} Quartic function, that is, noise $f_{32}(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$, $[-1.28, 1.28]$, $\min(f_{32}) = 0$.

f_{33} Rastrigin's function A $f_{33}(x) = F_{\text{Rastrigin}}(z)$, $[-5.12, 5.12]$, $\min(f_{33}) = 0$.

$z = x * M$ D is the dimension, M is a $D \times D$ orthogonal matrix.

f_{34} Multi-Gaussian Problem $f_{34}(x) = -\sum_{i=1}^5 a_i \exp(-((x_1 - b_i)^2 + (x_2 - c_i)^2)/d_i^2)$, $[-2, 2]$, $\min(f_{34}) = -1.29695$ (see Table 9).

f_{35} Inverted cosine wave function (Masters) $f_{35}(x) = -\sum_{i=1}^{n-1} (\exp(-((x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}))/8) \times \cos(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}}))$, $[-5, 5]$, $\min(f_{35}) = -n + 1$.

f_{36} Periodic Problem $f_{36}(x) = 1 + \sin^2 x_1 + \sin^2 x_2 - 0.1 \exp(-x_1^2 - x_2^2)$, $[-10, 10]$, $\min(f_{36}) = 0.9$.

f_{37} Bohachevsky 1 Problem $f_{37}(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$, $[-50, 50]$, $\min(f_{37}) = 0$.

f_{38} Bohachevsky 2 Problem $f_{38}(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) \cos(4\pi x_2) + 0.3$, $[-50, 50]$, $\min(f_{38}) = 0$.

f_{39} Salomon Problem $f_{39}(x) = 1 - \cos(2\pi\|x\|) + 0.1\|x\|$, $[-100, 100]$, $\|x\| = \sqrt{\sum_{i=1}^n x_i^2}$, $\min(f_{39}) = 0$.

f_{40} McCormick Problem $f_{40}(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - (3/2)x_1 + (5/2)x_2 + 1$, $-1.5 \leq x_1 \leq 4$, $-3 \leq x_2 \leq 3$, $\min(f_{40}) = -1.9133$.

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 (nos. 61272283 and U1334211), Shaanxi Science and Technology Project (nos. 2014KW02-01 and 2014XT-11), and Shaanxi Province Natural Science Foundation (no. 2012JQ8052).

References

- [1] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies—a comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.
- [2] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, A Bradford Book, 1992.
- [3] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, no. 6791, pp. 39–42, 2000.
- [4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.
- [5] A. Banks, J. Vincent, and C. Anyakoha, "A review of particle swarm optimization. I. Background and development," *Natural Computing*, vol. 6, no. 4, pp. 467–484, 2007.
- [6] B. Basturk and D. Karaboga, "An artificial bee colony(ABC) algorithm for numeric function optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium*, pp. 12–14, Indianapolis, Indiana, USA, 2006.
- [7] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [8] L. N. De Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, Springer, Berlin, Germany, 2002.
- [9] M. G. Gong, H. F. Du, and L. C. Jiao, "Optimal approximation of linear systems by artificial immune response," *Science in China. Series F. Information Sciences*, vol. 49, no. 1, pp. 63–79, 2006.
- [10] S. Twomey, "The nuclei of natural cloud formation part II: the supersaturation in natural clouds and the variation of cloud droplet concentration," *Geofisica Pura e Applicata*, vol. 43, no. 1, pp. 243–249, 1959.
- [11] F. A. Wang, *Cloud*, Meteorological Press, Beijing, China, 2002, (Chinese).
- [12] Z. C. Gu, *The Foundation of Cloud, Mist and Precipitation*, Science Press, Beijing, China, 1980 (Chinese).
- [13] R. C. Reid and M. Modell, *Thermodynamics and Its Applications*, Prentice Hall, New York, NY, USA, 2008.
- [14] W. Greiner, L. Neise, and H. Stöcker, *Thermodynamics and Statistical Mechanics*, Classical Theoretical Physics, Springer, New York, NY, USA, 1995.
- [15] P. Elizabeth, "On the origin of cooperation," *Science*, vol. 325, no. 5945, pp. 1196–1199, 2009.
- [16] M. A. Nowak, "Five rules for the evolution of cooperation," *Science*, vol. 314, no. 5805, pp. 1560–1563, 2006.
- [17] J. N. Thompson and B. M. Cunningham, "Geographic structure and dynamics of coevolutionary selection," *Nature*, vol. 417, no. 13, pp. 735–738, 2002.
- [18] J. B. S. Haldane, *The Causes of Evolution*, Longmans, Green & Co, London, UK, 1932.
- [19] W. D. Hamilton, "The genetical evolution of social behaviour," *Journal of Theoretical Biology*, vol. 7, no. 1, pp. 17–52, 1964.
- [20] R. Axelrod and W. D. Hamilton, "The evolution of cooperation," *American Association for the Advancement of Science: Science*, vol. 211, no. 4489, pp. 1390–1396, 1981.
- [21] M. A. Nowak and K. Sigmund, "Evolution of indirect reciprocity by image scoring," *Nature*, vol. 393, no. 6685, pp. 573–577, 1998.
- [22] A. Traulsen and M. A. Nowak, "Evolution of cooperation by multilevel selection," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 29, pp. 10952–10955, 2006.
- [23] L. Shi and R. Wang, "The evolution of cooperation in asymmetric systems," *Science China Life Sciences*, vol. 53, no. 1, pp. 1–11, 2010 (Chinese).
- [24] D. Y. Li, "Uncertainty in knowledge representation," *Engineering Science*, vol. 2, no. 10, pp. 73–79, 2000 (Chinese).
- [25] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky, "A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems," *Journal of Global Optimization*, vol. 31, no. 4, pp. 635–672, 2005.
- [26] 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.
- [27] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.
- [28] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.
- [29] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.
- [30] 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.
- [31] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the Congress on Evolutionary Computation*, pp. 1671–1676, Honolulu, Hawaii, USA, May 2002.
- [32] T. Peram, K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium*, pp. 174–181, Indianapolis, Ind, USA, April 2003.

- [33] 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>.
- [34] G. W. Zhang, R. He, Y. Liu, D. Y. Li, and G. S. Chen, "An evolutionary algorithm based on cloud model," *Chinese Journal of Computers*, vol. 31, no. 7, pp. 1082–1091, 2008 (Chinese).
- [35] N. Hansen and S. Kern, "Evaluating the CMA evolution strategy on multimodal test functions," in *Parallel Problem Solving from Nature—PPSN VIII*, vol. 3242 of *Lecture Notes in Computer Science*, pp. 282–291, Springer, Berlin, Germany, 2004.
- [36] D. Karaboga and B. Akay, "A comparative study of artificial Bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.
- [37] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *CAD Computer Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.



Hindawi

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

